

Quantifying Quality:

How to Tackle Quantification of the Critical Quality aspects for Projects for Both Requirements and Designs



Ask me for a free electronic copy!
(by email or memory stick here)

Tom@Gilb.com

www.Gilb.com

Result Planning Limited

Norway/UK

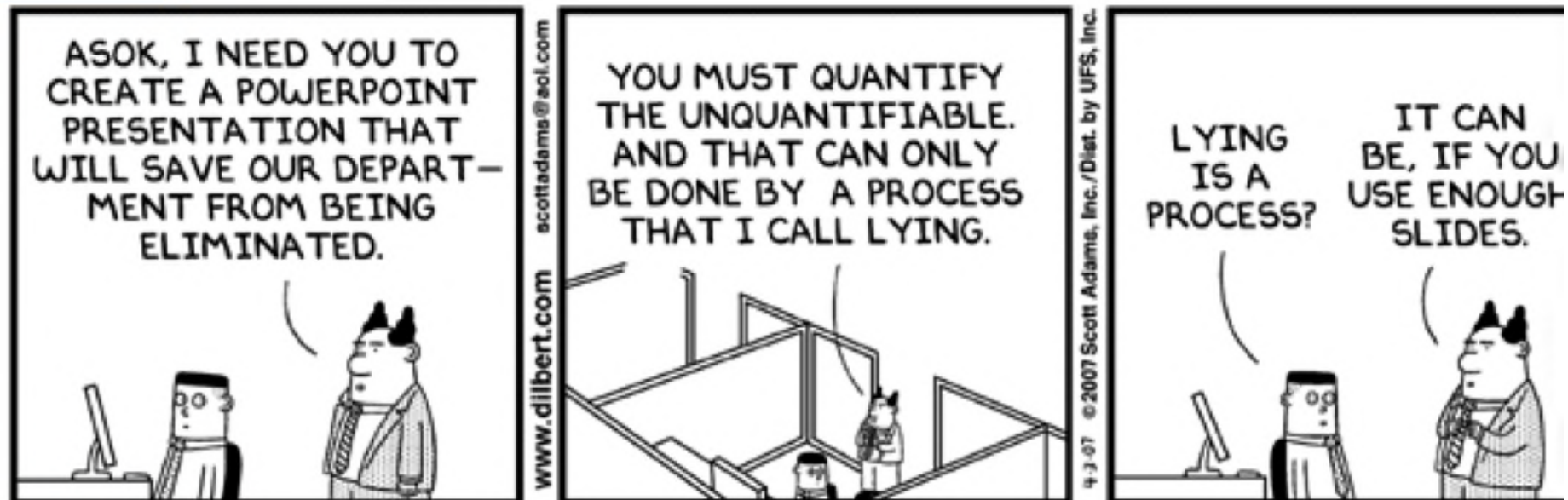
Updated October 4 2007

1:30 to 3:30 Middlesex University 18th October 2007

www.Gilb.com Slid

Quantifying the Unquantifiable

- Is this proof that Scott Adams works at one of my clients?



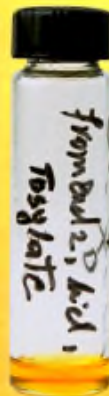
© Scott Adams, Inc./Dist. by UFS, Inc.

A Real Requirement: A Sample page Marked By Checker 2 General Rules = 153 majors/Page density

Sample 1



Sample 2



1.1 Look and feel

1.1.1 No knowledge of operating system

- The user will not be required to have ANY operating system knowledge or Unix skills to operate the system e.g. no Unix command line work, no terminal windows.

1.1.2 MS-Windows look and feel

- The system will have the look and feel of a Windows product.

1.1.3 MS-Windows concepts

- The system will make full use of the MS-Windows user-interface concepts such as Wizards to lead the user through user-defined parameters.

1.1.4 Hot keys

- The user will be able to use hot-keys for functionality. The user will be able to customize the hot-key assignment.

<- See rewriting of this on last slide

2 General Tools: Visualization (1.5) m -

The section will cover:

- Geographic display (2.1)
- Data viewer (Error! Reference source not found.)
- 2D graphs (Error! Reference source not found.)
- Distribution graphs (Error! Reference source not found.)

2.1 Geographic Display Area (1.5) m -

- The display area will be used for display of:
 - Reference information, such as raster and vector imagery
 - Plan information, such as locations of points, lines, exclusion areas etc.
 - Real-time information, such as vibrator status, or spread status

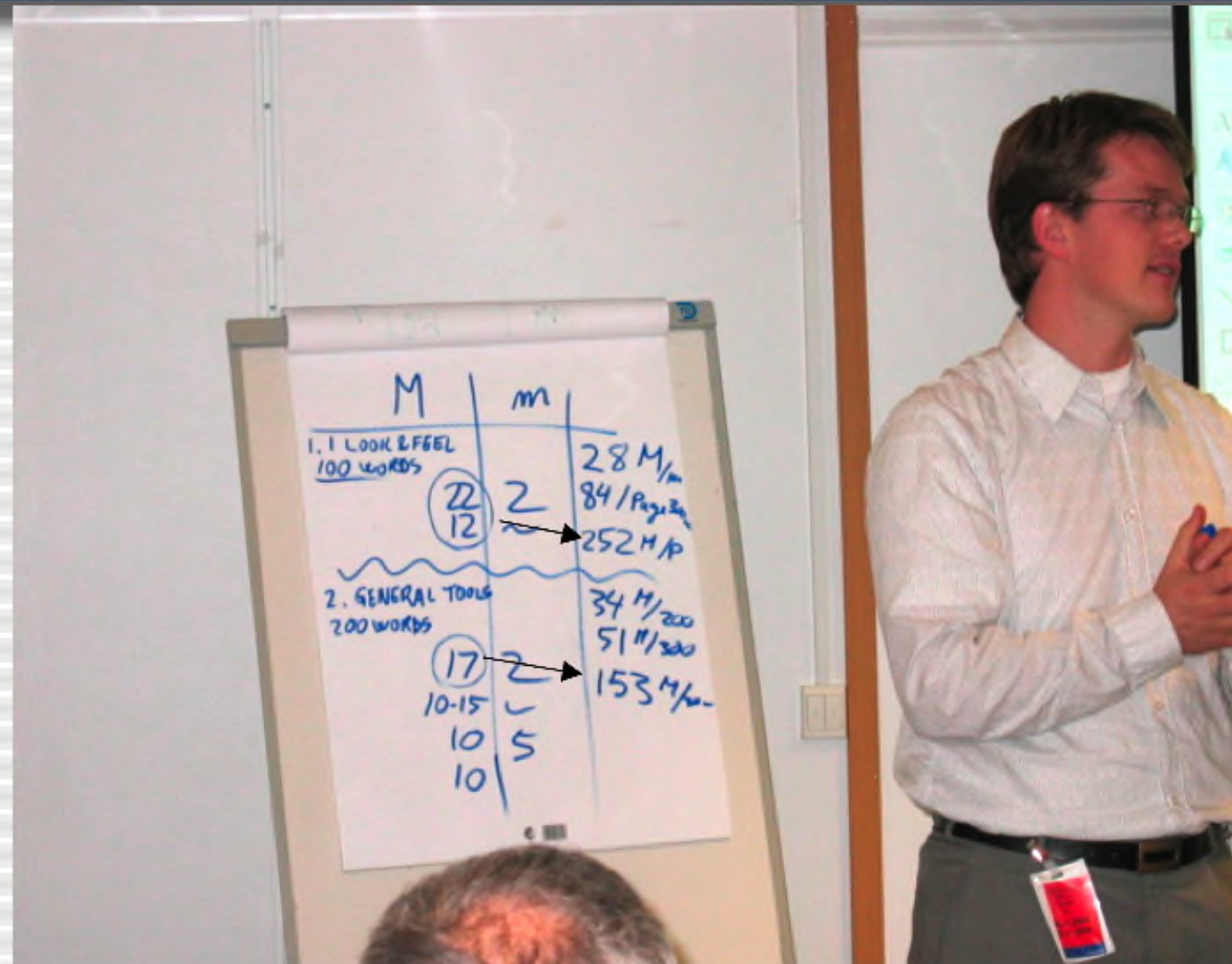
2.1.1 Geographic Display start-up (1) m -

- Upon starting up the display the mapped area will correspond to the one displayed upon last use, or the entire project area if no previous area default is available.
- Upon starting up the display the layer settings (which layers are visible, annotation setting etc.) will be retained.

2.1.2 Layer management (1.5) m -

- It will be quick and simple to add and remove all types of information on the display area. This display information will be based on layers.

Sample Major Defect --> Extrapolations Done
= 153 Majors/Page and 252 Majors/Page
from Samples of Real requirements
determination done by responsible managers, 2004



Rewrite of a real Defective 'Requirement at (Norway, 2004)

- 1.1.3 MS-Windows concepts
- The system will make full use of the MS-Windows user-interface concepts such as Wizards to lead the user through user-defined parameters.

False
Requirement
(a solution)



'Means'
not
'Ends'



Solutions (Designs):

The system will make full use of the MS-Windows user-interface concepts.
Examples: such as Wizards to lead the user through user-defined parameters.

Why? Lots of users ask for it. (MS-Windows)
Why? Easy to use. / Intuitive

Usability (intuitiveness, learn, training, mistakes)

Usability.Intuitive

Ambition: after initial training, (one week course, two week field) the user shall not have to ref the user manual.

Scale: % of defined [Elements] done Correctly, by defined [User], within <5> seconds.

Correctly: defined as: the System responded in a way the user thought the system should do.

System: Defined as: xxx

Record [ISX Sierra, 1994] 95%±5% <- Boss "as perc

Record [Product = 408] ??%

Past [Elements = Finding a menu option, User = Beginner, 2004] 70%±10% <- the team

Tolerable

Goal [Elements = Finding a menu option, User = Beginner, March 15th 2007] 70%±10% <- the team

Goal [Elements = Finding a menu option, User = Beginner, March 15th 2008, at Commercialization] 90%±5 <- the team

Analysis

The 'Real'

Requirement
in Planguag

Sample Requirement Rewrites

Overview of Requirement Types

- **High-Level Requirements**
 - 1. Introduction
 - 2. Business requirements
 - 2.1. Time to market
 - 2.2. Cost
 - 2.2.1. Capital investment
 - 2.2.2. Operational cost
 - 2.2.3. Support and maintenance cost
 - 2.3. Market constraints
 - 2.4. Trade Compliancy
 - 2.5. Environmental compliancy
 - 3. Functional requirements
 - 3.1. Recording
 - 3.2. Integration
 - 3.3. Sources
 - 3.4. Use-case xxx
- 4. Quality requirements
 - 4.1. Availability
 - 4.1.1. Reliability
 - 4.1.2. Recoverability
 - 4.1.3. Integrity
 - 4.2. Usability
 - 4.2.1. Learn-ability
 - 4.2.2. Like-ability
 - 4.2.3. User Productivity
 - 4.2.4. Intuitiveness
 - 4.2.5. Intelligibility
 - 4.3. Adaptability
 - 4.3.1. Flexibility
 - 4.3.2. Upgradeability
 - 4.4. Performance/Productivity
 - 4.5. Capacity
 - 4.6. Security

Example: Operator Usability

4.2. Usability

4.2.1. Learn-ability

4.2.2. Like-ability

4.2.3. User Productivity

ID	7	Title Faster spread layout handling				
Priority	1	Status	Open	Version	0.5	
Category	Usability/User Productivity		Type	Quality Requirement		
Date submitted	28.09.2004		Last Update	3 Feb 2005		
Reporter	S		Assigned to			
Stakeholders						
Ambition	Reduce operator time by at least factor 2, when laying out the spread: cables and connection					
Justification	Business Economics, specifically <Operational Cost, system efficiency>					
Scale	Average Time for defined [Crews {Layout Crew, Pickup Crew}] of defined [Crew Size] with a defined [Spread Configuration] per [1,000-Sensors], to successfully complete defined [Layout Work {Initial Layout, Layout Rolling}].					
Meter	Real field trial and operational data manually collected					
Goal	[1 st Release, Layout Crew, 5,000 Sensors, Desert, Crew Size = 10, Initial Layout] X/2 hour?					
Past	[2004, Layout Crew, 5,000 Sensors, Desert, Crew Size = 10] X hour?					
Links	req 2.5.3					

Example: Crew Usability

ID	8	Title	Reduced battery handling			
Priority	1	Status	Open	Version	0.5	
Category	Usability/User Productivity		Type	Quality Requirement		
Date submitted	28.09.2004		Last Update	3 Feb 2005		
Reporter	S: _____		Assigned to			
Stakeholders	Battery Handling Crew					
Ambition	reduce battery charging and replacement effort					
Comment	Assumption: The number of batteries will be reduced by reducing the power consumption per channel (This is a solution <-BN)					
Scale	Effort-hours per day for Battery Handling {Charging and Replacement}.					
Meter	Manual logs observing real operations.					
Goal	[]X/2?					
Past	[] X					
Links	req 2.5.4, supported by requirement 25Battery Power Consumption					

ID	20	Title	System Overhead Time: Note, name title needs reworking to reflect content) <- BN			
Priority	1	Status	Open	Version	0.51	
Category	Availability/Recoverability		Type	Quality Requirement		
Date submitted	28.09.2004		Last Update	3.2.2005		
Reporter	Stu		Assigned to	Tho		
Stakeholders	Field Operations (all levels).					
Ambition	"The system must be capable of passing uninterrupted seismic data from the full channel count (100,000 minimum live channels), plus any display information required, control information flow, QC information required, plus routing all data from any single broken link without significant time overhead " <- Stu					
Comment						
Scale	Time in seconds from when a Single Failure occurs, until Full Recovery achieved. Single Failure: defined as: broken link, or broken transport network node, Full Recovery: defined as: system is Operational again, and no data is lost. Operational: defined as: The network integrity and bandwidth is restored. <i>Note 1: this includes the time to pass uninterrupted seismic data from the full channel count (100,000 minimum live channels), plus any display information required, control information flow, QC information required, plus routing all data from any single broken link.</i> <i>Note 2: exceptions, short circuit? – cost implications, under investigation. <- Tho</i>					
Meter	Gist: Measure from <Single Failure occurred> to <Full Recovery>. Description: A set of artificial Single Failures is injected as a test, and time is measured until Full Recovery, using built in measure. Issue: is this already built in or do we have to plan a design to build it in – the seconds measure to recovery.					
Goal	[First Version] < 0.5 seconds ?? <- Tho He says 'closer to 10 seconds'					
Past	About 10 to 60 minutes?? "The old system does not have rapid automatic recovery. Manual fix". <-BN					
Links	req 5.3					

