Quantifying Quality:
How to Quantify Quality:
Finding Scales of measure
by
Tom Gilb

UIO, 1015-1200 Store Aud. Inst for Informatikk
Mandag 24. oktober
Tom@Gilb.com
www.Gilb.com
Result Planning Limited
Norway/UK
Updated 16 Sept 2005

Philolaus on Numbers

• Over four hundred years BC, a Greek by the name of Philolaus of Tarentum said:
  
• "Actually, everything that can be known has a Number;
• for it is impossible to grasp anything with the mind or to recognize it without this (number).

• Best regards (Aug 2005)
  N.V.Krishna www.microsensesoftware.com
"You can nearly measure everything but how can you measure style?"
That's Siemens catchphrase for its new S65

QuickTime™ and a TIFF (Uncompressed) decompressor are needed to see this picture.

How to Quantify any Qualitative Requirement

[Diagram from "Competitive Engineering" book.]
**Quality:** the concept, the noun

A *quality* is
- a scalar attribute
- reflecting *how well*
- a system functions.

**Quality** is characterized by these traits

1. Quality describes *how well* a function is done.
2. Quality describes the *partial effectiveness* of a function (as do all other performance attributes).
3. Quality is *valued* to some degree by some stakeholders of the system.
4. More quality is generally valued by stakeholders; especially if the increase is free, or lower cost, than the value of the increase.
5. Quality attributes can be *articulated* independently of the particular means (designs) used for reaching a specific quality level –
6. even though all quality levels *depend* on the particular designs used to achieve them.
7. A particular quality can be a described in terms of a *complex* concept, consisting of multiple elementary quality concepts.
8. Quality is *variable* (along a definable scale of measure: as are all scalar attributes).
9. Quality levels are capable of being specified *quantitatively* (as are all scalar attributes).
10. Quality levels can be *measured* in practice.
11. Quality levels can be traded off to some degree; with other system attributes valued more by stakeholders.
12. Quality can never be perfect (100%), in the real world.
13. There are some levels of a particular quality that may be outside the state of the art; at a defined time and circumstance.
14. When quality levels increase towards perfection, the resources needed to support those levels tend towards infinity.
Exercise: Aspects of Love, or Love is a many splendored thing!

- Make inventory of love’s many aspects
- Quantify one requirements for love
- Duration: 6 minutes

Love Attributes:
Brainstormed By Dutch Engineers

- Kissed-ness
- Care
- Sharing
- Respect
- Comfort
- Friendship
- Sex
- Understanding
- Trust
- Support
- Attention
- Passion
- Satisfaction
- ...
- ...
- ...

See note for Sutra
Trust [Caroline]

- **Love, Trust, Truthfulness**
  - Ambition: No lies.
  - Scale:
    - Average Black lies/month from [defined sources].
  - Meter:
    - independent confidential log from sample of the defined sources.
  - Past Lie Level:
    - Past [My Old Mate, 2004] 42 <-Bart
  - Goal
    - [My Current Mate, Year = 2005] Past Lie Level/2
    - **Black**: Defined: Non White Lies

- Other aspects of Trust:
  - Broken Agreements
  - Late Appointments
  - Late delivery
  - Gossiping to Others

Camaraderie (Real Case UK)

- Ambition: to maintain an exceptionally high sense of good personal feelings and cooperation amongst all staff: family atmosphere, corporate patriotism. In spite of business change and pressures.
- Scale: probability that individuals enjoy the working atmosphere so much that they would not move to another company for less than 50% pay rise.
- Meter: Apparently real offer via CD-S
- Past [September 2001] 60+ % <- R & CD
- Rationale:
  - maintain staff number, and morale as core of business and business predictability for customers.
A person who loves acts the following way toward the person being loved:

1. suffereth long
2. is kind
3. envieth not
4. vaunteth not itself, vaunteth...
    or, is not rash  (Vaunt = extravagant self praise)
5. is not puffed up
6. Doth not behave itself unseemly
7. seeketh not her own
8. is not easily provoked
9. thinketh no evil
10. Rejoiceth not in iniquity (=an unjust act)
11. rejoiceth in the truth
12. Bareth all things
13. believeth all things
14. hopeth all things
15. endareth all things
16. never faileth

The biblical citation (Book of First Corinthians) I included gives the quantification of the term "love" (agape in Greek). The 'quantification' for love would be as follows:

---

What can we do better (or ‘at all’), if we quantify quality ideas?

- Evaluation solutions/designs/architectures against the quantified quality requirements (Impact Estimation)
- Test and measure the degree to which solutions meet quality and cost expectations (when they were chosen)
- Measure evolutionary project progress towards quality goals
  - And get early & continuous improved estimates for time to completion
- Communicate quality goals much better to all parties (users, customers, developers, testers, lawyers)
- Contract for results
  - Pay for results only (not effort expended)
- Reward teams for results achieved
- Motivate technical people to focus on real business results
- Simplify requirements (the top few quantified- everything else is design)
- Collect numeric data about designs, processes, organizational structures, to learn and use in future.
- Permits systematic corporate or academic research of a development environment

© 2004 Tom@Gilb.com
Projected benefits of this include:
• reduced time lost in planning,
• quicker identification of actual and potential operational problems,
• reduced time in vehicle tracking for customers and internal purposes,
• better matching of operational costs and effort to sales contracts,
• better information for future contract negotiations & renegotiation
• -----------------

The perceived benefits of better planning and management of high & heavy cargo are:
• reduced manual effort in planning movements,
• better performance to target delivery dates for high & heavy,
• better terminal planning for the cargo,
• better terminal operation from better information about handling,
• better customer management from better information on progress.
• ===============================

Consolidated, consistent and timely planning information will:
• reduce the incidence of wrong booking and loading of cargo,
• reduce double handling and recording of information,
• give visibility of planning data along the full distribution chain,
• allow marketing to give more accurate information to customers,
• increase utilization of COMPANY’s own transport, and
• - reduce the amount of emergency third party charter.

