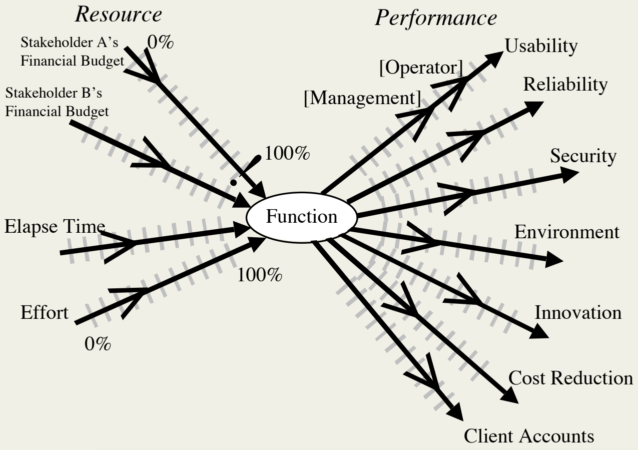
How to define and quantify any useful quality aspect of a system.
A Workshop

COMPETITIVE ENGINEERING
TOM GILB
A HANDBOOK FOR SYSTEMS ENGINEERING, REQUIREMENTS ENGINEERING, AND SOFTWARE ENGINEERING USING PLANGUAGE
GILB
ELSEVIER

Tom Gilb –
Roots, Bergen
Tuesday April 28th 2009
1330-1700



What it's about

- **How to define and quantify any useful quality aspect of a system- a workshop**
  - **A clear weakness with most all development methods, including Agile methods, is their failure to explicitly deal with critical qualities of the IT system. They stop at bug counting.**
  - **We need to deal directly with a vast array of stakeholder-valued qualities, including availability, maintainability, adaptability, security, portability, usability, robustness, and many more.**
  - **This practical workshop will equip participants with practical tools, experiences and exercises so they will master the ability to quantify any required quality requirement.**

Time Plan 13:30-17:00

1330-1400 Introduction: **Quantifying Quality**

1400-1550 Simple workshop exercises

1. identifying quality requirements
2. indirect identification (means ->ends)
3. Naming – the tag –
4. Ambition level
5. Ambiguity Test
6. Stakeholders Viewpoint

Break

1600-1700 Advanced Workshop Exercises

7. Scale definition
8. meter definition
9. Scale parameters
10. Scalar level expressions

(Qualifiers, level, uncertainty, source, justification)

11. Quality Benchmarks
12. Quality Constraints
13. Quality Targets
14. more if time ??

Impacted By <design>

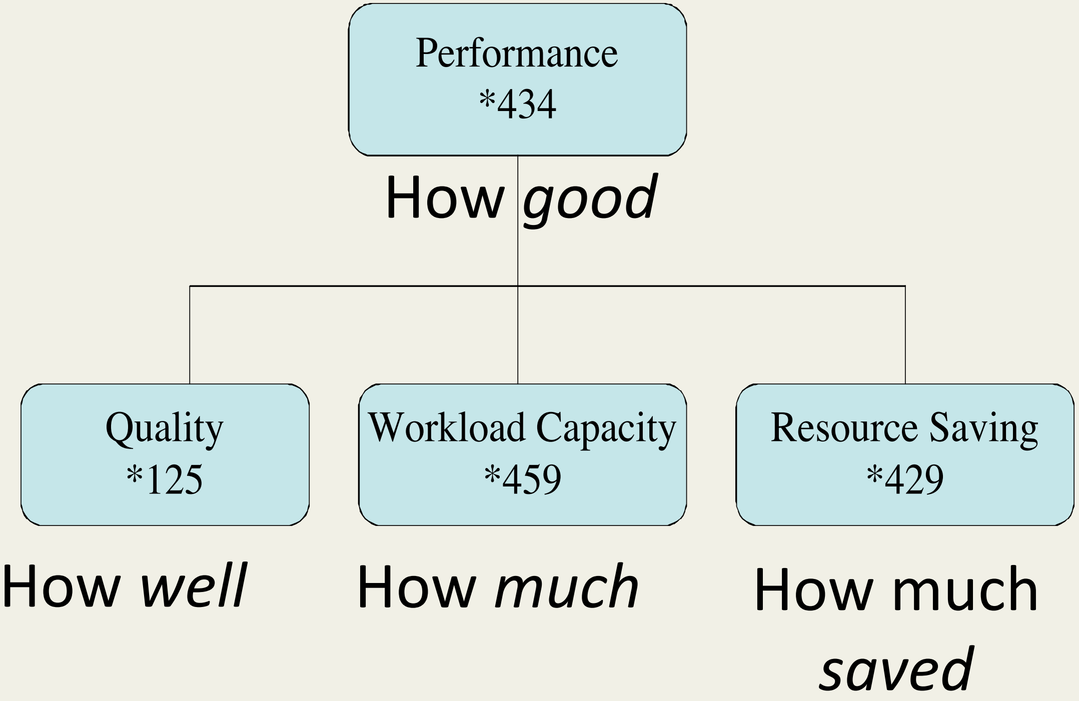
1330-1400 Introduction: **Quantifying Quality**