Scale Def
on next sl

real case

Detail of Scale for 'System Overhead Time' requirement

Scale	Time in seconds from when a Single Failure occurs, until Full Recovery is achieved. Single Failure: defined as: broken link, or broken transport network node. Full Recovery: defined as: system is Operational again, and no data is lost. Operational: defined as: The network integrity and bandwidth is restored. <i>Note 1: this includes the time to pass uninterrupted seismic data from the full channel count (100,000 minimum live channels), plus any display information required, control information flow, QC information required, plus routing all data from any single broken link.</i> <i>Note 2: exceptions, short circuit? – cost implications, under investigation.</i> <u>T</u>
--------------	---

Priority	1	Status	Open	Version	0.5
Category	Availability.Recoverability		Type	Quality requirement	
Date submitted	3.2.2005		Last Update	3.Feb.2005	
Reporter	Bj		Assigned to	yyy	
Stakeholders	Field Operations				
Ambition	Substantial reduction in component recovery speed				
Scale	Mean time in minutes to recover a defined [Sub-System] from a Failed State until the Sub-system is in a defined [State]: default Locally Fixed. State: {Failed, Locally Fixed, Repositioned}.				
Meter	Manual calculation from Introspection statistics				
Goal	[Whole System] 30 minutes? <- BN [Sub-system = Central System Software, 1 st Release] 5 minutes? <- BN [Central System Hardware, 1 st Release] 10 min.? <-BN [Sensor Network] 60 mins. ? [Transport Network] 60 mins. ? [Operators] 10 mins. ? [Power Supply] ? [All Other Components] ? <what else is there? Trucks?, Air Conditioning>				
Past	[Whole System] [Central System Software, 2004] 1? <- 2004 field observation? [Central System Hardware, 2004] ? [Sensor Network] ? [Transport Network] ? [Operators] ? [Power Supply] ? [All Other Components] <what else is there? Trucks?, Air Conditioning>				
Justification	Business productivity				
Definitions	Whole System: defined as: {Central Software System, Central hardware System, Sensor Network, Transport network, Operators, Power Supply, All Other Components}.				

real case

Requirement t: Recoverability

- Notice:
 - multiple Goal Levels
 - Parameterized Scale

4.1.1. Readiness

real case

ID	21	Title	System boot time			
Priority	1	Status	Open	Version	0.5	
Supports	Availability/Readiness		Type	Quality		
Date submitted	28.09.2004		Last Update	3.2.2005		
Reporter	St.		Assigned to			
Stakeholders	Field Operations					
Ambition	Substantially reduce the time from power is turned on, until ready for acquisition.					
Justification	More productive earning time. <refer to a higher level business objective>					
Scale	Maximum time from power is turned on to Ready For Acquisition. Ready For Acquisition: defined as: the system is completely ready to record data. The Master Display is fully on screen including GIS View Map, with Status information for all sensors and boxes. Assumption: the time to lay out the Spread is independent of this, and presumed completed by power on.					
Meter	Manual test and stopwatch recording.					
Goal	Goal1: [Spread] 3 minutes. Goal2: [Central System] 10 minutes					
Past	--- [Crew2, 2004] ~30 min. ?? <-BN					
Links	req ??					

Business Objective
TTM
Same Format

2.1. *Time to market*

ID	1	Title	Time to market			
Priority		1	Status	Open	Version	0.5
Category		Time to market		Type	Business requirement	
Date submitted		28.09.2004		Last Update	28.09.2004	
Reporter		S: 		Assigned to		
Stakeholders						
Description		It is expected that an average of 2 QX crews will be manufactured and deployed per year after 2007				
Scale		Point in time successful delivery to first customer				
Meter						
Goal		Goal1 [Q1 2007] 30000 live channel system earning revenue Goal2 [July 2007] 45000 live channel system earning revenue				
Past						
Links		req 2.7				

Template for Quality Requirements

Template for Quality Requirements:

ID	?	Title				
Priority	?	Status	Open	Version	0.5	
Category			Type	Quality Requirement		
Date submitted	x.x.2005		Last Update	X.X.2005		
Reporter	xxx		Assigned to	yyy		
Scope	<define what this applies to of operations or system components>					
Stakeholders	Zz, xx					
Ambition						
Scale						
Meter						
Goal						
Past						
Justification	<link to business requirements>					
Links						



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).



Phylolaus: Quantifying Sound Qualities

Below is the image in its original context on the page: www.philophony.com/sensprop/pythagor.html



< Pythagoras is here shown quantifying the weight of the bells, and glasses, plucking the monochord with measured weights, and arguing the finest points of dissonance [comparing flute lengths] with Philolaus

Clockwise from top left: the hammers in Jubal [Tubalcain] smithy, playing tuned bells and water filled cups, experimenting with weights on the end of fixed length strings, and on the length of pipes to determine the exact ratios of consonant sounds one to another [from F Gafurio Theorica Musice 1492] [rep. Wittkower 1949.]

"You can nearly measure everything,
but how can you measure style?"

That's Siemens catchphrase for its new S65

cnet CNETAsia Reviews

GO! Top 100 searches»

NEWS ENTERPRISE TECH BUILDER PRODUCT REVIEWS DOWNLOADS

GO TO: » SMB Disaster Recovery » GameSpotAsia » New: Digital Living WIN: » Nokia 7260 » Juniper RA-500 » iRiver MP3

GO DIRECTLY TO CNETASIA'S REVIEWS:

» HANDPHONES » DIGITALS » NOTEBOOKS » DESKTOPS » HANDHELD » PRINTERS » HOME A/V » VIDEO GAMES » PORTABLE A/V

Product Reviews » Handphones » Siemens S65

OVERVIEW REVIEW SPECS IMAGES USER OPS

Siemens S65

[Enlarge photo](#)

CNETASIA EDITORS' RATING Rating system explained

8.0
Very Good

Design	8
Features	8
Performance and Battery Life	8
Average	8.0

User recommendation:

User opinions: 79% 3 21% 3 24 votes

[Read user comments](#) | [Write your own review](#)

\$559.00 (US\$361.88)
for 2-year price plan

CNETASIA REVIEW

Reviewed By **Mark Tan**
(1/10/2004)

The good: Stylish shell; high resolution 1.3-megapixel camera; triband; Bluetooth; RS-MMC expansion slot.

The bad: Lacks MP3 player or FM radio; sluggish while performing certain operations.

The bottom line: Siemens' latest megapixel camera-phone is sophisticated-looking and has excellent features to match.

ADDITIONAL HANDPHONE COVERAGE

- » [Hottest phones](#)
- » [Upcoming phones](#)
- » [Service center locator](#)
- » [Latest reviews](#)
- » [Help & how-to](#)
- » [Buying adviser](#)
- » [Wireless Watch](#)

CNETASIA NEWSLETTERS

CLICK ON A TITLE BELOW TO LEARN MORE ABOUT IT.

- ☒ [Cool Gear Alert](#)
- ☒ [Hardware Alert](#)
- ☒ [Mobile Tech](#)
- ☒ [Spot-On Game Bulletin](#)

More newsletters

Your email here

SIGN UP NOW



How to Quantify any Qualitative Requirement

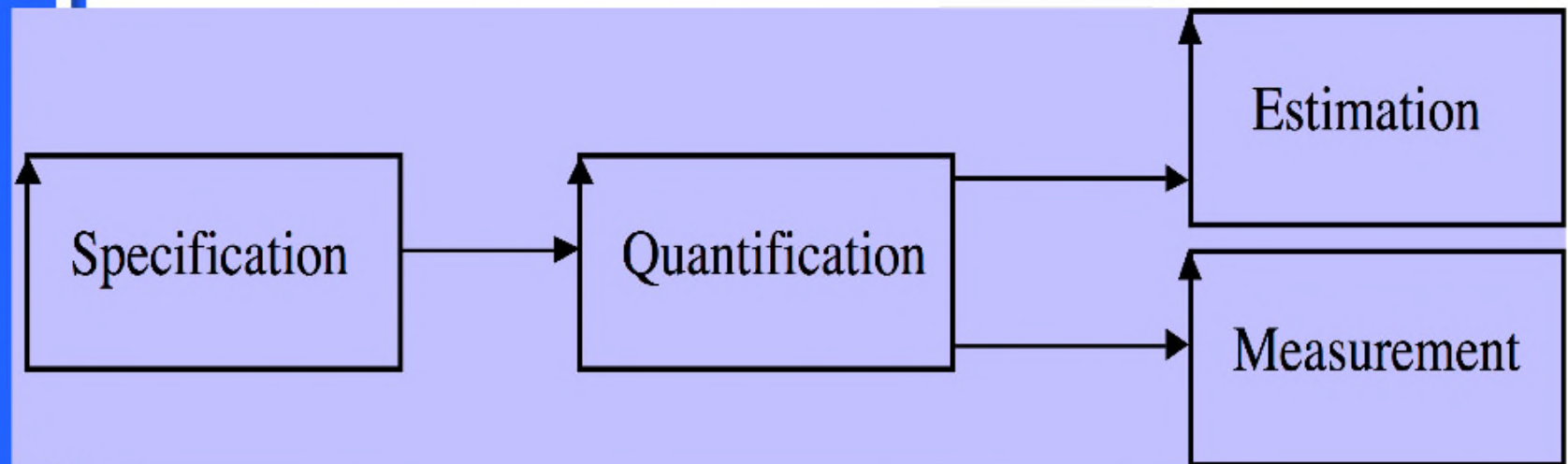


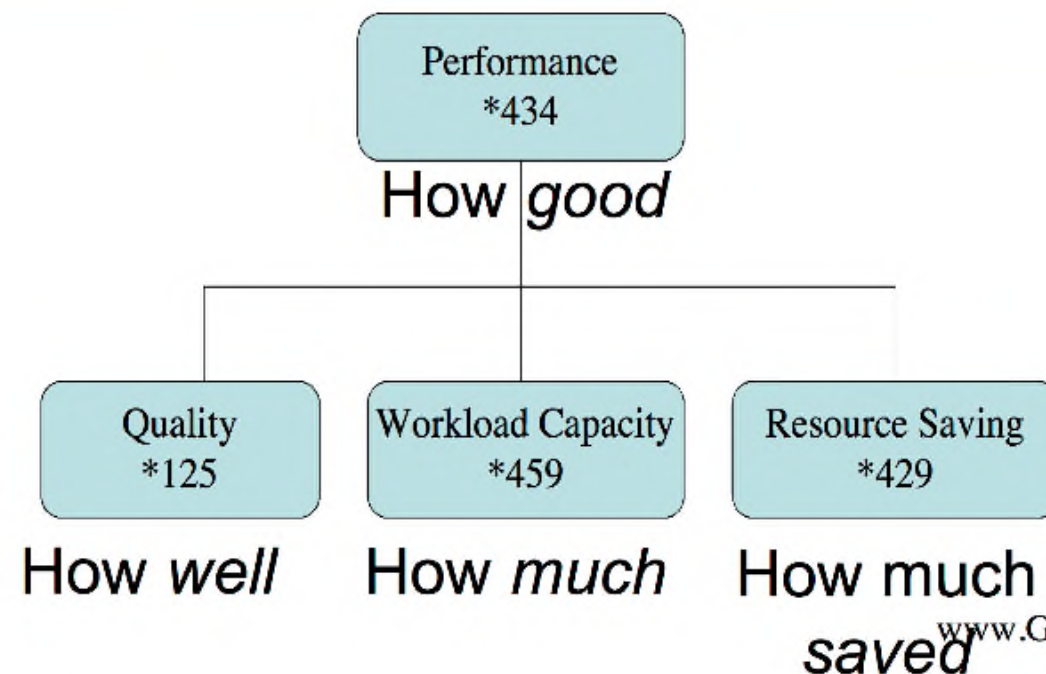
Diagram from 'Competitive Engineering.' book.

Quality: the concept, the noun: 'HOW WELL'

Planguage Concept *125, Version: March 20, 2003

A 'quality' is

- a scalar attribute -|-|-|-| (Scale symbol)
- reflecting 'how well' -----Past Level<----->
- a system functions. (Fn)-----Past Level<----->



Quantifying Quality

Quality is characterized by these 'QUANTIFICATION' principle.

Quality is characterized by these 'QUANTIFICATION' principle.

Quality describes 'how well' a function is done.

Quality is characterized by these '**QUANTIFICATION**' principle.

Quality describes 'how well' a function is done.

Quality describes the *partial effectiveness* of a function (as do all other performance attributes).

Quality is characterized by these '**QUANTIFICATION**' principle.

Quality describes 'how well' a function is done.

Quality describes the *partial effectiveness* of a function (as do all other performance attributes).

Quality is *valued* to *some* degree by *some* stakeholders of the system

Quality is characterized by these '**QUANTIFICATION**' principle.

Quality describes 'how well' a function is done.

Quality describes the *partial effectiveness* of a function (as do all other performance attributes).

Quality is *valued to some degree* by *some* stakeholders of the system

More quality is generally *valued* by stakeholders; especially if the increase is free, or lower cost, than the value of the increase.

Quality is characterized by these '**QUANTIFICATION**' principle.

Quality describes 'how well' a function is done.

Quality describes the *partial effectiveness* of a function (as do all other performance attributes).