What is wrong with this (previous slide) picture?

Some more detail in the same ‘functional’ requirements: (is this a design?)

1. It must be possible to select any cargo, including High & Heavy and MAFI, based on any of:
   - VIN (either complete or a subset, typically the last 5, 6, 8 or 10 characters)
   - tracking number
   - serial number
   - multiple VINS (eg cut & paste input),
   - movement,
   - customer’s batch number,
   - transport ID (rail wagon no or MAFI, lorry, vessel),
   - customer code
   - customer’s sales order number
   - customer’s manufacturing order no (also called Commission or ED no)
   - at location on date (by destination)
   - dealer code
   - model type & make

• No identification of the main benefits (just bullet points)
• No definition of the quantification (no ‘Scale’ specification)
• No benchmark to help define ‘better’.
• No target to define ‘better’
• No dates to define when ‘better’
• No evidence that the ‘designs’ in the requirements will give any of the cited results
• No specification of the long term value or costs of the suggested designs (in the requirements)
• AND MANY MORE PROBLEMS
   - Sources
   - Authority
   - Risks
   - Priorities
From waterfall to Evo
The path and experiences
Trond Johansen, QA & Process Manager, FIRM
AS
Trond.Johansen@firmglobal.com
Seminar IT Fornebu 13.september, Oslo

Future Information Research Management – FIRM

*Powered by Innovation*
*Driven by Commitment*

Customer Successes in Corporate Sector
FIRM R&D department

- Developers (13)
- Management/(CSO) (2)
- Tech Support NY (1)
- Microsoft .NET framework, SQL

- SEPG group (3) with responsibility of process improvement and quality assurance (QA).
  - Configuration Management, setup +
  - Testing
  - Software Process Improvement (SPI)

FIRM’s interpretation of Evo Method:
Basis for the 3 month trial period

- After
  - the one day crash course with Tom and Kai Gilb and
  - a literature study (“Competitive Engineering” by Tom Gilb,
  - “The Evo Manuscript” by Kai Gilb,
  - and other material on the subject),
- our overall understanding of EVO was this:
  - Find stakeholders (End users, super-users, support, sales, IT Operations etc)
  - Define the stakeholders’ real needs and the related Product Qualities
  - Identify past/status of product qualities and your goal (how much you want to improve)
  - Identify possible solutions for meeting your goals
  - Develop a step-by-step plan for
    - delivering improvements
    - with respect to Stakeholder Values & Product Quality goals:
      - Deliveries every week
      - Measure: are we moving towards our goals?
Requirements - 1, Paradigm Shift

- With EVO, our requirements process changed.
  - Previously we focused mostly on function requirements.
  - We realized that it’s the quality requirements that really separate us from our competitors.
- Why are spell checkers, which is included in many software products today, killer applications?
  - Superior product qualities: Performance, Speed, Usability

Requirements - 2, Our ‘standards’

- We tried to define our requirements according to a basic standard (in ‘Competitive Engineering’ (Gilb, 2005), Rules):
  - Clear & Unambiguous
  - Testable
  - Measurable
  - No Solutions (Designs)
  - Stakeholder Focus
- Example: specification of User-Friendliness
  - “We will achieve state-of-the-art user friendliness by using MS-Windows GUI combined with pop up help and one push button to chat directly with a live support person”

  - User Friendliness, Learn
  Stakeholder: End User
  Scale: average time to learn, how to do, 10 defined tasks.
Requirements - 3, Real Example of Spec

- **Usability.Productivity** *(taken from Confirmit 8.5 development)*
  - **Scale:** Time in minutes to set up a typical specified MR-report
  - **Past:** 65 min,
  - **Tolerable:** 35 min,
  - **Goal:** 25 min
    - (end result was 20 min 😊)
  - **Meter:** Candidates with Reportal experience and with knowledge of MR-specific reporting features performed a set of predefined steps to produce a standard MR Report. (The standard MR report was designed by Mark Phillips, an MR specialist at our London office)

- The focus is here on the **day-to-day operations** of our MR users,
  - not a list of features that they might or might not like.
  - We KNOW that increased efficiency which leads to more profit will please them. 45 min * thousands of reports= $$ saved

- After **one week** we had defined more or less all the requirements for the next version of Confirmit.

---

Solutions

- For every quality requirement we looked for possible Solutions (Design Ideas)
- E.g. for Quality Requirement: Usability.Productivity we identified the following Design Ideas:
  - DesignIdea.Recoding
  - DesignIdea.MRTotals
  - DesignIdea.Categorizations
  - DesignIdea.TripleS
  - ...and many more

- We evaluated all these, and specified in more detail those we believed would add the **most value** (take us closer to the goal)
EVO - 1, The Impact Estimation Table

• We
  – collected the most promising Solutions/Design Ideas and
  – included them in an EVO plan
    • (using Impact Estimation Table: IET, as an Evo tool)
• The IE Table is
  – our tool for controlling the qualities and
  – delivering improvements to real stakeholders,
    • or as close as we can get to them. (e.g. ProS acting as clients)
• One Evo Step = 1 week!