Quality: the concept, the noun

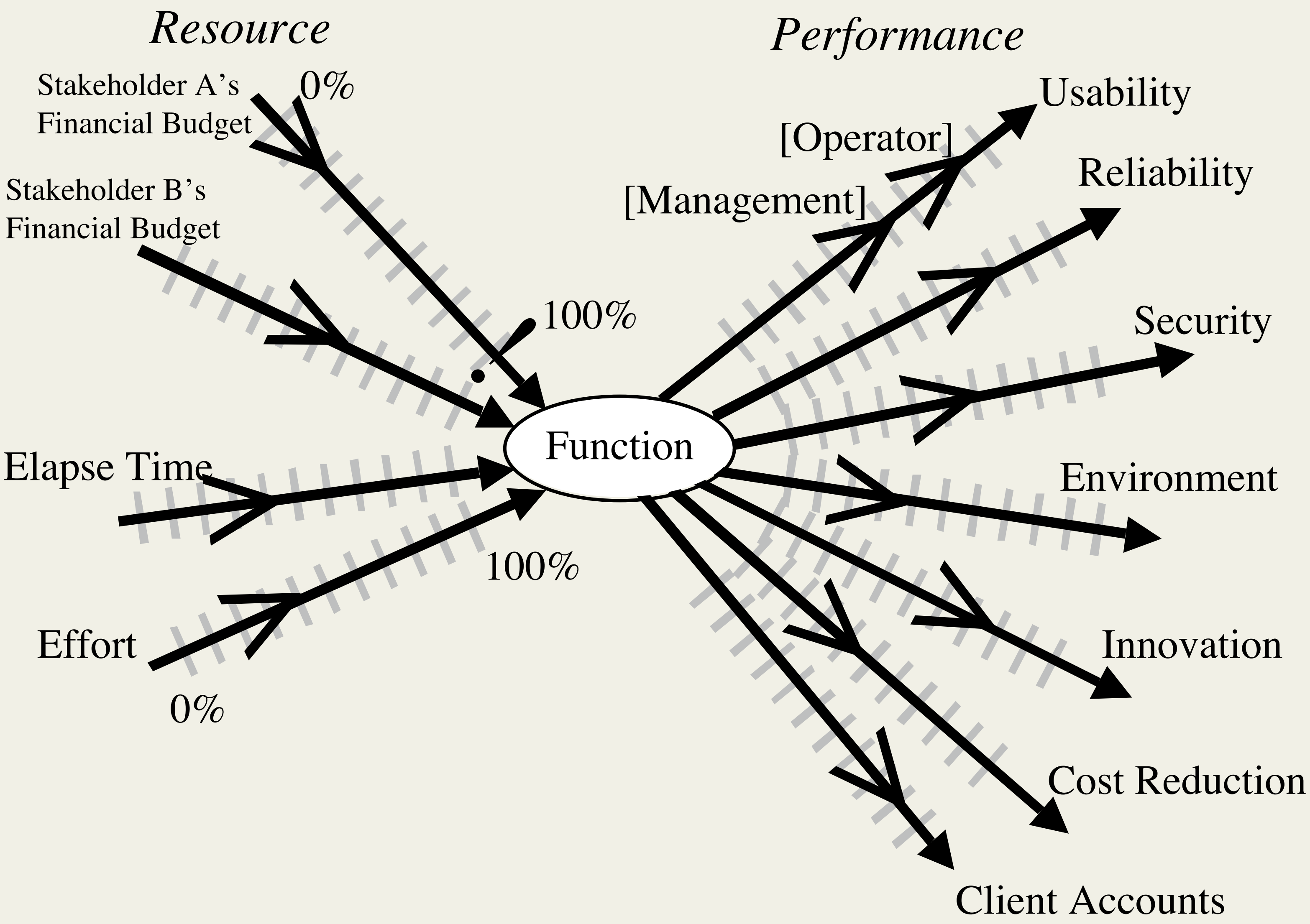
Planguage Concept *125, Version: March 20, 2003