Quality is *valued to some degree* by *some* stakeholders of the system

More quality is generally *valued* by stakeholders; especially if the increase is free, or lower cost, than the value of the increase.

Quality attributes can be *articulated* independently of the particular means (designs) used for reaching a specific quality level –

Quality is characterized by these '*QUANTIFICATION*' principle.

Quality describes 'how well' a function is done.

Quality describes the *partial effectiveness* of a function (as do all other performance attributes).

Quality is *valued to some* degree by *some* stakeholders of the system

More quality is generally *valued* by stakeholders; especially if the increase is free, or lower cost, than the value of the increase.

Quality attributes can be *articulated* independently of the particular means (designs) used for reaching a specific quality level –

even though all quality levels *depend* on the particular designs used to achieve them.

Quality is characterized by these '*QUANTIFICATION*' principle.

Quality describes 'how well' a function is done.

Quality describes the *partial effectiveness* of a function (as do all other performance attributes).

Quality is *valued* to *some* degree by *some* stakeholders of the system

More quality is generally *valued* by stakeholders; especially if the increase is free, or lower cost, than the value of the increase.

Quality attributes can be *articulated* independently of the particular means (designs) used for reaching a specific quality level –

even though all quality levels *depend* on the particular designs used to achieve them.

A particular quality can be described in terms of a *complex* concept, consisting of multiple elementary quality concepts.

Quality is characterized by these '**QUANTIFICATION**' principle.

Quality describes 'how well' a function is done.

Quality describes the *partial effectiveness* of a function (as do all other performance attributes).

Quality is *valued* to *some* degree by *some* stakeholders of the system

More quality is generally *valued* by stakeholders; especially if the increase is free, or lower cost, than the value of the increase.

Quality attributes can be *articulated* independently of the particular means (designs) used for reaching a specific quality level –

even though all quality levels *depend* on the particular designs used to achieve them.

A particular quality can be described in terms of a *complex* concept, consisting of multiple elementary quality concepts.

Quality is *variable* (along a definable scale of measure: as are all scalar attributes).

Quality is characterized by these '**QUANTIFICATION**' principle.

Quality describes 'how well' a function is done.

Quality describes the *partial effectiveness* of a function (as do all other performance attributes).

Quality is *valued* to *some* degree by *some* stakeholders of the system

More quality is generally *valued* by stakeholders; especially if the increase is free, or lower cost, than the value of the increase.

Quality attributes can be *articulated* independently of the particular means (designs) used for reaching a specific quality level –

even though all quality levels *depend* on the particular designs used to achieve them.

A particular quality can be described in terms of a *complex* concept, consisting of multiple elementary quality concepts.

Quality is *variable* (along a definable scale of measure: as are all scalar attributes).

Quality levels are capable of being specified *quantitatively* (as are all scalar attributes).

Quality is characterized by these '**QUANTIFICATION**' principle.

Quality describes 'how well' a function is done.

Quality describes the *partial effectiveness* of a function (as do all other performance attributes).

Quality is *valued* to *some* degree by *some* stakeholders of the system

More quality is generally *valued* by stakeholders; especially if the increase is free, or lower cost, than the value of the increase.

Quality attributes can be *articulated* independently of the particular means (designs) used for reaching a specific quality level –

even though all quality levels *depend* on the particular designs used to achieve them.

A particular quality can be described in terms of a *complex* concept, consisting of multiple elementary quality concepts.

Quality is *variable* (along a definable scale of measure: as are all scalar attributes).

Quality levels are capable of being specified *quantitatively* (as are all scalar attributes).

Quality levels can be *measured* in practice.

Quality is characterized by these '**QUANTIFICATION**' principle.

Quality describes 'how well' a function is done.

Quality describes the *partial effectiveness* of a function (as do all other performance attributes).

Quality is *valued to some degree* by *some* stakeholders of the system

More quality is generally *valued* by stakeholders; especially if the increase is free, or lower cost, than the value of the increase.

Quality attributes can be *articulated* independently of the particular means (designs) used for reaching a specific quality level –

even though all quality levels *depend* on the particular designs used to achieve them.

A particular quality can be described in terms of a *complex* concept, consisting of multiple elementary quality concepts.

Quality is *variable* (along a definable scale of measure: as are all scalar attributes).

Quality levels are capable of being specified *quantitatively* (as are all scalar attributes).

Quality levels can be *measured* in practice.

Quality levels can be traded off to some degree; with other system attributes valued more by stakeholders.

Quality is characterized by these '**QUANTIFICATION**' principle.

Quality describes 'how well' a function is done.

Quality describes the *partial effectiveness* of a function (as do all other performance attributes).

Quality is *valued to some degree* by *some* stakeholders of the system

More quality is generally *valued* by stakeholders; especially if the increase is free, or lower cost, than the value of the increase.

Quality attributes can be *articulated* independently of the particular means (designs) used for reaching a specific quality level –

even though all quality levels *depend* on the particular designs used to achieve them.

A particular quality can be described in terms of a *complex* concept, consisting of multiple elementary quality concepts.

Quality is *variable* (along a definable scale of measure: as are all scalar attributes).

Quality levels are capable of being specified *quantitatively* (as are all scalar attributes).

Quality levels can be *measured* in practice.

Quality levels can be traded off to some degree; with other system attributes valued more by stakeholders.

Quality can never be perfect (100%), in the real world.

Quality is characterized by these '**QUANTIFICATION**' principle.

Quality describes 'how well' a function is done.

Quality describes the *partial effectiveness* of a function (as do all other performance attributes).

Quality is *valued* to *some* degree by *some* stakeholders of the system

More quality is generally *valued* by stakeholders; especially if the increase is free, or lower cost, than the value of the increase.

Quality attributes can be *articulated* independently of the particular means (designs) used for reaching a specific quality level –

even though all quality levels *depend* on the particular designs used to achieve them.

A particular quality can be described in terms of a *complex* concept, consisting of multiple elementary quality concepts.

Quality is *variable* (along a definable scale of measure: as are all scalar attributes).

Quality levels are capable of being specified *quantitatively* (as are all scalar attributes).

Quality levels can be *measured* in practice.

Quality levels can be traded off to some degree; with other system attributes valued more by stakeholders.

Quality can never be perfect (100%), in the real world.

There are some levels of a particular quality that may be outside the state of the art; at a defined time and circumstance.

Quality is characterized by these '**QUANTIFICATION**' principle.

Quality describes 'how well' a function is done.

Quality describes the *partial effectiveness* of a function (as do all other performance attributes).

Quality is *valued* to *some* degree by *some* stakeholders of the system

More quality is generally *valued* by stakeholders; especially if the increase is free, or lower cost, than the value of the increase.

Quality attributes can be *articulated* independently of the particular means (designs) used for reaching a specific quality level –

even though all quality levels *depend* on the particular designs used to achieve them.

A particular quality can be described in terms of a *complex* concept, consisting of multiple elementary quality concepts.

Quality is *variable* (along a definable scale of measure: as are all scalar attributes).

Quality levels are capable of being specified *quantitatively* (as are all scalar attributes).

Quality levels can be *measured* in practice.

Quality levels can be traded off to some degree; with other system attributes valued more by stakeholders.

Quality can never be perfect (100%), in the real world.

There are some levels of a particular quality that may be outside the state of the art; at a defined time and circumstance.

When quality levels increase towards perfection, the resources needed to support those levels tend towards infinity.

Quality is characterized by these '**QUANTIFICATION**' principle.

Quality describes 'how well' a function is done.

Quality describes the *partial effectiveness* of a function (as do all other performance attributes).

Quality is *valued* to *some* degree by *some* stakeholders of the system

More quality is generally *valued* by stakeholders; especially if the increase is free, or lower cost, than the value of the increase.

Quality attributes can be *articulated* independently of the particular means (designs) used for reaching a specific quality level –

even though all quality levels *depend* on the particular designs used to achieve them.

A particular quality can be described in terms of a *complex* concept, consisting of multiple elementary quality concepts.

Quality is *variable* (along a definable scale of measure: as are all scalar attributes).

Quality levels are capable of being specified *quantitatively* (as are all scalar attributes).

Quality levels can be *measured* in practice.

Quality levels can be traded off to some degree; with other system attributes valued more by stakeholders.

Quality can never be perfect (100%), in the real world.

There are some levels of a particular quality that may be outside the state of the art; at a defined time and circumstance.

When quality levels increase towards perfection, the resources needed to support those levels tend towards infinity.

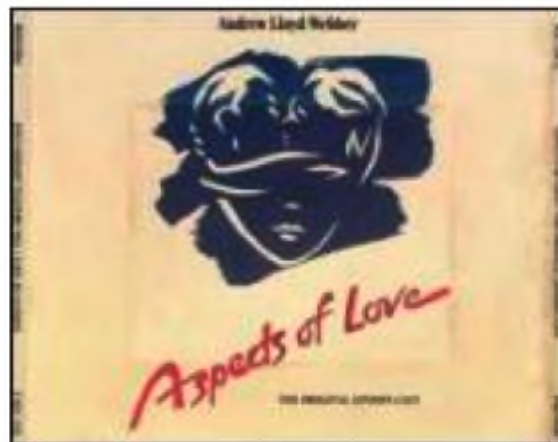
Exercise: Aspects of Love, or Love is a many splendored thing!

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

See note for Sutra

Love Attributes: Brainstormed By Dutch Engineers

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



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:
- 1. 'Truthfulness'
- 2. **Broken Agreements**
- 3. **Late Appointments**
- 4. **Late delivery**
- 5. **Gossiping to Others**

Camaraderie (Real Case UK)

Ambition: *to maintain an exceptionally high sense of good personal feelings and co-operation 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

Goal [Mid 2002] 10%, [End 2002] <1% <- R & CD

Rationale:

maintain staff number, and morale as core of business and business predictability for customers.

Love: Biblical Dimensions <- Lawrence Day, Boein

**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:**



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)
rejoiceth in the truth
11. Beareth all things
12. believeth all things
13. hopeth all things
14. endureth all things
15. never faileth

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

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

- **Evaluation** solutions/designs/architectures against the quantified quality requirements (Impact Estimation)

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)

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

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)

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)

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

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

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)

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.

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

Real Examples of Requirements (Oct 2004)

37 Page Detailed 'Functional' Requirement

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.

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



Future Information Research Management

Tom & Kai Gilb version of Trond Johansen's Presentation

Trond Johansen, QA & Process Manager, Firm AS

Trond.Johansen@firmglobal.com

From Waterfall to Evo firm

Future Information Research Management

Tom & Kai Gilb version of Trond Johansen's Presentation

Trond Johansen, QA & Process Manager, Firm AS

Trond.Johansen@firmglobal.com

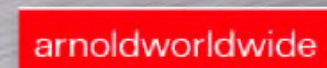
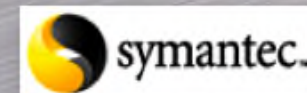


• FIRM R&D department •

- Developers (13)
- Management/(CSO) (2)
- Tech Support NY (1)
- Microsoft .NET framework, SQL
- SEPG group (3) with responsibility for process improvement and quality assurance (CMMI)
 - Configuration Management, setup ++
 - Testing
 - Software Process Improvement (SPI)



Customer Successes in Corporate Sector



Basis for the 3 month trial period

- a **one day crash course** with Tom and Kai Gilb
- and a **literature study**:
 - “Competitive Engineering” by Tom Gilb,
 - “Evo” by Kai Gilb,
 - and other material on the subject.



● **FIRM's interpretation of Evo Method** ●

our overall understanding of EVO was this:

● **FIRM's interpretation of Evo Method** ●

our overall understanding of EVO was this:

- **Find stakeholders** (End users, super-users, support, sales, IT Operations etc)

● **FIRM's interpretation of Evo Method** ●

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**

● **FIRM's interpretation of Evo Method** ●

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)

● **FIRM's interpretation of Evo Method** ●

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

● **FIRM's interpretation of Evo Method** ●

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

● **FIRM's interpretation of Evo Method** ●

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

● **FIRM's interpretation of Evo Method** ●

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:

● **FIRM's interpretation of Evo Method** ●

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

● **FIRM's interpretation of Evo Method** ●

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?

• Paradigm Shift •

With EVO, our requirements process changed.

- **Previously** we focused mostly on **function requirements**.
- We realized that **it's the product *quality* requirements that really separate us from our competitors.**



Our new • Requirement • 'standards'

We tried to **define**
our **requirements**
according to a **basic**
standard given by
Gilb

☒ Clear & Unambiguous

☒ Testable

☒ Quantified

☒ No Solutions (Designs)

☒ Stakeholder Focus

• Real Requirements Example •

- 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 Conformat.