EVO - 2, project step planning and accounting: using an Impact Estimation Table

• IET for MR Project – Confirmit 8.5
• Solution: Recoding
  – Make it possible to recode variable on the fly from Reportal.
  – Estimated effort: 4 days
  – Estimated Productivity Improvement: 20 minutes (50% way to Goal)
  – actual result 38 minutes (95% progress towards Goal)
EVO – 3, The Evo cycle

- We decided that
  - one EVO step should last one week
  - because of practical reasons,
  - even though we violate the Evo policy guideline of not spending more than 2% of project schedule, in each step

- Project management meetings
  - In the Project Management meetings on Fridays,
    - each project leader presents the results from the previous step (IET)
    - as well as the content of next EVO step (one week)
  - Possible new Solutions are
    - discussed and
    - weighted against each other;
      - Most value for development resources

Tom Gilb 26

EVO Plan Confirmit 8.5
4 more product areas were attacked concurrently

Tom Gilb 26
FIRM EVO week

<table>
<thead>
<tr>
<th>Development Team</th>
<th>Users (PMT, Pros, Doc writer, other)</th>
<th>CEO (Sys. Arch, Process Mgr)</th>
<th>QA (Configuration Manager &amp; Test Manager)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Friday</td>
<td>✓ PM: Send Version N detail plan to CEO ✓ prior to Project Mgmt meeting ✓ PM: Attend Project Mgmt meeting: 12.00-1.00 ✓ Developers: Focus on general maintenance work, documentation.</td>
<td>✓ Approve/reject design &amp; Step N ✓ Attend Project Mgmt meeting: 12-15</td>
<td>✓ Run final build and create setup for Version N-1 ✓ Install setup on test servers (external and internal) ✓ Perform initial crash test and then release Version N-1</td>
</tr>
<tr>
<td>Monday</td>
<td>✓ Develop test code &amp; code for Version N ✓ Use Version N-1</td>
<td></td>
<td>✓ Follow up C1 ✓ Review test plans, tests</td>
</tr>
<tr>
<td>Tuesday</td>
<td>✓ Develop Test Code &amp; Code for Version N ✓ Meet with users to Discuss Action Taken Regarding Feedback From Version N-1</td>
<td>✓ Meet with developers to give Feedback and Discuss Action Taken from previous actions ✓ System Architect to review code and test code</td>
<td>✓ Follow up C1 ✓ Review test plans, tests</td>
</tr>
<tr>
<td>Wednesday</td>
<td>✓ Develop test code &amp; code for Version N</td>
<td></td>
<td>✓ Review test plans, tests ✓ Follow up C1</td>
</tr>
<tr>
<td>Thursday</td>
<td>✓ Complete Test Code &amp; Code for Version N ✓ Complete GUI tests for Version N-2</td>
<td></td>
<td>✓ Review test plans, tests ✓ Follow up C1</td>
</tr>
</tbody>
</table>

Experiences and conclusions – 1

- We launched our first major release based on Evo in May 2004
  - and we have already gotten feedback from users on some of the leaps in product qualities.
  - E.g. the time for the system to generate a complex survey has gone from 2 hours (=wait for the system to do work) to 15 seconds!
- EVO has resulted in
  - increased motivation and
  - enthusiasm amongst developers,
  - it opens up for empowered creativity
- Developers
  - embraced the method and
  - saw the value of using it,
  - even though they found parts of Evo difficult to understand and execute
- Project leaders feel:
  - Defining good requirements can be hard.
  - It was hard to find meters which were practical to use, and at the same time measure real product qualities.
  - Sometimes we would like to spend more than a day on designs, but this was not right according to our understanding of Evo. (Concept of backroom activity was new to us)
  - Sometimes it takes more than a week to deliver something of value to the client. (Concept of backroom activity was new to us)
Experiences and conclusions – 2

– Team members (developers)
  • “Sometimes it felt like we’re rushing to the next weekly step before we had finished the current step”
  • Testing was sometimes postponed
    – in order to start next step,
    – some of these test delays were not compensated for up in later testing.

Evo’s impact on Confirmit product qualities - 1

• The impact described is based on:
  – Internal usability test, productivity tests ++
  – Performance tests carried out at Microsoft Windows ISV laboratory in Redmond USA
  – Direct customer feedback
    • “I just wanted to let you know how appreciative we are of the new “entire report” export functionality you recently incorporated into the Reportal. It produces a fantastic looking report, and the table of contents is a wonderful feature. It is also a HUGE time saver.”
  – These leaps in product qualities would not have been achieved without Evo.
EVO’s impact on Confirmit product qualities - 2

Only highlights of the impacts are listed here

<table>
<thead>
<tr>
<th>Description of requirement/work task</th>
<th>Past</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usability/Productivity: Time for the system to generate a survey</td>
<td>7200 sec</td>
<td>15 sec</td>
</tr>
<tr>
<td>Usability/Productivity: Time to set up a typical specified Market Research-report (MR)</td>
<td>65 min</td>
<td>20 min</td>
</tr>
<tr>
<td>Usability/Productivity: Time to grant a set of End-users access to a Report set and distribute report login info.</td>
<td>80 min</td>
<td>5 min</td>
</tr>
<tr>
<td>Usability/Intuitiveness: The time in minutes it takes a medium experienced programmer to define a complete and correct data transfer definition with Confirmit Web Services without any user documentation or any other aid</td>
<td>15 min</td>
<td>5 min</td>
</tr>
<tr>
<td>Performance/Runtime.Concurrency: Maximum number of simultaneous respondents executing a survey with a click rate of 20 sec and an response time&lt;500 ms, given a defined [Survey-Complexity] and a defined [Server Configuration, Typical]</td>
<td>250 users</td>
<td>6000</td>
</tr>
</tbody>
</table>