A ‘**quality**’ is

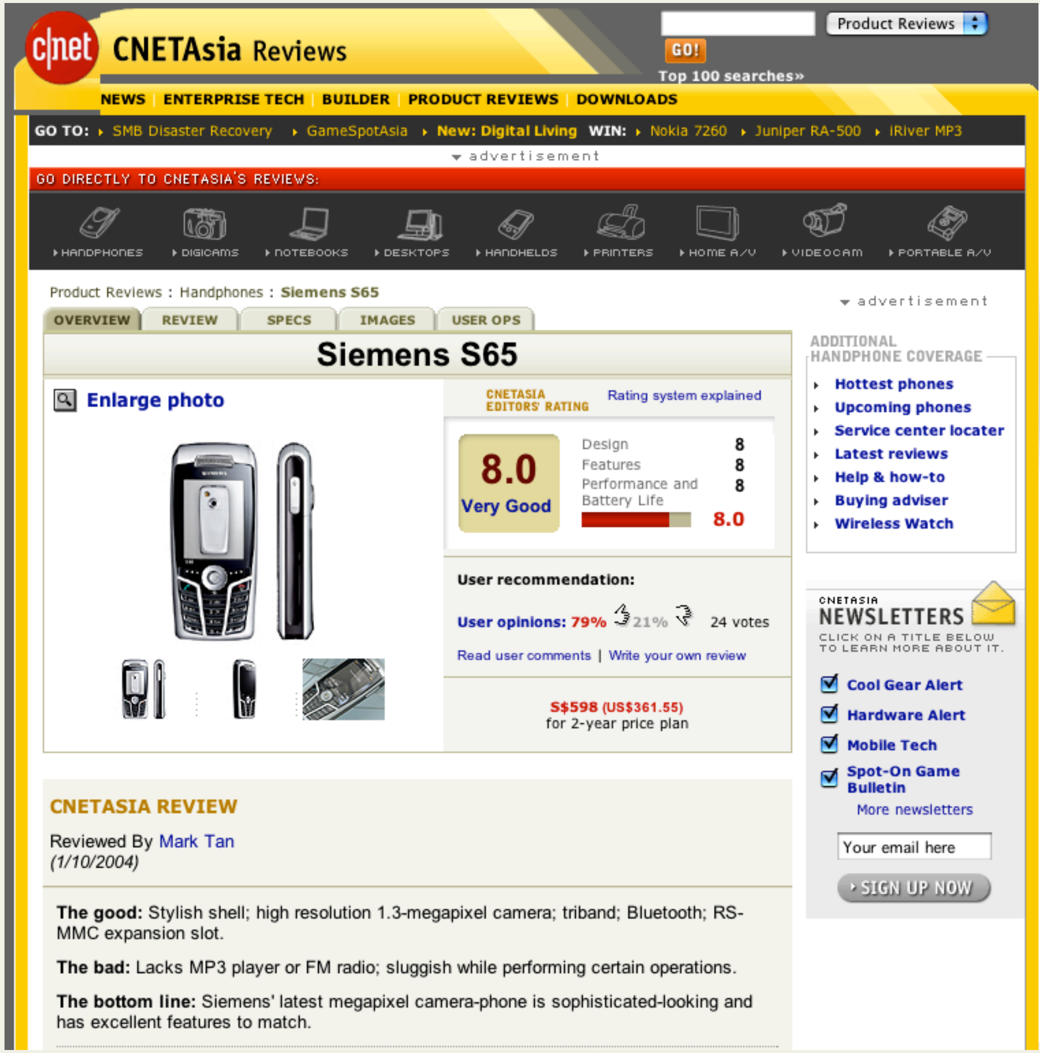
- a scalar attribute -|-|-|-|- (Scale symbol)
- reflecting ‘how *well*’ ------Past Level<------------>
- a system functions. (Fn)------Past Level<--------->