• Real Requirements Example •

Usability.Productivity (*taken from Conformat 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)

• Real Requirements Example •

Usability.Productivity (*taken from Conformat 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)

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)



IET, project step planning and accounting: using an Impact Estimation Table

- IET for MR Project – Conformat 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)

	A	B	C	D	E	F	G	BX	BY	BZ	CA
1											
2		Current Status	Improvements		Goals			Step9			
3								Recoding			
4								Estimated impact		Actual impact	
5		Units	Units	%	Past	Tolerable	Goal	Units	%	Units	%
6					Usability.Replacability (feature count)						
7		1,00	1,0	50,0	2	1	0				
8					Usability.Speed.NewFeaturesImpact (%)						
9		5,00	5,0	100,0	0	15	5				
10		10,00	10,0	200,0	0	15	5				
11		0,00	0,0	0,0	0	30	10				
12					Usability.Intuitiveness (%)						
13		0,00	0,0	0,0	0	60	80				
14					Usability.Productivity (minutes)						
15		20,00	45,0	112,5	65	35	25	20,00	50,00	38,00	95
20					Development resources						
			101,0	91,8	0		110	4,00	3,64	4,00	3

IET, 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)

	A	B	C	D	E	F	G	BX	BY	BZ	CA
1											
2		Current Status	Improvements		Goals			Step9			
3								Recoding			
4								Estimated impact		Actual impact	
5		Units	Units	%	Past	Tolerable	Goal	Units	%	Units	%
6					Usability.Replacability (feature count)						
7		1,00	1,0	50,0	2	1	0				
8					Usability.Speed.NewFeaturesImpact (%)						
9		5,00	5,0	100,0	0	15	5				
10		10,00	10,0	200,0	0	15	5				
11		0,00	0,0	0,0	0	30	10				
12					Usability.Intuitiveness (%)						
13		0,00	0,0	0,0	0	60	80				
14					Usability.Productivity (minutes)						
15		20,00	45,0	112,5	65	35	25	20,00	50,00	38,00	95
20					Development resources						
			101,0	91,8	0		110	4,00	3,64	4,00	3

EVO Plan Confirmit 8.5

4 more product areas were attacked concurrently

Impact Estimation Table: Reportal codename "Hyggen"

Current Status	Improvements		Reportal - E-SAT features		
	Units	%	Past	Tolerable	Goal
			Usability.Intuitivness (%)		
75,0	25,0	62,5	50	75	90
			Usability.Consistency.Visual (Elements)		
14,0	14,0	100,0	0	11	14
			Usability.Consistency.Interaction (Components)		
15,0	15,0	107,1	0	11	14
			Usability.Productivity (minutes)		
5,0	75,0	96,2	80	5	2
5,0	45,0	95,7	50	5	1
			Usability.Flexibility.OfflineReport.ExportFormats		
3,0	2,0	66,7	1	3	4
			Usability.Robustness (errors)		
1,0	22,0	95,7	7	1	0
			Usability.Replacability (nr of features)		
4,0	5,0	100,0	8	5	3
			Usability.ResponseTime.ExportReport (minutes)		
1,0	12,0	150,0	13	13	5
			Usability.ResponseTime.ViewReport (seconds)		
1,0	14,0	100,0	15	3	1
			Development resources		
203,0			0		191

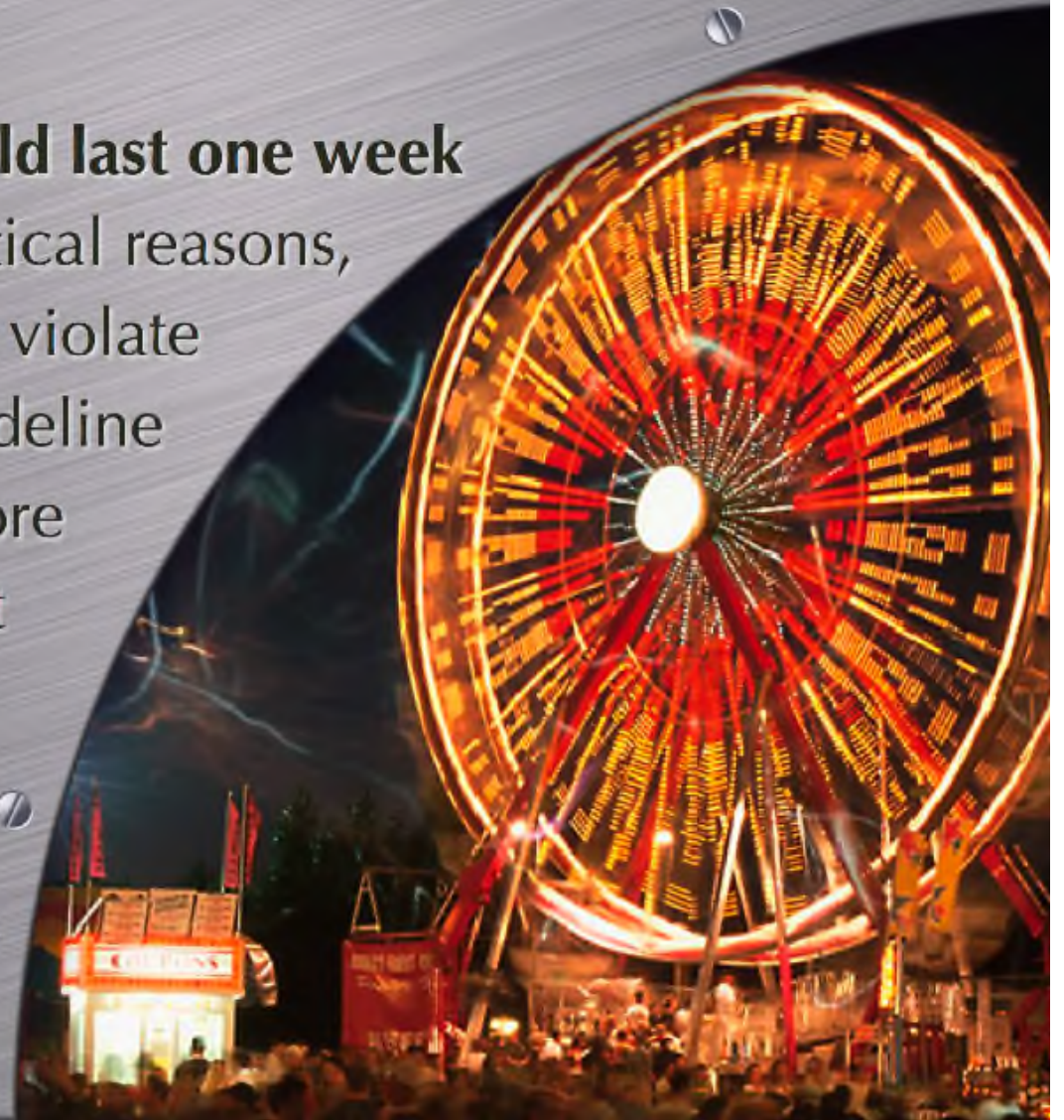
Current Status	Improvements		Reportal - MR Features		
	Units	%	Past	Tolerable	Goal
			Usability.Replacability (feature count)		
1,0	1,0	50,0	14	13	12
			Usability.Productivity (minutes)		
20,0	45,0	112,5	65	35	25
			Usability.ClientAcceptance (features count)		
4,4	4,4	36,7	0	4	12
			Development resources		
101,0			0		88

Current Status	Improvements		Survey_Engine_.NET		
	Units	%	Past	Tolerable	Goal
			Backwards.Compatibility (%)		
83,0	48,0	80,0	40	85	95
0,0	67,0	100,0	67	0	0
			Generate.WI.Time (small/medium/large seconds)		
4,0	59,0	100,0	63	8	4
10,0	397,0	100,0	407	100	10
94,0	2290,0	103,9	2384	500	180
			Testability (%)		
10,0	10,0	13,3	0	100	100
			Usability.Speed (seconds/user rating 1-10)		
774,0	507,0	51,7	1281	600	300
5,0	3,0	60,0	2	5	7
			Runtime.ResourceUsage.Memory		
0,0	0,0	0,0		?	?
			Runtime.ResourceUsage.CPU		
3,0	35,0	97,2	38	3	2
			Runtime.ResourceUsage.MemoryLeak		
0,0	800,0	100,0	800	0	0
			Runtime.Concurrency (number of users)		
1350,0	1100,0	146,7	150	500	1000
			Development resources		
64,0			0		

Current Status	Improvements		XML Web Services		
	Units	%	Past	Tolerable	Goal
			TransferDefinition.Usability.Efficiency		
7,0	9,0	81,8	16	10	5
17,0	8,0	53,3	25	15	10
			TransferDefinition.Usability.Response		
943,0	-186,0	#####	170	60	30
			TransferDefinition.Usability.Intuitiveness		
5,0	10,0	95,2	15	7,5	4,5
			Development resources		
2,0			0		48

• 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.



• FIRM EVO week •

Fri Mon Tue Wed Thu

Fri Mon Tue Wed Thu

Fri Mon Tue Wed Thu

Fri Mon

Cycle N

Cycle N1

Cycle N2

Cycle

FIRM EVO week

	Development Team	Users (PMT, Pros, Doc. writer, other)	CTO (Sys Arch, Process Mgr)	QA (Configuration Manager & Test Manager)
Fri day	PM: Send Version N detail plan to CTO + prior to Project Mgmt meeting PM: Attend Project Mgmt meeting: 12.00- 15.00 Developers: Focus on general maintenance work, documentation.		Approve/reject design & Step N Attend Project Mgmt meeting: 12- 15	Run final build and create setup for Version N-1. Install setup on test servers (external and internal) Perform initial crash test and then releas Version N-1
Monday	Develop test code & code for Version N	Use Version N-1		Follow up CI Review test plans, tests
Tuesday	Develop Test Code & Code for Version N Meet with users to Discuss Action Taken Regarding Feedback From Version N-1	Meet with developers to give Feedback and Discuss Action Taken from previous actions	Approve/reject design & Step N Attend Project Mgmt meeting: 12- 15	Follow up CI Review test plans, tests
Wednes day	Develop test code & code for Version N			Review test plans, tests Follow up CI
Thurs day	Complete Test Code & Code for Version N Complete GUI tests for Version N-2			Review test plans, tests Follow up CI

The

• Project Managers Meeting •

- 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**



Experiences and Conclusions

• Product Qualities •

- 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!**



Experiences and Conclusions

• Product Qualities •

- 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!**

Experiences and Conclusions

• Developers •

- EVO has resulted in
 - increased **motivation** and
 - **enthusiasm** amongst developers,
 - it opens up for **empowered creativity**



Experiences and Conclusions

• Developers •

- Developers
 - **embraced the method** and
 - **saw the value of using it,**
 - even though they found parts of Evo difficult to understand and execute





Experiences and Conclusions

• Project Leaders •

- 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 •

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

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 50

EVO's impact on Confrimit product qualities



- Only highlights of the impacts are listed here

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

• 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 Conformat 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

• Confirmit 9.0 •

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

- Theme for 9.0:
 - Extend usage in large corporations,
 - hence focus on usability, intuitiveness, easy to learn
- Market Research:
 - Support for large panels, up to 200 000 panellists.
 - Improve productivity in general for those who work with such large panels
- Improve throughput
 - for users that receive reports with more than 1 000 000 responses
 - (important for large corporations; HP, Microsoft, Accenture etc)

Confirmit v9

during development

Description of requirement/work task	Past	Status 11.09	Goal
Usability.Intuitiveness: Probability that a defined User can intuitively figure out how to do a defined Task correctly (without any errors needing correction)	30%	45%	80%
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.	30000	500000	200000
Performance.DataVolume: Numbers of survey responses that can be handled by Reportal. Tables should be generated within 5 seconds.	20000	500000	500000

Initial

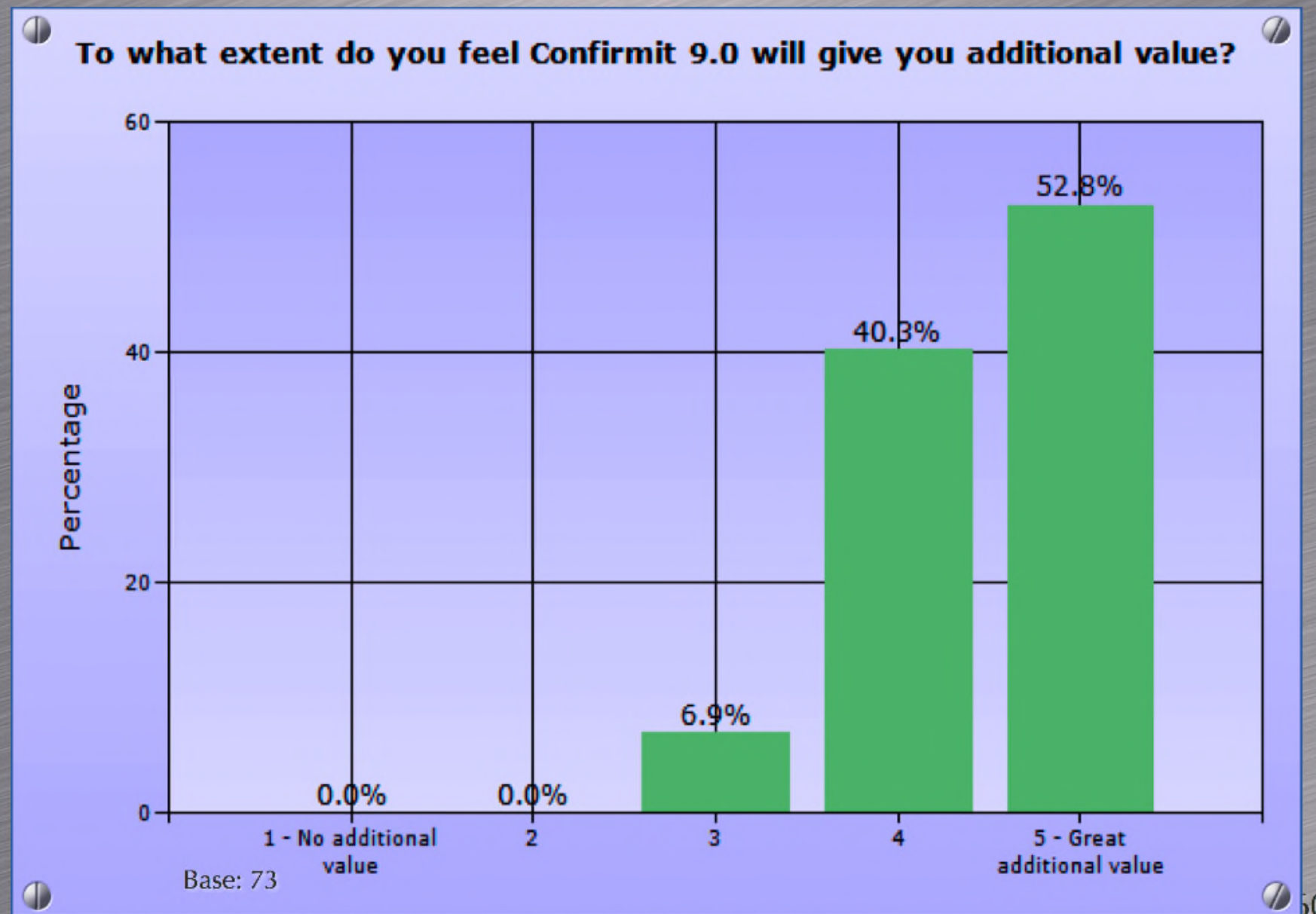
- **Customer Feedback** •

on the new Confirmat

9.0

November 24th, 2004

Initial perceived value of the new release



Confirmit 9.0 product qualities

Product quality	Description	Customer value
Intuitiveness	Probability that an inexperienced user can intuitively figure out how to set up a defined Simple Survey correctly.	Probability increased by 175%
Productivity	Time in minutes for a defined advanced user, with full knowledge of 9.0 functionality, to set up a defined advanced survey correctly.	Time reduced by 38%

Confirmit 9.0 product qualities

Product quality	Description	Customer value
Productivity	Time (in minutes) to test a defined survey and identify 4 inserted script errors, starting from when the questionnaire is finished to the time testing is complete and is ready for production. (Defined Survey: Complex survey, 60 questions, comprehensive JScripting.)	Time reduced by 83% and error tracking increased by 25%

Confirmit 9.0 product qualities

Product quality	Description	Customer value
Performance	Max number of panelists that the system can support without exceeding a defined time for the defined task, with all components of the panel system performing acceptable.	Number of panelists increased by 1500%
Scalability	Ability to accomplish a bulk-update of X panelists within a timeframe of Z second.	Number of panelists increased by 700%
Performance	Number of responses a database can contain if the generation of a defined table should be run in 5 seconds.	Number of responses increased by 1400%

- **Initial qualitative feedback on the new release** •

- **Initial qualitative feedback on the new release** •

"... keep up the good work."