Lessons learned

- We will have increased **focus on feedback** from clients. We will select the ones that are willing to dedicate time to us
- Demonstrate new functionality with screen recording software or early test plans >> easier for internal stakeholders to do **early testing**
- Tighter integration between EVO and the test process
- Plan for less, and deliver less better!
Conclusions - 1

- The method’s **positive impact** on Confirmit product qualities has convinced us that
  - Evo is a better suited development process than our former waterfall process, and
  - we will continue to use Evo in the future.
- What **surprised** us the most was
  - the method’s **power of focusing on delivering value** for clients versus cost of implementation.
  - Evo enables you to **re-prioritize** the next development-steps based on the weekly feedback
  - What seemed important
    - at the start of the project
    - may be replaced by other solutions
    - based on gained knowledge from previous steps.
- The method has
  - high focus on **measurable product qualities**, and
  - defining these clearly and testably requires training and maturity.
  - It is important to **believe that everything can be measured** and to seek guidance if it seems impossible.

Conclusions - 2

- A pre-requisite related to the method for using Evo is an **open architecture**.
- Another pre-requisite is **management support** for changing the work process, and this is important in any software process improvement initiative.
- The concept of **Continuous Integration (CI)/daily builds**
  - was valuable
  - with respect to delivering new versions of the software every week.
- Evo,
  - as most other software processes,
  - requires continuous focus
  - and learning about the methodology.
The way ahead

• Overall, the whole organization has embraced EVO.
• We all think it has great potential,
  – and we will work hard to utilize it to the full.
• In June 2004
  – we had Tom and Kai Gilb for a 4 days course for the whole
    R&D department and related resources

• The next version of Confirmit, Confirmit 9.0, will prove whether
  we have matured in our understanding and execution of EVO
• Confirmit 9.0 is due to be released Q4 2004, here is a sneak
  preview…

Confirmit 9.0 and product qualities

<table>
<thead>
<tr>
<th>Description of requirement/work task</th>
<th>Past</th>
<th>Status 11.09</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usability, Intuitiveness: Probability that a defined User can intuitively figure out how to do a defined Task correctly (without any errors needing correction)</td>
<td>30%</td>
<td>45%</td>
<td>80%</td>
</tr>
<tr>
<td>Panel, Scalability: Maximum number of panelists that the system can support within a timeframe of 120 seconds for creating a sample of 50 000, with all components of the panel system performing acceptably</td>
<td>30,000</td>
<td>300,000</td>
<td>200,000</td>
</tr>
<tr>
<td>Performance, Data Volume: Numbers of survey responses that can be handled by Reportal. Tables should be generated within 5 seconds.</td>
<td>20,000</td>
<td>300,000</td>
<td>500,000</td>
</tr>
</tbody>
</table>
EVO’s impact on Confirmit product qualities - 2

- Only highlights of the impacts are listed here

<table>
<thead>
<tr>
<th>Description of requirement/work task</th>
<th>Past</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usability: Productivity: Time for the system to generate a survey</td>
<td>7200 sec</td>
<td>15 sec</td>
</tr>
<tr>
<td>Usability: Productivity: Time to set up a typical specified Market Research-report (MR)</td>
<td>65 min</td>
<td>20 min</td>
</tr>
<tr>
<td>Usability: Productivity: Time to grant a set of End-users access to a Report set and distribute report login info.</td>
<td>80 min</td>
<td>5 min</td>
</tr>
<tr>
<td>Usability: Intuitiveness: The time in minutes it takes a medium experienced programmer to define a complete and correct data transfer definition with Confirmit Web Services without any user documentation or any other aid</td>
<td>15 min</td>
<td>5 min</td>
</tr>
<tr>
<td>Performance: Runtime: Concurrency: Maximum number of simultaneous respondents executing a survey with a click rate of 20 sec and a response time&lt;500 ms, given a defined [Survey-Complexity] and a defined [Server Configuration, Typical]</td>
<td>250 users</td>
<td>6000</td>
</tr>
</tbody>
</table>

Release 8.5

Initial Customer Feedback on the new Confirmit 9.0

November 24th, 2004
Initial perceived value of the new release
(Base 73 people)

![Bar chart showing the extent to which users feel Conformit 9.0 will give them additional value.]

Tom Gilb 39

Initial qualitative feedback on the new release

"... keep up the good work."

"It looks like you have listened to the people that actually use the software daily and aimed to make it easier for them."

"I was very impressed with the version 9.0."

- Seminar observations
  - On several occasions, customers gave spontaneous "WOWs" and applauses!
  - The training room in London was literally packed with people eager to test the new version.
  - Several clients asked if they could access the test server from home as well.
  - Great participation rate; 95% of all registered people showed up.