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



"You can nearly measure everything but **how can you measure style?**" That's Siemens catchphrase for its new S65



CNETAsia Reviews

Product 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

▾ advertisement

GO DIRECTLY TO CNETASIA'S REVIEWS:

‣ HANDPHONES ‣ DIGICAMS ‣ NOTEBOOKS ‣ DESKTOPS ‣ HANDHELDS ‣ PRINTERS ‣ HOME A/V ‣ VIDEOCAM ‣ 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
	8.0

User recommendation:

User opinions: 79% 21% 24 votes

Read user comments | Write your own review

S$598 (US$361.55)
for 2-year price plan

▾ advertisement

ADDITIONAL HANDPHONE COVERAGE

- Hottest phones
- Upcoming phones
- Service center locater
- 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

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.

Enthoven on Numbers

- “Numbers are a part of our **language**.
- Where a quantitative matter is being discussed,
  - the greatest **clarity of thought** is achieved by using numbers
  - instead of avoiding them,
  - ***even when uncertainties** are present.*
- *This is not to rule out judgment and insight.*
  - *Rather, it is to say, that*
  - *judgments and insights need,*
  - *like everything else,*
  - *to **be expressed with clarity***
  - *if they are to be **useful**.”*
- Alain Enthoven, June 1963, Naval War College, Newport Rhode Island (see note for more detail), Hughes98, Rescuing Prometheus p164

See the note for more detail on Enthoven

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**

__Quality__ is characterized by these traits

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

The Confirmit Case Study 2003-2009

See paper on this case at www.gilb.com

Papers/Cases/Slides, Gilb Library,

value slide w… http://www.gilb.com/tiki-download_file.php?fileId=152

ppr wrong ag… http://www.gilb.com/tiki-download_file.php?fileId=50

Paper Firm http://www.gilb.com/tiki-download_file.php?fileId=32

And see papers (IEEE Software Fall 2006) by Geir K Hanssen, SINTEF

Their product = confirmit®

Chief Storyteller = Trond Johansen

Customer Successes in Corporate Sector

AMERICAN EXPRESS	APC Legendary Reliability	AVAYA	BARCLAYS
BRITISH AIRWAYS	Countrywide	DOWJONES	egg™
hp invent	JPMorganChase	Microsoft®	Pfizer
PROGRESSIVE	SIEMENS	symantec.	telenor
TOYOTA	UBS Warburg	virgin.net	arnoldworldwide

Real Example of 1 of the 25 Quality Requirements

Usability.Productivity ***(taken from Confirmit 8.5,***
performed a set of predefined steps, to produce a standard MR Report.

development)

Scale for quantification: Time in minutes to set up a typical specified Market Research-report

Past Level [Release 8.0]: 65 mins.,

Tolerable Limit [Release 8.5]: 35 mins.,

Goal [Release 8.5]: 25 mins.