- **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 ..."

- **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"

9.0 Customer Preview

● Observations ●

Seminar observations

On several occasions, customers gave **spontaneous "WOWs" and applause**

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.





► **Press releases**

► In the media

► Events

► Press area

► Contact

► Fact sheet

OSLO • LONDON • NEW YORK • SAN FRANCISCO • STOCKHOLM

News release

2004-11-29: Press Release from FIRM

New version of Confirmit increases user productivity up to 80 percent

NOVEMBER 29th, 2004: FIRM, the world's leading provider of online survey & reporting software, today announced the release of a new version of Confirmit delivering substantial value to customers including increased user productivity of up to 80 percent.

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

shortcuts

- [Press releases](#)
- [In the media](#)
- [Events](#)
- [Press area](#)

- Up to 175 percent more intuitive user interface*
- Up to 80 percent increased user productivity in questionnaire design and testing*
- Up to 1500 percent increased performance in Reportal and Panel Management*

Features delivering increased customer value include:

- A completely new and state-of-the-art user interface
- 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 Grinberg, Greenfield Online's Chief Information Officer. - We believe the improvements in Confirmit 9.0 will benefit Greenfield Online's survey programming, data collection and data delivery capabilities, helping us to bring even more value to our clients.

FIRM's VP of Marketing, Kjell Øksendal, comments;

- FIRM, through evolutionary development, is able to substantially increase customer value by focusing on key product qualities important for clients and by continuously asking for their feedback throughout the development period. Confirmit is used by the leading market research agencies worldwide and Global 1000 companies, and together, we have defined the future of online surveying and reporting, represented with the Confirmit 9.0.

Confirmit 9.0 was released onto FIRM's ASP environments in London and New York on November 27th. The new version will be available for server customers in January 2005.

** Measured in FIRM's Testlab by monitoring internal and external*

- Up to 175 percent more intuitive user interface*
- Up to 80 percent increased user productivity in questionnaire design and testing*
- Up to 1500 percent increased performance in Reportal and Panel Management*

Product Qualities

Features delivering increased customer value include:

- A completely new and state-of-the-art user interface
- 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

Solutions

- 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 Grinberg, Greenfield Online's Chief Information Officer. - We believe the improvements in Confirmit 9.0 will benefit Greenfield Online's survey programming, data collection and data delivery capabilities, helping us to bring even more value to our clients.

FIRM's VP of Marketing, Kjell Øksendal, comments;

- FIRM, through evolutionary development, is able to substantially increase customer value by focusing on key product qualities important for clients and by continuously asking for their feedback throughout the development period. Confirmit is used by the leading market research agencies worldwide and Global 1000 companies, and together, we have defined the future of online surveying and reporting, represented with the Confirmit 9.0.

Confirmit 9.0 was released onto FIRM's ASP environments in London and New York on November 27th. The new version will be available for server customers in January 2005.

* Measured in FIRM's Testlab by monitoring internal and external

• Confirmit 9.x •

Confirmit continues to evolve

Code quality – “green” week

- **1 week every month**
used for delivering towards
Internal Stakeholder Requirements.
- In these “green” weeks, some of the deliverables will be less visible for the end users, but more visible for our QA department.
- We manage code quality through an Impact Estimation table.

Code quality – “green” week

Speed

Maintainability.Doc.Code

NUnit Tests

Peer Tests

FX Cop

Test Director Tests

Robustness.Correctness

Robustness.Boundary Conditions

Speed

Resource Usage.CPU

Maintainability.Doc.Code

N Unit Tests

Code quality – “green” week

Current Status		Improvement	Goals			Step 6 (week 14)		Step 7 (week 15)	
	Units		Past	Tolerable	Goal	Estimated Impact	Actual Impact	Estimated Impact	Actual Impact
	100,0	100,0	0	80	100			100	100
Speed									
	100,0	100,0	0	80	100	100			
Maintainability.Doc.Code									
	100,0	100,0	0	80	100	100	100		
InterviewerConsole									
NUnitTests									
	0,0	0,0	0	90	100				
PeerTests									
	100,0	100,0	0	90	100				
FxCop									
	0,0	10,0	10	0	0				
TestDirectorTests									
	100,0	100,0	0	90	100				
Robustness.Correctness									
	2,0	2,0	0	1	2	2	2		
Robustness.BoundaryConditions									
	0,0	0,0	0	80	100				
Speed									
	0,0	0,0	0	80	100				
ResourceUsage.CPU									
	100,0	0,0	100	80	70	70			
Maintainability.Doc.Code									
	100,0	100,0	0	80	100	100	100		
SynchronizationStatus									
NUnitTests									

June 2006 Evo Cycle -Day Tasks

- **Monday:** Evo documents completed, tests written (NUNIT and GUI)
- **Tuesday:** Development
- **Wednesday:** Development
- **Thursday:** Development
- **Friday:** Development
- **Monday:** Full feedback from all stakeholders (timing of this day can be adjusted by the project)
- **Tuesday:** Test manual published and Initial feedback received friendliness & Code quality and environment
- **Wednesday:** Tests updated and passed
- **Thursday & Friday:** Feature team & project planning (QA: 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

Trond.Johansen@firmglobal.com



Trond.Johansen@firmglobal.com

• **Thanks** •

www.Gilb.com



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

•**Design Effort** [Roadbed.Drainage System, Product XX]: 'Approved by Team' 13:59 Tuesday (Day 1)

-Ambition Level: 10X "at least 10 times less engineering effort than now"

-Administration:

- Approved: by Team' 13:59 Tuesday (Day 1) ok to progress to strategy phase.
- Type: Product Quality [Product XX].
- Version: 12 Oct 2004 10:12, 11:38
- Owner: Idar

-Stakeholders: Senior Road Designers, Road Designers, Drainage System Designers, Contractors.

-Scale: **Hours of Engineering Effort per 10 km road to Complete Roadbed Description for a defined Ideal Engineering Level: default 100%.**

•Assumption: the level of qualities is the same for comparative measurements. E.g. we do not save time, only to turn around and use it to increase quality. We still saved time for the old quality level. <-TG

---- Benchmarks----- Analysis-----

- Past [VXX Our Infrastructure Design, Finland] ??? <-IK
- Bad: Past [IEL = 20%? "<wrong mass calculations & drawings, absence of stakeout data>", 2004, Project = <general guess IK>] <30> ±10 hours/10km <- SWAG IK
- Good: Past [IEL = 90%? "<better mass calculations & drawings, some stakeout data>", 2004, Project = <general guess IK>, Excel & our product, Swedish Users] <80> ±20 hours/10km <- SWAG IK
- Trend [Our Product XX] <customers are more demanding, 20% is no longer a good enough level>. <- Heidi
- Record [InXXX, 2004] <better than us in design, high Engineering Level, more consistent but not redesign> <-IK "our system is clearly worse here than the competitor - so we must improve"

---- Constraints----- Requirement-----

- Tolerate [End 2005, Road Designers,] Past - 25% of hours
- Rationale: least powerful sales argument for selling new version.
- Survival [Anytime,] <today's level or better>
- Rationale: we could lose customers to competitors.

---- Targets ----- Requirements -----

- G1. Goal [IEL = 90%?, 2005 Q4, Norwegian & English] <8> ±2 hours/10 km <- Heidi, Berit, Inge
- G2. Goal [IEL = 90%?, 2006 Q1, Swedish] <8> ±2 hours/10 km <- Heidi, Berit, Inge

---- Evolutionary Goals-----

- Short Term 1: Goal [End November 2004, Stakeholder = {SVV, Road Designers}] Past - 20%?
- Short Term 2: Goal [End 2004] Past -40%?
- Goal [End January 2005] Past -50%?
- Goal [End Feb 2005] Past -60%?

GLOSSARY-----

Hours of Engineering Effort: net, actually applied to task hours.

Complete: {all considerations taken, engineering quality controlled, contractor approved, to a defined level of IEL}

Roadbed Description: define as: {cross-section drawings, mass calculation, Geometrical Description: {existing terrain, related water at sewer, other roads, tunnel}, geometrical control}.

<Ideal> Engineering Level: **IEL**: defined as: doing tasks to an ideal level of completion. This is often compromised intentionally to save engineering effort and time. <table to define % must be developed, at least classify things>

Meter: <how to measure this practice>

Design: defined as: design and redesign

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.**

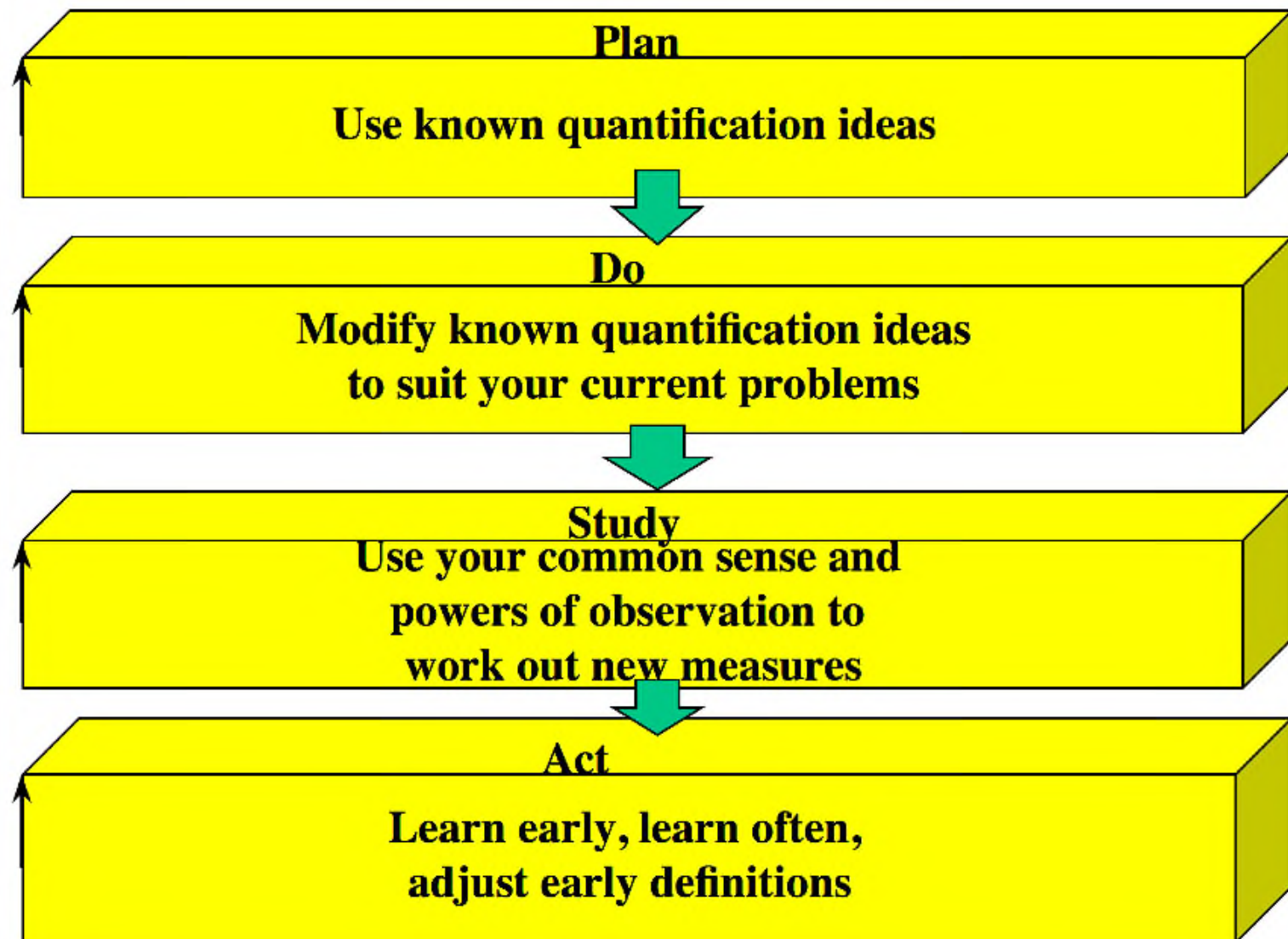
We do not have to work in the dark, now that we have professional systems analysis.