Tom Gilb 40
New version of Confirmit increases user productivity up to 80 percent

We believe the improvements in Confirmit 9.0 will benefit Greenfield Online's survey programming, data collection and reporting capabilities allowing us to bring leading edge technology to our customers. -

FIRM is using Evolutionary (EVO) development to ensure the highest focus on customer value through early and continuous feedback from stakeholders. A key component in EVO is measuring the effect new and improved product qualities have on customer value. Increased customer value in Confirmit 9.0 includes:

- Up to 175 percent more intuitive user interface*
- Up to 1500 percent increased performance in Reportal and Panel Management* Features delivering increased customer value include:
- Random Data Generator enabling automated testing of questionnaires
- Real-time Script Checker for on-the-fly script validation
- Block Randomization of questions to avoid respondent bias
- Reportal BitStream for fast online tabulation on high volume of responses
- We are very pleased to see major improvements in Confirmit 9.0, including updates to both the user interface and survey engine. We plan to deploy this new version when it becomes available to server customers, stated Alex Umsnberg, Greenfield Online's Chief Information Officer.

We believe the improvements in Confirmit 9.0 will benefit Greenfield Online's survey programming, data collection and reporting capabilities allowing us to bring leading edge technology to our customers. -

FIRM is using Evolutionary (EVO) development to ensure the highest focus on customer value through early and continuous feedback from stakeholders. A key component in EVO is measuring the effect new and improved product qualities have on customer value. Increased customer value in Confirmit 9.0 includes:

- Up to 175 percent more intuitive user interface*
- Up to 1500 percent increased performance in Reportal and Panel Management* Features delivering increased customer value include:
- Random Data Generator enabling automated testing of questionnaires
- Real-time Script Checker for on-the-fly script validation
- Block Randomization of questions to avoid respondent bias
- Reportal BitStream for fast online tabulation on high volume of responses
- We are very pleased to see major improvements in Confirmit 9.0, including updates to both the user interface and survey engine. We plan to deploy this new version when it becomes available to server customers, stated Alex Umsnberg, Greenfield Online's Chief Information Officer.

*Measured in FIRM's ASP environments in London and New York on November 27th. The new version will be available for server customers in January 2005.

Press contact: Kjell Öksendal, FIRM’s VP of Marketing +1 646 229 5635

A detailed real example of Quality Specification (Oct 2004, Europe)

In an effort to standardize product development and engineering quality evaluation, we have chosen to define a set of common definitions. We believe this effort helps align and synchronize our practices and auditable achievements. Under this approach, we define the following set of terms and definitions.

Rationale: least powerful sales argument for selling new version.

Assumption: the level of qualities is the same for comparative measurements. E.g. we do not assume that, only to turn around and use it to increase quality, the old level was for the old quality level. <70%-

Note: we lack clarity in Stakeholder to be served at each step. This decides some things to be at least classified to define % must be developed, or engineering effort and time. <table

GLOSSARY------------------

Complete:

Tom Gilb 41

Quantifying Quality

Tom Gilb 42
Al Says

“Not everything that can be counted counts, and not everything that counts can be counted.”

Albert Einstein

I agree. But this does not include system qualities.

Tom

Simon Ramo (tRw)

“No matter how complex the situation,
good systems engineering involves putting value measurements on the important parameters of desired goals and performance of pertinent data,
and of the specifications of the people and equipment and other components of the system.

It is not easy to do this
and so, very often, we are inclined to assume that it is not possible to do it to advantage.

But skilled systems engineers can
change evaluations and comparisons of alternative approaches
from purely speculative to highly meaningful.

If some critical aspect is not known,
the systems experts seek to make it known.
They go dig up the facts.
If doing so is very tough, such as setting down the public’s degree of acceptance among various candidate solutions, then perhaps the public can be polled.
If that is not practical for the specific issue, then at least an attempt can be made to judge the impact of being wrong in assuming the public preference.

Everything that is clear is used with clarity:
what is not clear is used with clarity as to the estimates and assumptions made,
with the possible negative consequences of the assumptions weighed and integrated.

We do not have to work in the dark, now that we have professional systems analysis.

Ramo98 page 81
**How to Quantify Quality**

- **Plan**
  - Use known quantification ideas

- **Do**
  - Modify known quantification ideas to suit your current problems

- **Study**
  - Use your common sense and powers of observation to work out new measures

- **Act**
  - Learn early, learn often, adjust early definitions

---

**‘Environmentally Friendly’ Quantification Example**

- Give the quality a stable name tag
  - Environmentally Friendly

- Define approximately the target level
  - Ambition Level: A high degree of protection

- Define a scale of measure:
  - Scale: % change in environment

- Decide a way to measure in practice.
  - Meter: [scientific data...]

- Define benchmarks.
  - Past [2003] +50% <-intuitive
  - Record [2002,...] 0%
  - Trend [2007,...] -30%

- Define Constraints (Fail) and targets (Goal, Wish).
  - Fail[next year] +0% <-not worse
  - Goal +5 years, ... +30% <-TG
  - Wish [2007,...] +50% <-Marketing

---

Tom Gilb 45

Tom Gilb 46
Using ‘Parameters’ when defining a Scale of Measure

• Using [qualifiers] in the SCALE definition
  – gives flexibility of detailed specification later.
• Example
  – SCALE: the % of
    • defined [Users]
    • using defined [system Components]
    • who can successfully accomplish defined [Tasks]

Goal
[ Users = NOVICES, Components = USER MANUAL, Tasks = ERROR CORRECTION ]
60%
**Quality Quantification Process**

(full detail ‘Competitive Engineering’, Scales chapter, & slide here later ‘QQ’)

**Entry**

E1. Do not enter if you can **reuse** existing standards.
E2. Do not enter if your **source** documents are **poor**.