Note: end result was actually 20 minutes ☺

Meter [Weekly Step]: Candidates with Reportal experience, and with knowledge of MR-specific reporting features

Trond Johansen

Shift: from Function to Quality

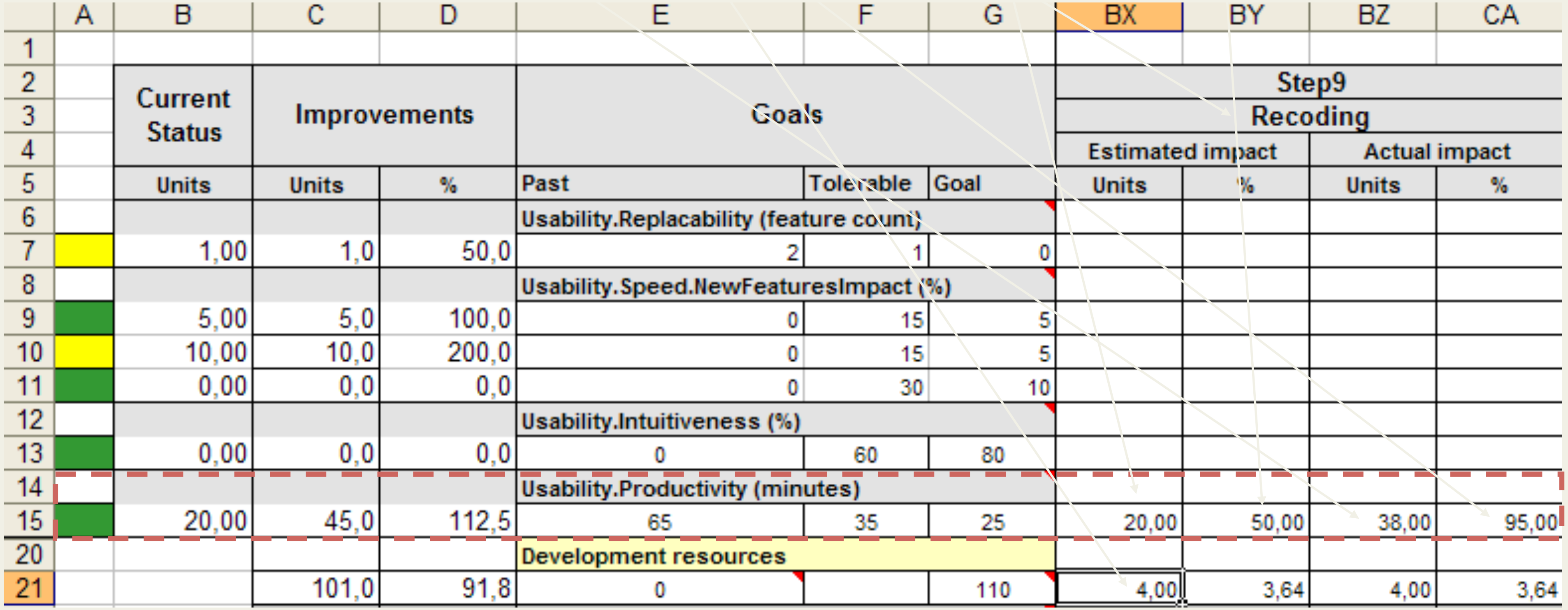
- **Our new focus is on the day-to-day operations of our Market Research users,**
  - **not a list of features that they might or might not like. 50% never used!**
  - **We KNOW that increased efficiency, which leads to more profit, will please them.**
  - **The '45 minutes actually saved x thousands of customer reports'**
    - **= big $$$ saved**
- **After one week we had defined more or less all the requirements for the next version (8.5) of Confirmit.**

confirmit®

FIRM (Future Information Research Management, Norway)
project step planning and accounting:
using **an Impact Estimation Table**