Ramo98 page 81

Simon Ramo and Robin K. St.Clair, The Systems Approach: Fresh Solutions to Complex Civil Problems Through Combining Science and Practical Common Sense, 1998, 150pp, © TRW, Inc., Manufactured in USA, KNI Incorporated, Anaheim CA. Free copy at TRW Stand at INCOSE conference 2002



How to Quantify Quality



Quantifying Quality

‘Environmentally Friendly’ Quantification Example

'Environmentally Friendly' Quantification Example

Give the quality a stable name tag

Environmentally Friendly

'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

'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

'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

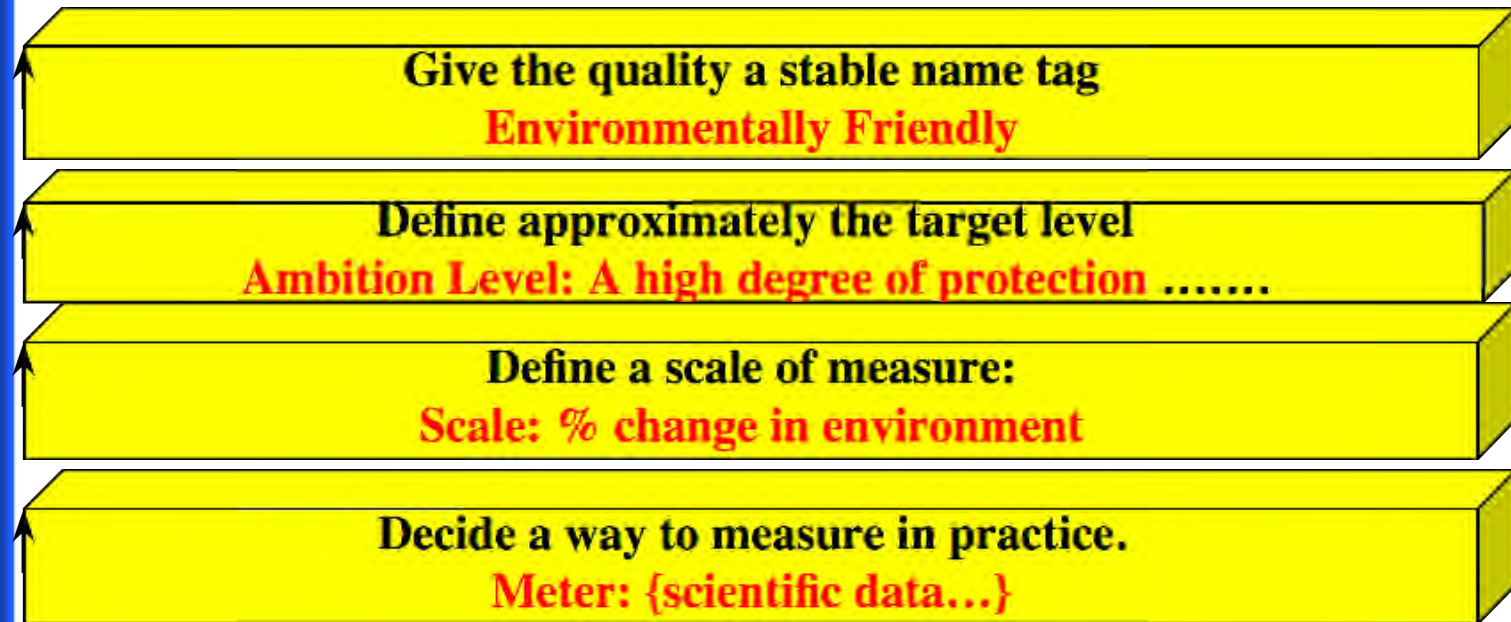
Define a scale of measure:

Scale: % change in environment

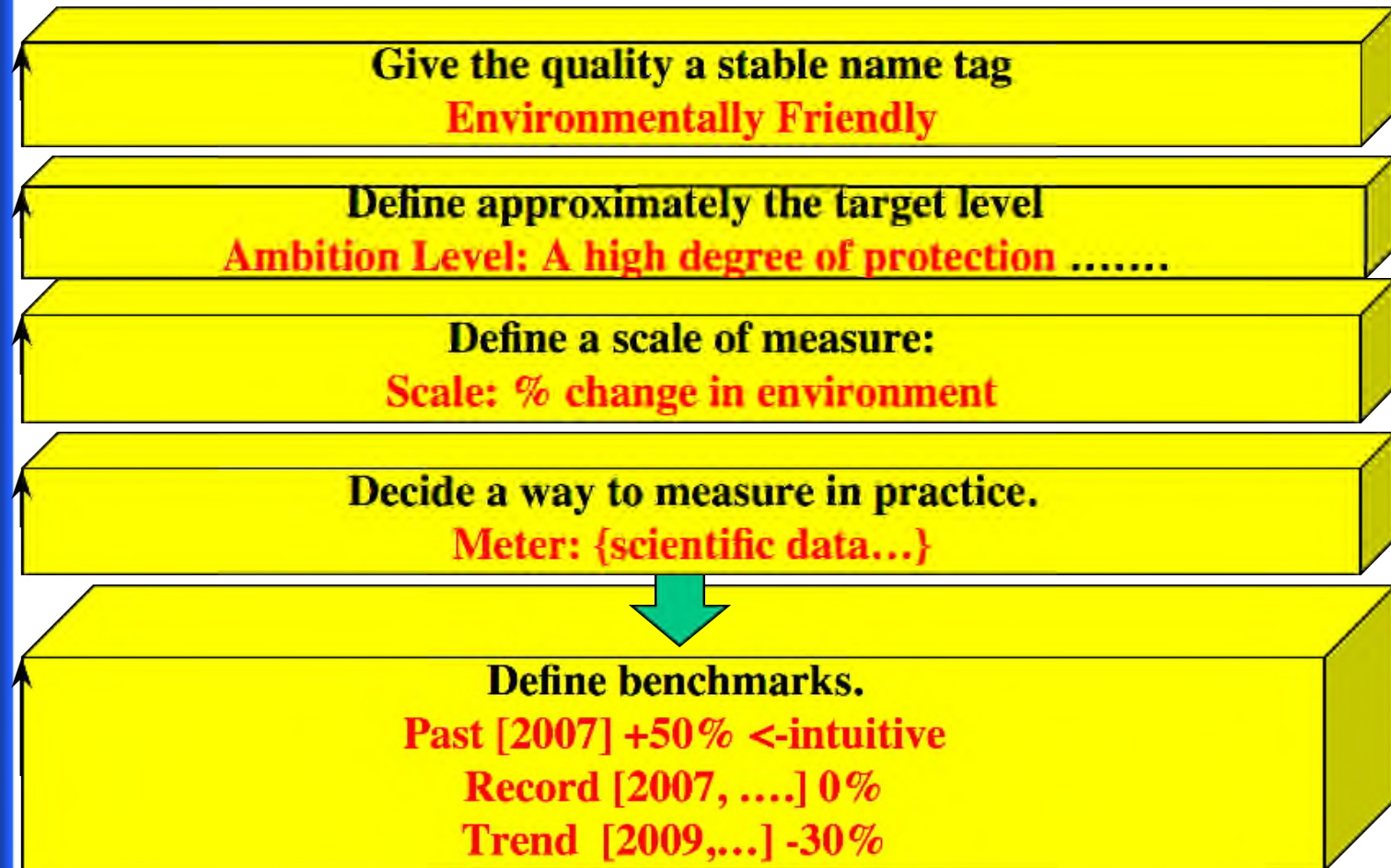
Decide a way to measure in practice.

Meter: {scientific data...}

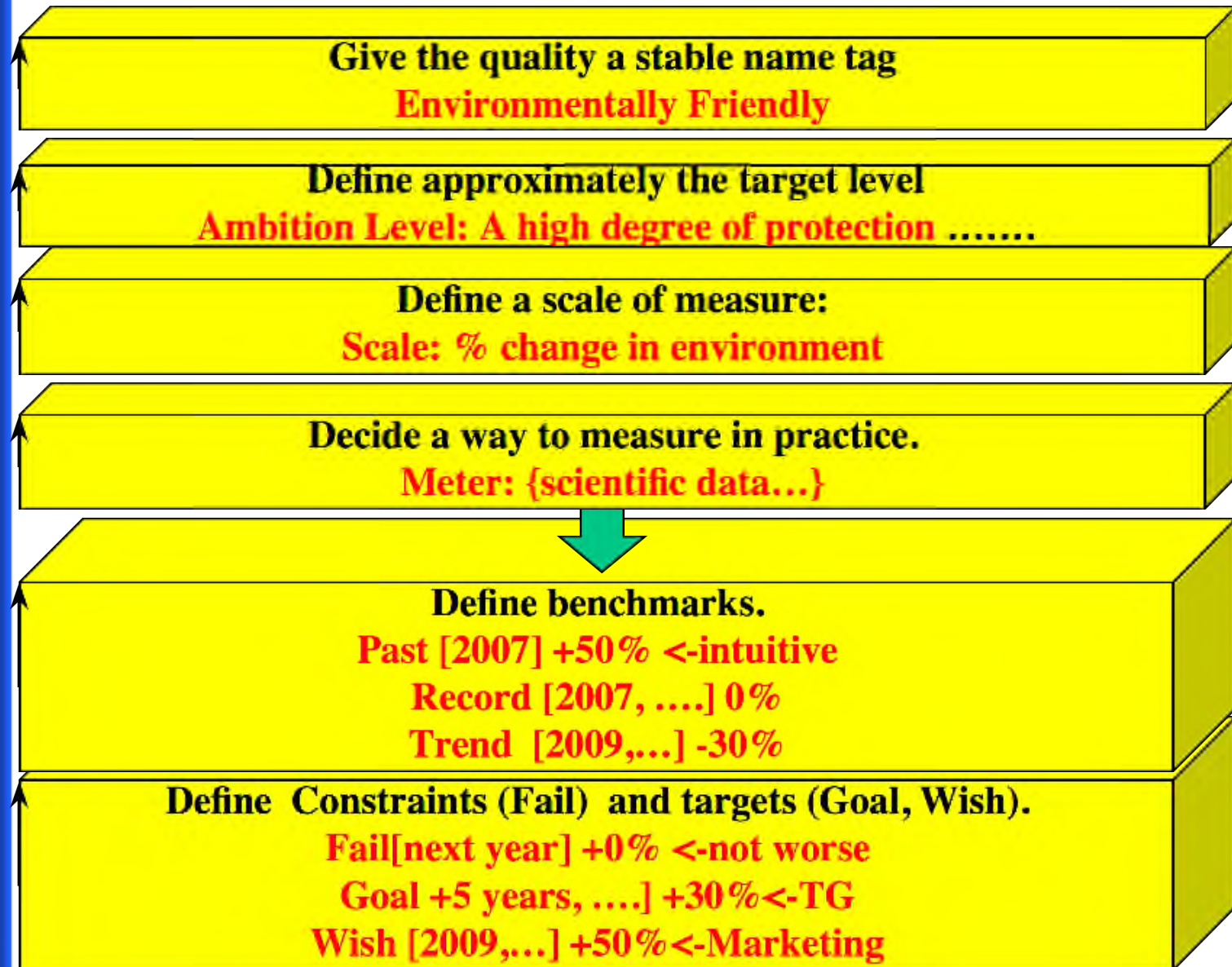
'Environmentally Friendly' Quantification Example