**Procedure**

P1. Use applicable **rules** (GR, QR, QQ).
P2. **Build list** of quality ideas needing control.
P3. **Detail** qualities by exploding hierarchically.
   - use evolutionary or pilot **feedback**.
P4. Revise your draft based on **design work**.
P5. **Quality Control** the specification.
P6. Get **experience** and then revise specifications.

**Exit**

X1. Don’t exit if calculated **remaining defects** are more than one per page.
X2. Unless you intentionally do so to learn more from experience.

---

**General Hatmanship:**

**GIST:** improve ability to have hats on head and nearby

**Hatmanship On Head:**

**SCALE:** hats on top of persons head

PAST [Me, This year] 10 ← Guess

RECORD [2003, UK] 15 ← GB Record

WISH [Guinness Record, April] 20 ← Tom

**Hatmanship Nearby:**

**SCALE:** hats not on head, but on, or near, body within 10 meter radius.

Past... Goal......etc.

**A ‘Quality Quantification’ Principle**

0. **THE PRINCIPLE OF**

'BAD NUMBERS BEAT GOOD WORDS'

Poor quantification is more useful than none; at least it can be improved systematically.
Quantify for realistic judgements

• “To leave [soft considerations] out of the analysis
  – simply because they are not readily quantifiable
  – or to avoid introducing “personal judgments,”
  – clearly biases decisions against investments
    • that are likely to have a significant impact on considerations
      – as the quality of one’s product, delivery speed and reliability, and the rapidity with which new products can be introduced”

• ⇨ R. H. Hayes et al “Dynamic Manufacturing”, p. 77 in MINTZBERG94: page124

---

**Principles for Quality Quantification.**

• Some hopefully deep and useful guidelines
  • to help you quantify quality ideas
0. THE PRINCIPLE OF 'BAD NUMBERS BEAT GOOD WORDS' (re-visited!)

- Poor quantification is more useful than none;
- at least it can be improved systematically.

State of the Art Flexibility
Enhanced Usability
Improved Performance

1. THE PRINCIPLE OF 'QUALITY QUANTIFICATION'

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

“If you think you know something about a subject, try to put a number on it. If you can, then maybe you know something about the subject. If you cannot then perhaps you should admit to yourself that your knowledge is of a meager and unsatisfactory kind.

Lord Kelvin, 1893
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

2. THE PRINCIPLE OF 'MANY SPLENDORED THINGS'

• Most quality ideas
  –are usefully broken into several measures of goodness.

Usability:

Entry Qualification: Scale IQ, .......
Learning Effort: Scale: Hours to learn, ..... 
Productivity: Scale: Tasks per hour, .......
Error Rate: Faults per 100 tasks, ..... 
Like-ability: % Users who like the system, ....
Quantifying Quality

Multiple Required Performance and Cost Attributes are the basis for architecture selection and evaluation

Resource
- Stakeholder A’s Financial Budget
- Stakeholder B’s Financial Budget

Performance
- [Operator]
- [Management]
- Usability
- Reliability
- Security
- Environment
- Innovation
- Cost Reduction
- Client Accounts

Elapse Time
100%
- 0%

Effort
100%
- 0%

Function
100%

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

Intelligibility
GIST: Super ease of immediate understand:
SCALE: % OK interpretations.
METER: 10 ops., 100 info, 15 mins.
P: PAST [20 ops., 300 info, 30 min.] 99%
RECORD [P] 99.0%
Fail [DELIVERY] 99.0% <MAB
[ACCEPTANCE] 99.5%
Goal [MH] 99.9% <LN
3. THE PRINCIPLE OF 'SCALAR DEFINITION'

- A *Scale of measure* is a powerful practical *definition* of a quality

**Flexibility:**

*Scale: Speed of Conversion to New Computer Platform*

---

(Quality) Requirements Specification Template with <hints>

**HOW WE SPECIFY SCALAR ATTRIBUTE PRIORITY**

- **Ambition:** <give overall real ambition level in 5-20 words>
- **Version:** <dd-mm-yy each requirements spec has a version, at least a date>
- **Owner:** <the person or instance allowed to make official changes to this requirement>
- **Type:** <quality|objective|constraint>
- **Stakeholder:** { , , } “who can influence your profit, success or failure?”
- **Scale:** <a defined units of measure, with [parameters] if you like>
- **Meter** [ <for what test level?> ]

---

====Benchmarks ============= the Past
Past [ ] <estimate of past> <--<source>
Record [ <where>, <when >, <estimate of record level> ] <-- <source of record data>
Trend [ <future date>, <where?> ] <prediction of level> <-- <source of prediction>

---

==== Targets ============= the future needs
Wish [ ] <-- <source of wish>
Goal [... ]<target level> <-- Source

Value [Goal] <refer to what this impacts or how much it creates of value>
Stretch [ ] <motivating ambition level> <-- <source of level>

---

Constraints """"------------------'
Fail [ ] <-- <source> ‘Failure Point’
Survival [ ] <-- <source of limit> ‘Survival Point’
4. THE PRINCIPLE OF 'THREATS ARE MEASURABLE'

• If lack of quality can destroy your project
• then you can measure it sometime;
• the only discussion will be 'how early?'.

5. THE PRINCIPLE OF 'LIMITS TO DETAIL'

• There is a practical limit to the number of facets of quality you can define and control,
• which is far less than the number of facets that you can imagine might be relevant.
6. THE PRINCIPLE OF 'METERS MATTER'

Practical measuring instruments improve the practical understanding and application of ‘Scales of measure’.