Trond Johansen

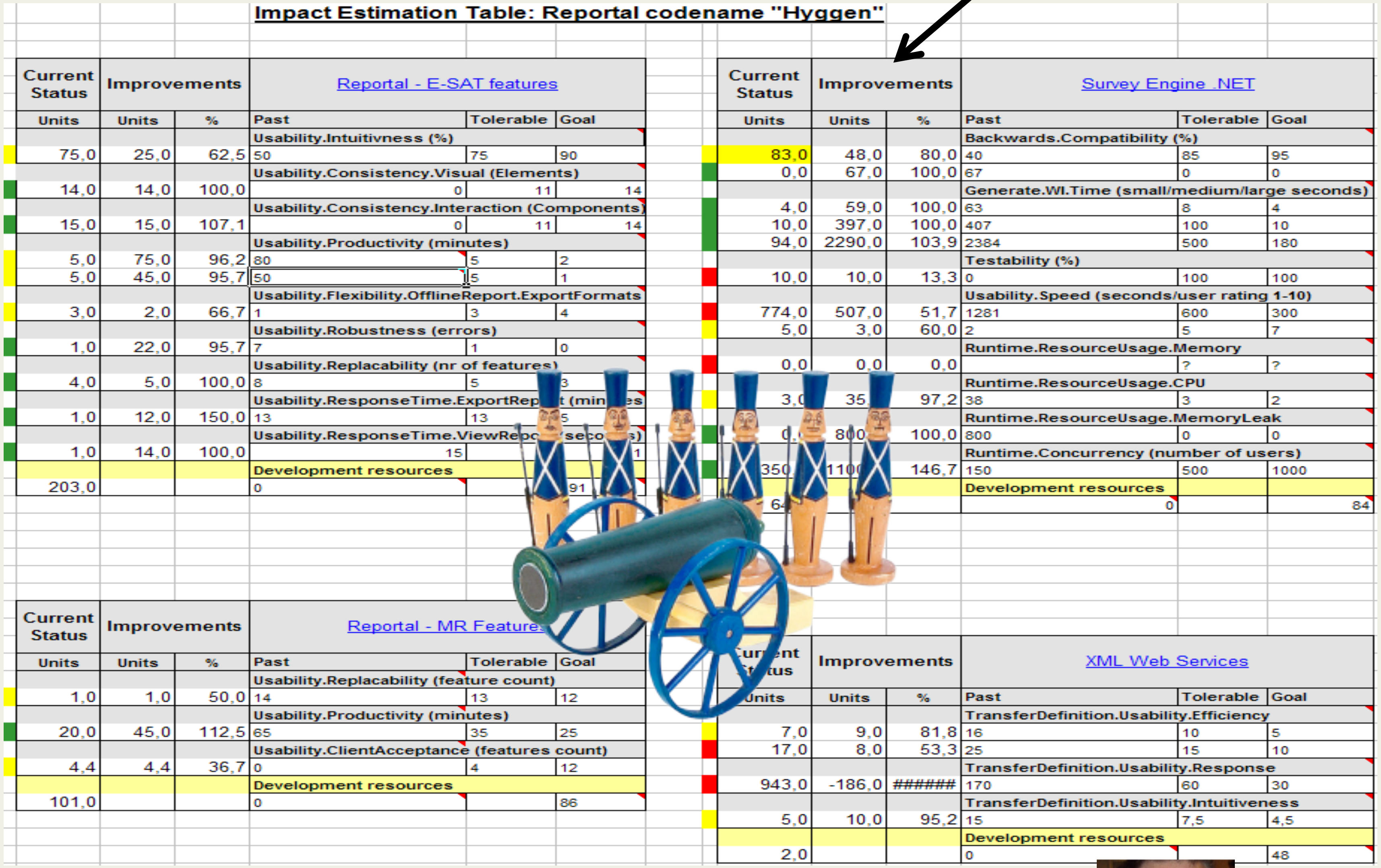
- IET for MR Project – Confirmit (<-FIRM Product Brand) 8.5
- **Solution:** Recoding
  - Make it possible to recode variable on the fly from Reportal.
  - Estimated effort: 4 days
  - **Estimated** Productivity Improvement: 20 minutes (50% way to Goal)
  - actual result 38 minutes (95% progress towards 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	95,00
20					Development resources						
21			101,0	91,8	0		110	4,00	3,64	4,00	3,64