'Environmentally Friendly' Quantification Example

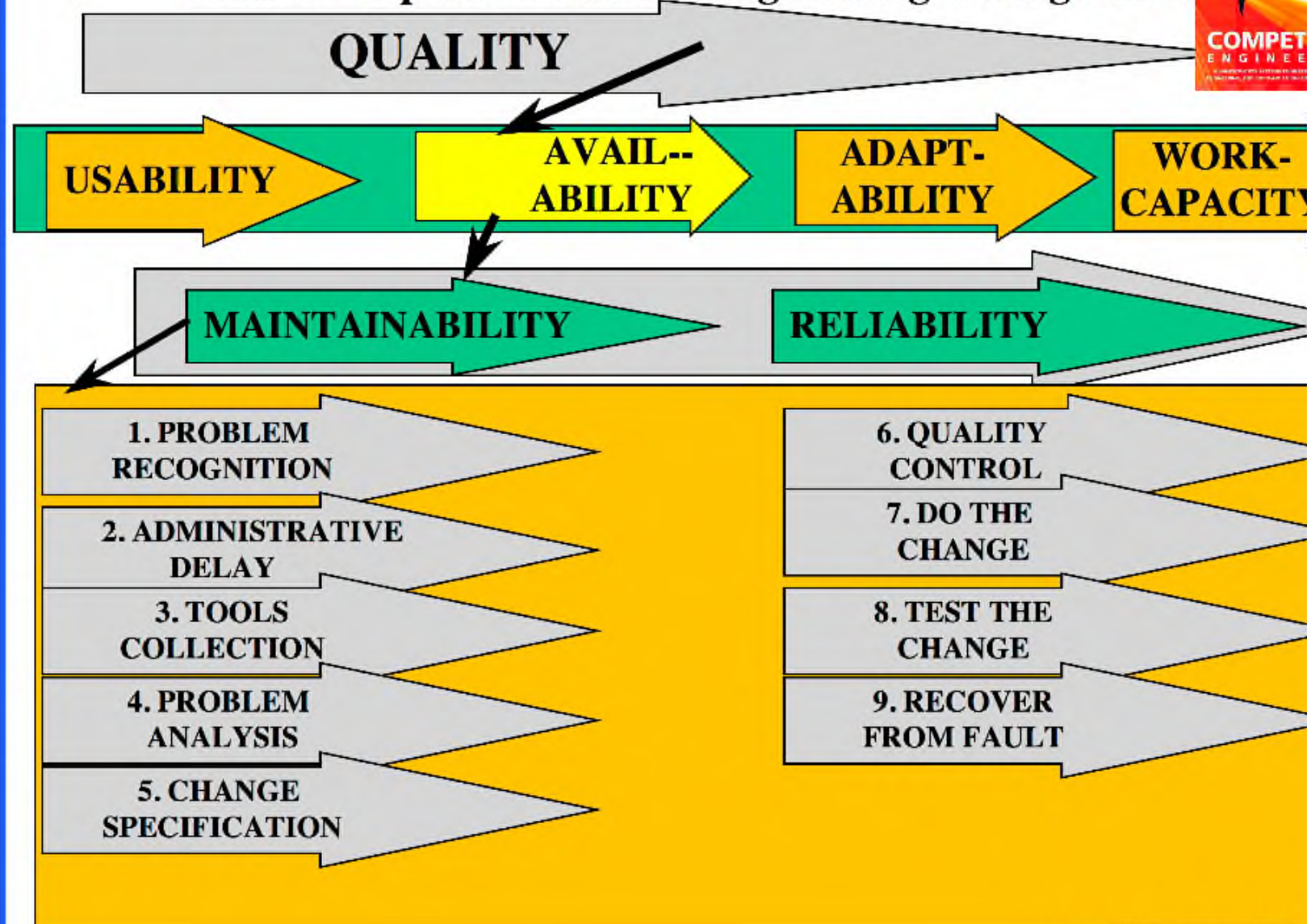


'Environmentally Friendly' Quantification Example

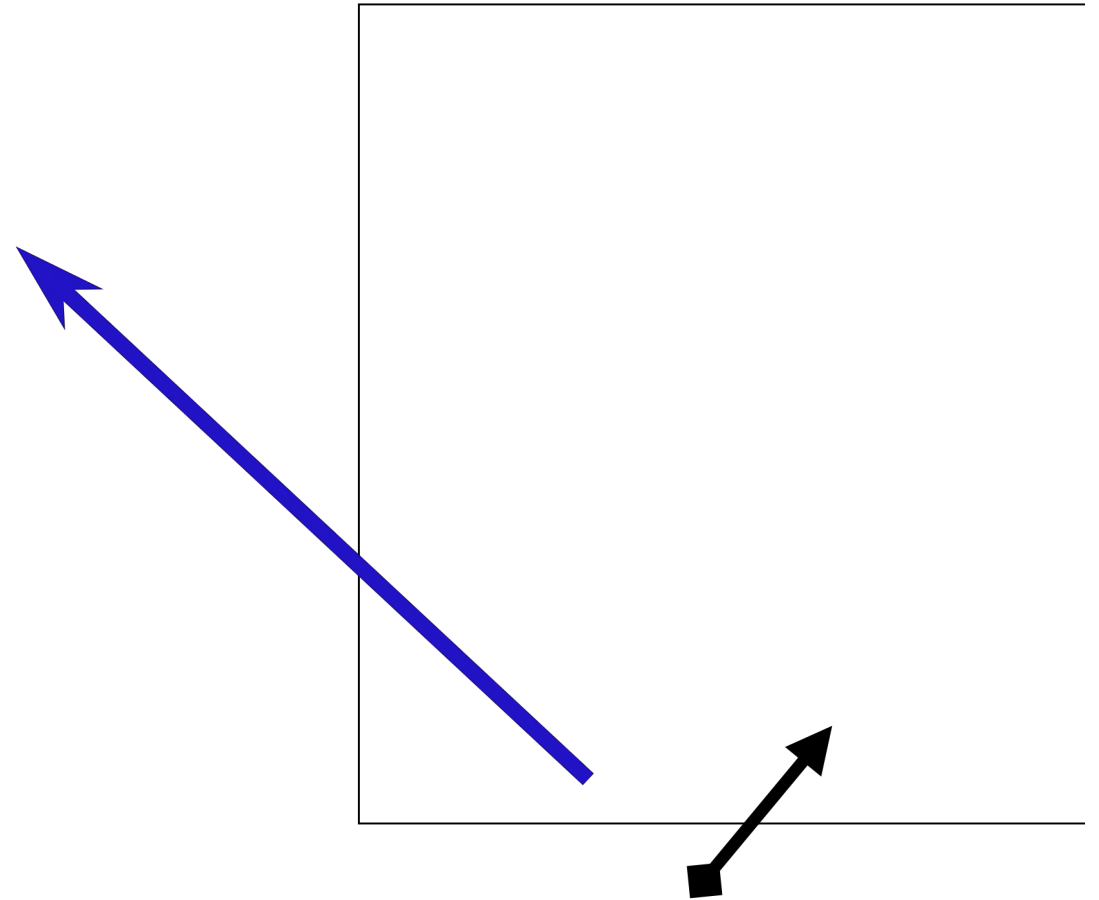


Quantifying Quality

Devices to help quantify quality ideas:
Standard Hierarchy of Concepts from
Gilb: Principles of Software Engineering Management.

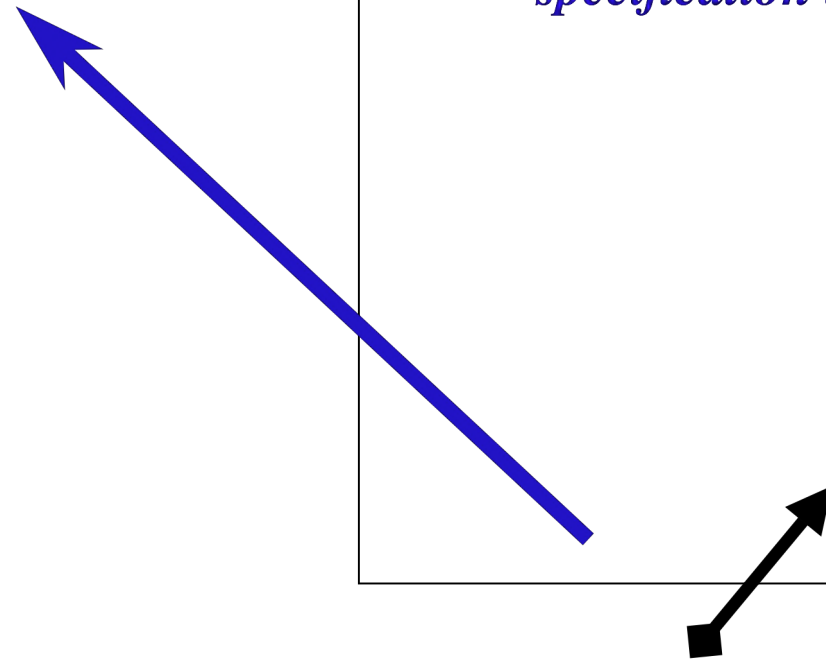


Using 'Parameters' when defining a Scale of Measure



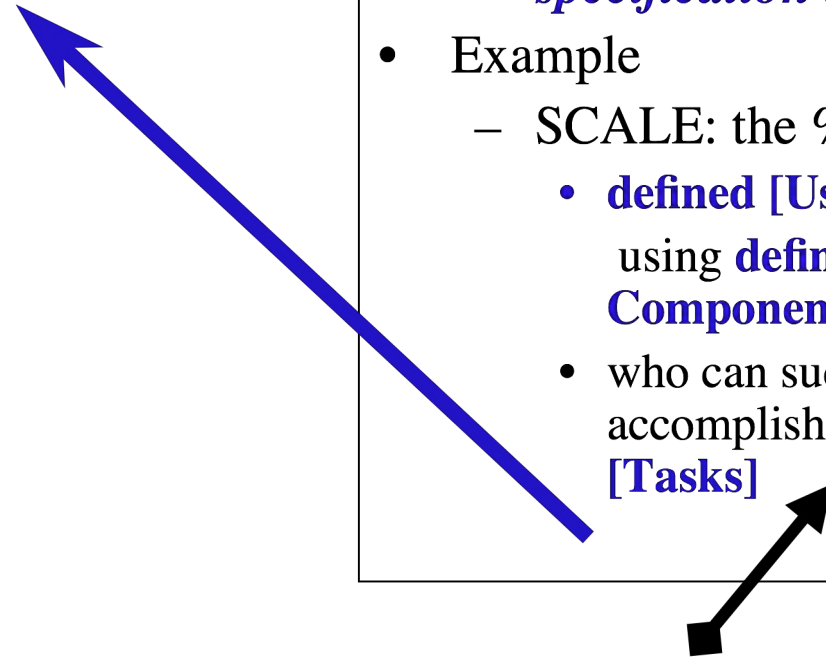
Using 'Parameters' when defining a Scale of Measure

- *Using [qualifiers] in the SCALE definition*
 - *gives flexibility of detailed specification later.*



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]**



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]**

[Scale Parameters]

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]

[Scale Parameters]

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.

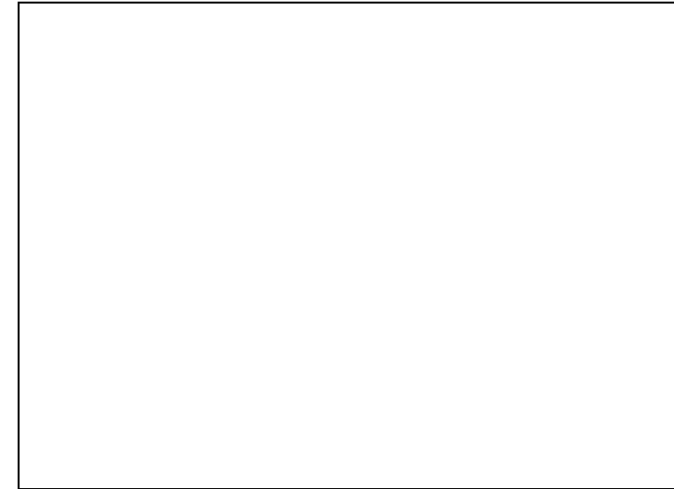
Quantifying Quality

A 'Quality Quantification' Principle



Quantifying Quality

A 'Quality Quantification' Principle



Quantifying Quality

A 'Quality Quantification' Principle

0. *THE PRINCIPLE OF*



A 'Quality Quantification' Principle

0. THE PRINCIPLE OF 'BAD NUMBERS BEAT GOOD WORDS'



A 'Quality Quantification' Principle



He had a lot of hats.
He wants to be best in hatmanship.

Scale: hats on his head.

Past:3

Goal: 13



0. *THE PRINCIPLE OF 'BAD NUMBERS BEAT GOOD WORDS'*

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

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

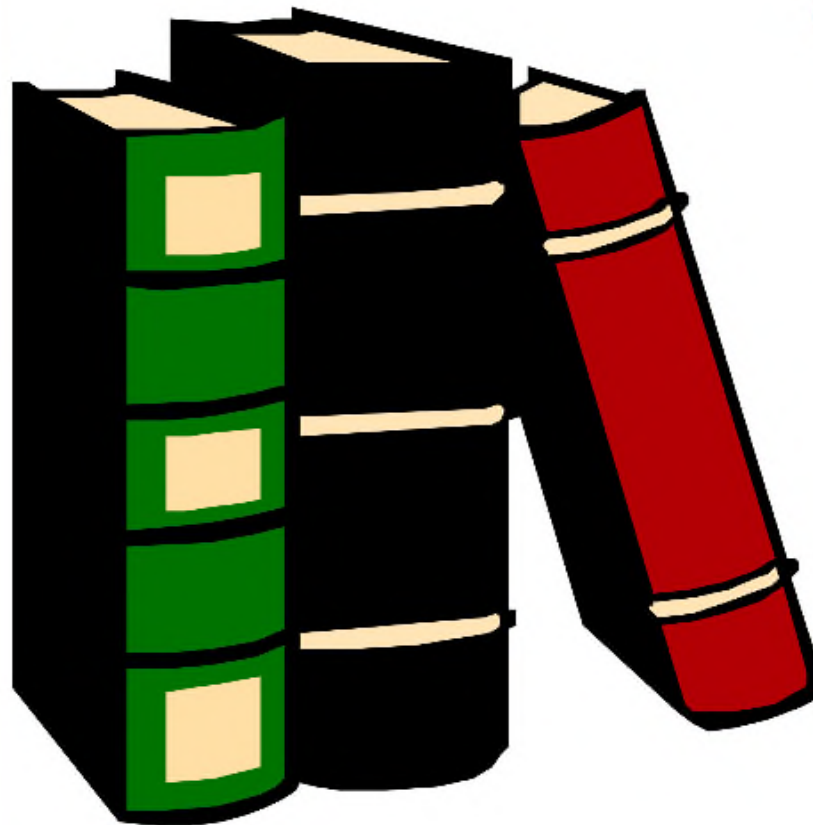
Past.... Goal.....etc.



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: page 124*

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

Not Clear!