Portability:
Scale: Cost to convert/Module

| Meter [Data] measure/1,000 words converted |
| Meter [Logic] measure/1,000 Function Points Converted |

7. THE PRINCIPLE OF 'HORSES FOR COURSES'

Different quality-Scale measuring processes will be necessary for different points in time, different events and different places.

Availability:
Scale: % Uptime for System

| Meter [USA, 2001] Test X |
| Meter [UK, 2002] Test Y |
8. THE PRINCIPLE OF 'BENCHMARKS'

Past history and future trends *help* define words like "improve" and "reduce".

<table>
<thead>
<tr>
<th>Reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scale: Mean Time To Failure</td>
</tr>
<tr>
<td><strong>Past</strong> [US DoD, 2002] 30,000 Hours</td>
</tr>
<tr>
<td><strong>Trend</strong> [Nato Allies, 2003] 50,000 Hours</td>
</tr>
<tr>
<td><strong>Goal</strong> [UK MOD, 2005] 60,000 Hours</td>
</tr>
</tbody>
</table>

9. THE PRINCIPLE OF 'NUMERIC FUTURE'

Numeric future requirement levels *complete* the quality definition of relative terms like 'improved'.

<table>
<thead>
<tr>
<th>Usability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scale: Time to learn average task.</td>
</tr>
<tr>
<td><strong>Past</strong> [Old product, 2003] 20 minutes</td>
</tr>
<tr>
<td><strong>Wish</strong> [New product, 2007] 1 minute</td>
</tr>
<tr>
<td><strong>Stretch</strong> [End 2008, Students] 2 minutes</td>
</tr>
<tr>
<td><strong>Goal</strong> [End 2005, Teachers] 5 minutes</td>
</tr>
</tbody>
</table>
Some Planguage ‘Quality Quantification’ Concepts

PAST: any useful reference point. Your old product, a competitors organization, a quality achieved in same discipline but different branch of business.

RECORD: best in some class. State of the art. Something to beat. A challenge for you. An extreme PAST.

TREND: a future guess based on the PAST.

Survival: a level needed for survival of the entire system.

Goal: the level needed for satisfaction, happiness, joy and 100% full payment!

Wish: a level desired by someone, but which might not be feasible. Project is not committed to it.

A Corporate Quality Policy (Euro Multinational)

1. QUANTIFY QUALITY
2. CONTROL MULTIPLE DIMENSIONS
3. EVALUATE RISK
4. CONFIGURATION MANAGEMENT - TRACEABILITY
5. DOCUMENT QUALITY EVALUATION
6. EVOLUTIONARY DELIVERY CONTROL
7. CONTINUOUS WORK PROCESS IMPROVEMENT
Policy on QUANTIFICATION, CLARIFICATION AND TESTABILITY OF CRITICAL OBJECTIVES:

“All critical factors or objectives (quality, benefit, resource) for any activity (planning, engineering, management) shall be expressed clearly, measurably, testably and unambiguously at all stages of consideration, presentation, evaluation, construction and validation. “

<- (Quality Manual Source is) 5.2.2, 4.1.2, 4.1.5, 5.1.1, 6.1, 6.4.1, 7.1.1, 7.3 and many others.

Einstein on Stretching

• “One should not pursue goals that are easily achieved.
• One must develop an instinct for what one can just barely achieve through one’s greatest efforts.” (1915)

“We have to do the best we can.
This is our sacred human responsibility” (1940)
Supporting Standards for Quality Quantification

These following slides contain supporting Standards in detail which I do not expect to have time to show in my lecture
A Process for Quality Quantification. (PROCESS.QQ)

ENTRY: (ENTRY.QQ)

1. Do not enter if company files or standards already have adequate quantification devices.
   – Use existing quantification SCALES and METERS preferably.

2. Enter only if your process input documents
   – (contracts, marketing plans, product plans, requirements specification for example)
   – are Quality Controlled,
   – and have exited at a known and acceptable standard of defect-freeness
     • (default standard; less than 1 Major defect/page estimated remaining).
Procedure for the Quality Quantification Task (PROCEDURE.QQ)

NOTE: these following steps cannot be simply sequentially. They need to be repeated many times to evolve realistic quality quantifications.

1. Use applicable rules (RULES.GR, RULES.QR, RULES.QQ)

2. Build a list of all quality concerns from your process input documents. Include implicit quality requirements derived from design requirements. Include any recent practical experience such as from evolutionary steps of this project, pilot experiences or prototypes.

3. Detail the specification to a useful level. Include any recent practical experience such as from evolutionary result delivery steps of this project.

4. Revise these specifications when some design engineering/planning work is done on their basis. Only through design work can you know about the available technology and its costs.

5. Perform Quality Control (Inspection method) calculating remaining Major defects per page for the exit control. Apply valid rules (RULES.GR, RULES.QR, RULES.QQ)

6. Get experience using these specifications and revise specifications to be more realistic.

7. Repeat this process until you are satisfied with the result.

8. Cumulate your improved idea experiences and make available to others.

EXIT: (EXIT.QQ)

1. Calculated remaining Major defects/page less than 1.

2. or exit condition “1.” above is waived with the intent of getting experience or opinions so as to refine it for official exit and more-serious use.
Specific Rules for Quality Quantification (QQ)

4.3. Rules: Quality Quantification. (RULES.QQ)

The following rules would be
– appropriate for a culture which was intent on raising quality specifications to a high level
– and to systematically learn as a group,
– in the long term,
– from the experiences of themselves and others.

The rules are guidance to the any writer or maintainer of quality specifications.

Violations of these rules would be classed as 'defects' in a quality control process on the document.

Rules for Quality Quantification:(RULES.QQ) 1of2

0:RULES: Rules for technical specification (RULES.GR) apply. This may be used in addition to the Quality Requirement Specification Rules (RULES.QR) or whenever serious emphasis on quality definition is required.

1:STANDARD: The Scale shall wherever possible be derived from a standard SCALE (in named files or referenced sources) and the standard shall be source referenced (<>) in the specification.

2:SCALENOTE: If the Scale is not standard, a notification to Scale owner will inform about this case. "Note sent to <owner>" will be included as comment to confirm this act.

3:RICH: Where appropriate, a quality concept will be specified with the aid of multiple Scale definitions, each with their own unique tag, and appropriate set of defining parameters.

4: Meter: a practical and economic Meter or set of Meters will be specified for each Scale. Preference will be given to previously defined Meters in our Quantification archives.

5: Meter. NOTE: When 'essentially new' (no reference to previous case in generic archives) Meter specifications are made a Notification to Meter owner will notify about this case. "Note sent to <owner>" will be included as comment.
Rules for Quality Quantification:(RULES.QQ) 2of2

6:BENCHMARK: Reasonable attempt to establish 'baselines' (Past, Record, Trend) will be made for our system’s past, and for relevant competition.

7:TERMS: Future-priority requirements (Fail, Goal) will be made with regard to both long and short term.

8:DIFFERENTIATE: A distinction will be made, using qualifiers, between those system components which must have significantly higher quality levels than others, and components which do not require such levels. “The best can cost too much”.

9:SOURCE: Emphasis will be placed on giving the exact and detailed source (even if a personal guess) of all numeric specifications, and of any other specification which is derived from a process input document (like a Meter which is contractually defined).

10:UNCERTAINTY) Whenever numbers are uncertain, we will have rich annotation about the degree (plus/minus) and reason (a comment like “because contract & supplier not determined yet”). The reader shall not be left to guess or remember what is known, or could be known, with reasonable inquiry by the author.

Generic Rules for Technical Specification (including Quality Quantification) GR
0.3. Rules/Forms/Standards: Generic Rules and Requirements Rules sample.

- Here are some formal rules which could serve as a standard for how to communicate such ideas.
- We call this standard ‘Generic’ because it applies to many types of specification.
- ‘Rules’ are a ‘best practice’ procedure for writing a document. Violation of rules constitutes a formal ‘defect’ in that document.
- Rules are the local law of practice, and violation of them is an ‘illegal’ act.

**GENERIC RULES FOR TECHNICAL AND MANAGEMENT DOCUMENTATION**

Tag: RULES.GR

1: CLEAR Statements should be clear and unambiguous to their intended reader.
2: SIMPLE: Statements should be written in their most elementary form.
3: TAG. Statements shall have a unique identification tag.
4: SOURCE: Statements shall contain information about their detailed source, AUTHORITY and REASON/Rationale.
5: GIST: Complex statements should be summarized by a GIST statement.
6: QUALIFY: When any statement depends on a specific time, place or event being in force then this shall be specified by means of the [qualifier square brackets].
7: FUZZY: When any element of a statement is unclear then it shall be marked, for later clarification, by the <fuzzy angle brackets>.
8: COMMENT: any text which is secondary to a specification, and where no defect could result in a costly problem later, shall be written in italic text statements, or/and headed by suitable warning (NOTE, RATIONALE, COMMENT) or moved to footnotes. Non-commentary specification shall be in plain text. Italic can be used for emphasis of single terms in non-commentary statements. Readers shall be able to visually distinguish critical from not critical specification.
9: UNIQUE: requirements and design specifications shall be made one single time only. Then they shall be re-used by cross reference to their identity tag. Duplication is strongly discouraged.
In addition to the general rules, we can specify some special rules for the specific types of statement we are dealing with.

For example SR (below), QQ (above), QR (above).

REQUIREMENTS SPECIFICATION RULES.

SPECIFIC RULES: SR

- 0:GR-BASE: The generic rules (RULES.GR) are assumed to be at the base of these rules.
- 1:TESTABLE: The requirement must be specified so that it is possible to define an unambiguous test to prove that it is later implemented.
- 2:METER: Any test of SCALE level, or proposed tests, may be specified after the parameter METER.
- 3:SCALE: Any requirement which is capable of numeric specification shall define a numeric scale fully and unambiguously, or reference such a definition.
- 4:MEET: The numeric level needed to meet requirements fully shall be specified in terms of one or more [qualifier defined] target level {PLAN, MUST, WISH} goals; mainly the PLAN level here.
- 5:FAIL: The minimum numeric levels to avoid system, political, or economic failure shall be specified in terms of one or more [qualifier defined] ‘MUST’ level goals.
- 6. QUALIFY. Rich use of [qualifiers] shall specify [when, where, special conditions].
Tom’s New Book, Summer 2005

• Tom Gilb,
  – ISBN 0750665076 Publisher:
  – Elsevier Butterworth-Heinemann
  – Ask me for a free electronic copy!
  – Ordering details at www.gilb.com

  – Norsk Bokhandler: Ravenholm
    hanne@ravenholm.no, 48 05 55 00