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,

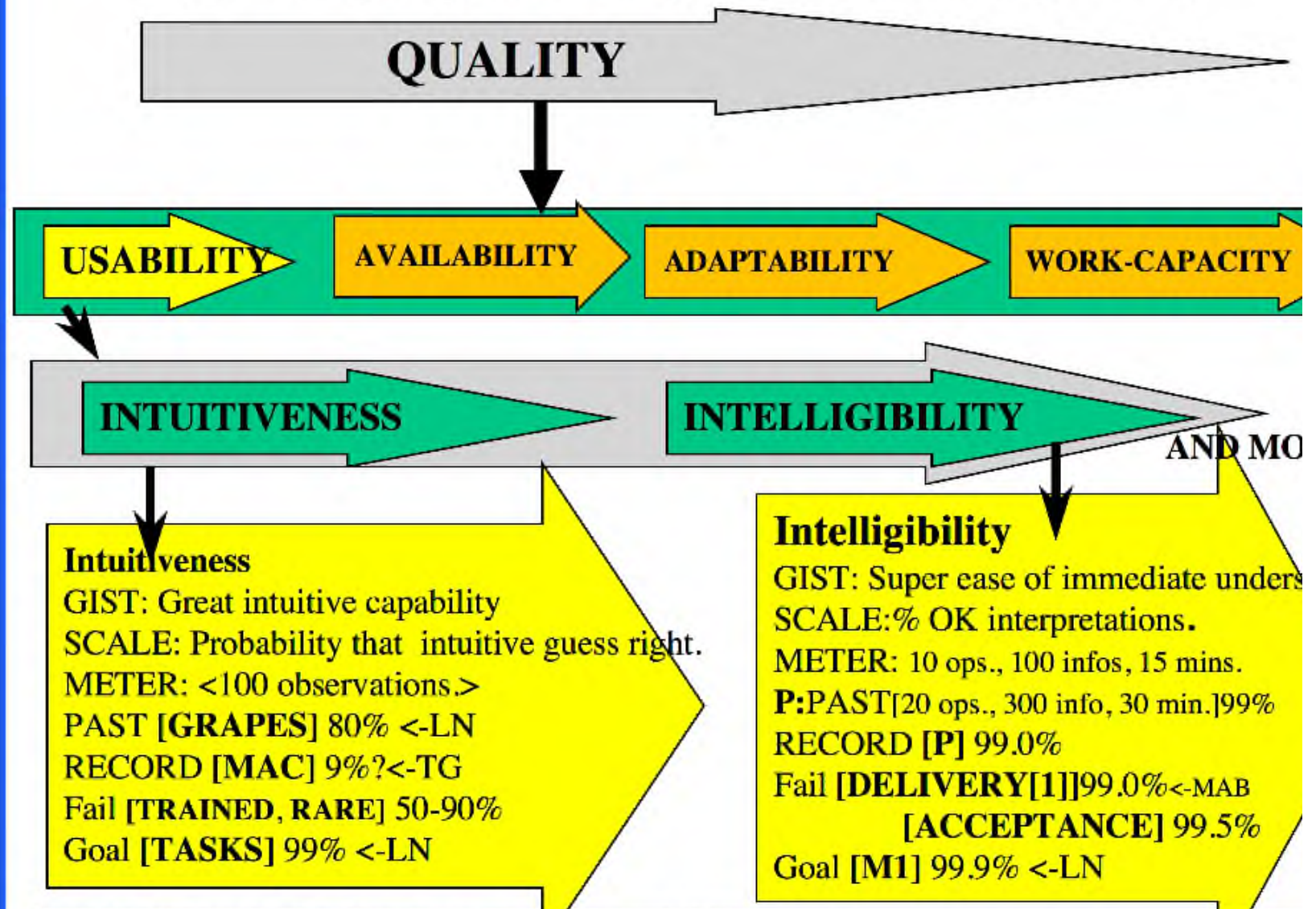
Learning Effort: Scale: Hours to learn,

Productivity: Scale: Tasks per hour,

Error Rate: Faults per 100 tasks,

Like-ability: % Users who like the system,

Quantifying Usability (Real C&C System)



TRAINED: DEFINED:C&Ctl. operator, approved course, 200 hours duration.

RARE: DEFINED: types of tasks performed less than once a week per op.

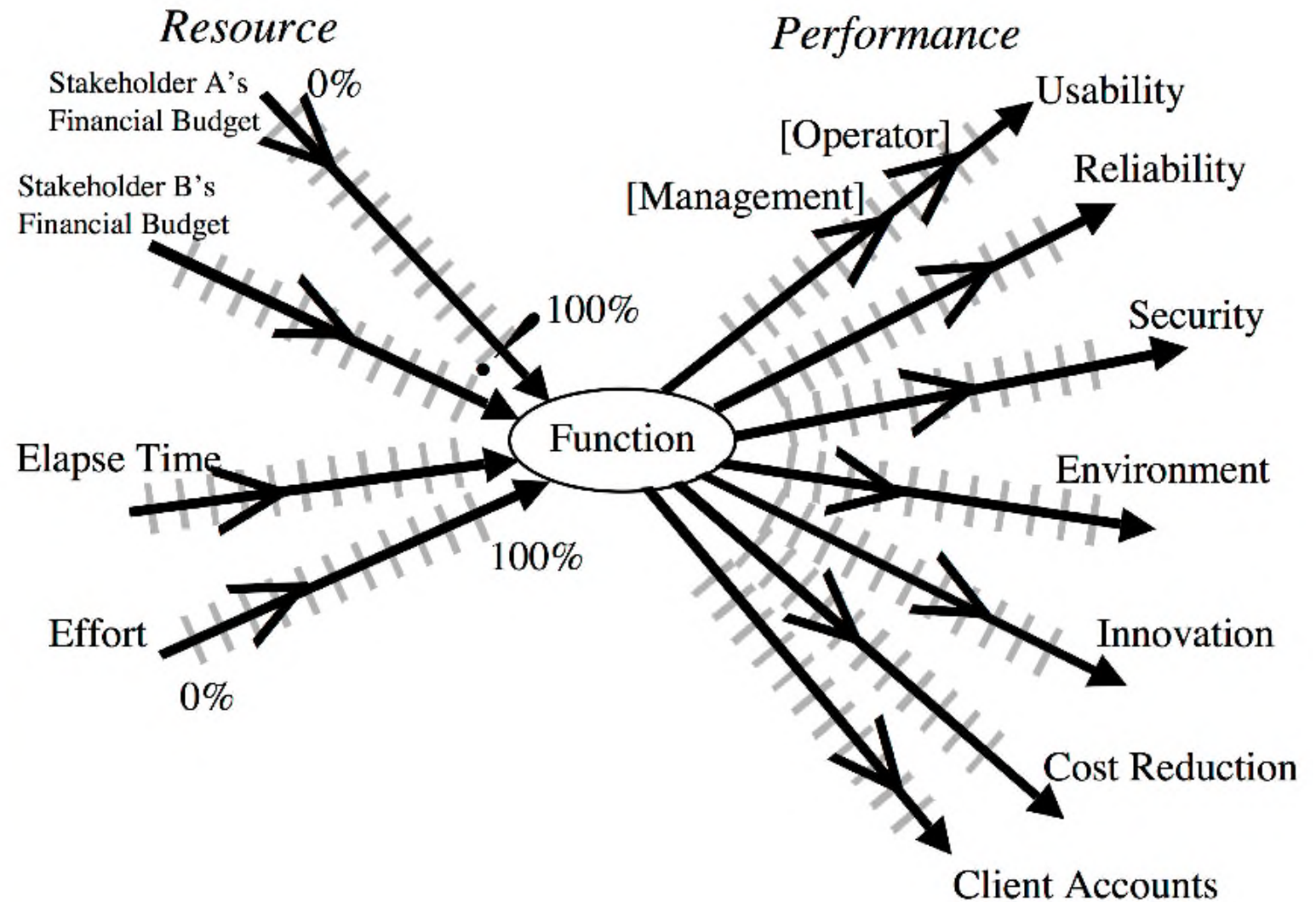
TASKS: DEFINED: onboard operator distinct tasks carried out.

ACCEPTANCE: DEFINED: formal acceptance testing via customer contract.

DELIVERY: DEFINED: Evolutionary delivery cycle, integrated and useful.

Quantifying Quality

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



3. THE PRINCIPLE OF 'SCALAR DEFINITION'

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

3. THE PRINCIPLE OF 'SCALAR DEFINITION'

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

Flexibility:

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

<name tag of the objective>

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 Converte

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, 2008] Test X

Meter [UK, 2009] Test Y

8. THE PRINCIPLE OF 'BENCHMARKS'

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

Reliability

Scale: Mean Time To Failure

Past [US DoD, 2007] 30,000 Hours

Trend [Nato Allies, 2009] 50,000 Hours

Goal [UK MOD, 2010] 60,000 Hours

9. THE PRINCIPLE OF 'NUMERIC FUTURE'

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

Usability:

Scale: Time to learn average task.

Past [Old product, 2007] 20 minutes

Wish [New product, 2009] 1 minute

Stretch [End 2010, Students] 2 minutes

Goal [End 2011, Teachers] 5 minutes

Some Planguage 'Quality Quantification' Concepts

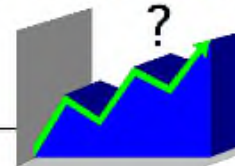
Quantifying Quality



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

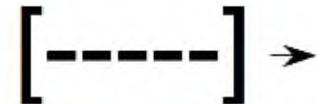


RECORD: best in some class, state of the art. Something to beat. A challenge for you. An extreme PAST.



TREND: a future ? guess based on the PAST.

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



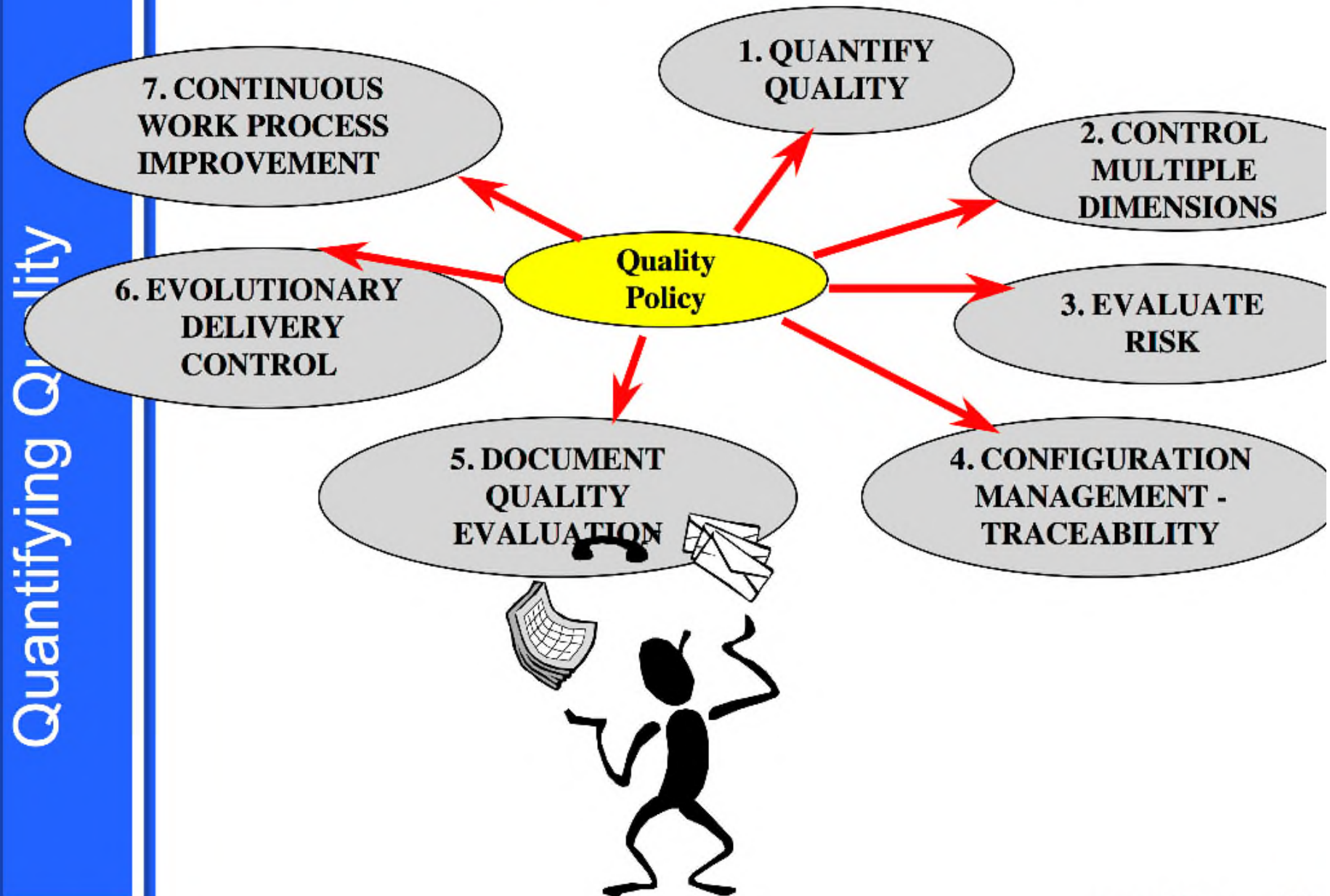
Goal: the level needed for satisfaction, happiness, joy and 100% full payment!



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



A Corporate Quality Policy (Euro Multinational)



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. “**

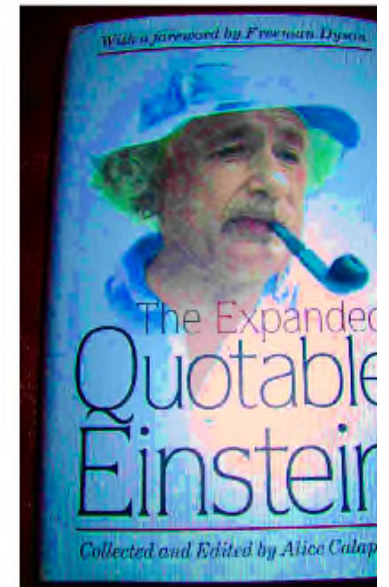
<- (Quality Manual Source is) 5.2.2, 4.1.2, 4.1.5, 5.1.1, 6.1,
6.4.1, 7.1.1, 7.3 and many others.

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)**

Source detail in notes section of this slide. (Calaprice, 2000)



www.Gilb.com Slide

LAST SLIDE

SEE

WWW.Gilb.COM

FOR MORE DETAIL

“Competitive Engineering” at www.gilb.com
(or via memory stick here at conference from
presenter):



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 1Major 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 this 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 project 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) 1 of 2

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 Meter s will be specified for each Scale. Preference will be given to previously defined Meter s 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.

Continued next slide

www.Gilb.com Slide

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 a 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 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 Latest Book, Summer 2005

- Tom Gilb,
 - **Competitive Engineering:**
 - A Handbook For Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage
 - ISBN 0750665076
 - Publisher:
 - Elsevier Butterworth-Heinemann
 - [Ask me for a free electronic copy !](#)
 - [Email Tom@Gilb.com](mailto:Tom@Gilb.com)