EVO Plan Confirmit 8.5 in **Evo Step Impact *Measurement***

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



Impact Estimation Table: Reportal codename "Hyggen"

9

	Current Status	Improvements		Reportal - E-SAT features		
	Units	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.ExportRep[illegible]t (min[illegible]es		
🟩	1,0	12,0	150,0	13	13	5
				Usability.ResponseTime.ViewRepo[illegible]seco[illegible]s)		
🟩	1,0	14,0	100,0	15	[illegible]	1
				Development resources		
	203,0			0		91

8

	Current Status	Improvements		Survey Engine .NET		
	Units	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,[illegible]	35,[illegible]	97,2	38	3	2
				Runtime.ResourceUsage.MemoryLeak		
🟩	0,[illegible]	800	100,0	800	0	0
				Runtime.Concurrency (number of users)		
🟩	350,[illegible]	110[illegible]	146,7	150	500	1000
				Development resources		
	64[illegible]			0		84

3

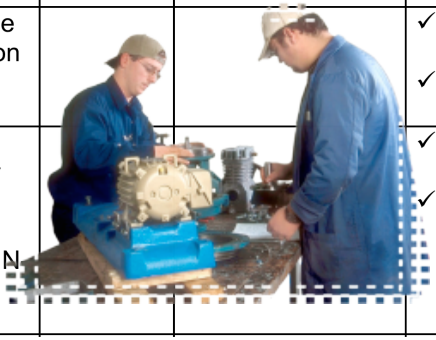
	Current Status	Improvements		Reportal - MR Feature[illegible]		
	Units	Units	%	Past	Tolerable	Goal
				Usability.Replacability (feature count)		
🟨	1,0	1,0	50,0	14	13	12
				Usability.Productivity (minutes)		
🟩	20,0	45,0	112,5	65	35	25
				Usability.ClientAcceptance (features count)		
🟨	4,4	4,4	36,7	0	4	12
				Development resources		
	101,0			0		86

3

	Current Status	Improvements		XML Web Services		
	Units	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

Confirmit **Evo Weekly Value Delivery Cycle**

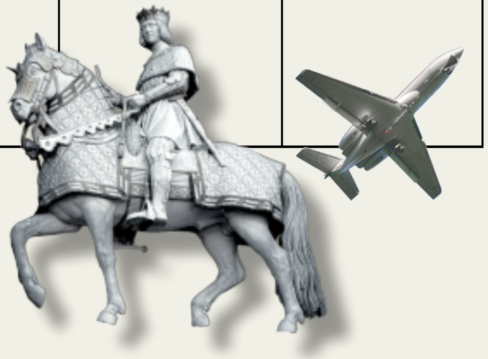
	Development Team	**Users (PMT, Pros, Doc writer, other)**	**CTO (Sys Arch, Process Mgr)**	**QA (Configuration Manager & Test Manager)**
Friday	✓ PM: Send Version N detail plan to CTO + prior to Project Mgmt meeting ✓ PM: Attend Project Mgmt meeting: 12.00-15.00 ✓ Developers: Focus on genereal maintenance 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 release 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	✓ System Architect to review code and test code	✓ Follow up CI ✓ Review test plans, tests
Wednesday	✓ Develop test code & code for Version N			✓ Review test plans, tests ✓ Follow up CI
Thursday	✓ Complete Test Code & Code for Version N ✓ Complete GUI tests for Version N-2			✓ Review test plans, tests ✓ Follow up CI



Evo's impact on Confirmit product qualities 1st Qtr

- Only 5 highlights of the 25 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 Confirmit 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



confirmit

Release 8.5

Initial Experiences and conclusions

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

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

confirmit

Trond Johansen

Conclusions -

The Evo 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.

Trond Johansen

Evo's impact on Confirmit 9.0 product qualities
Results from the second quarter of using Evo. 1/2

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%**

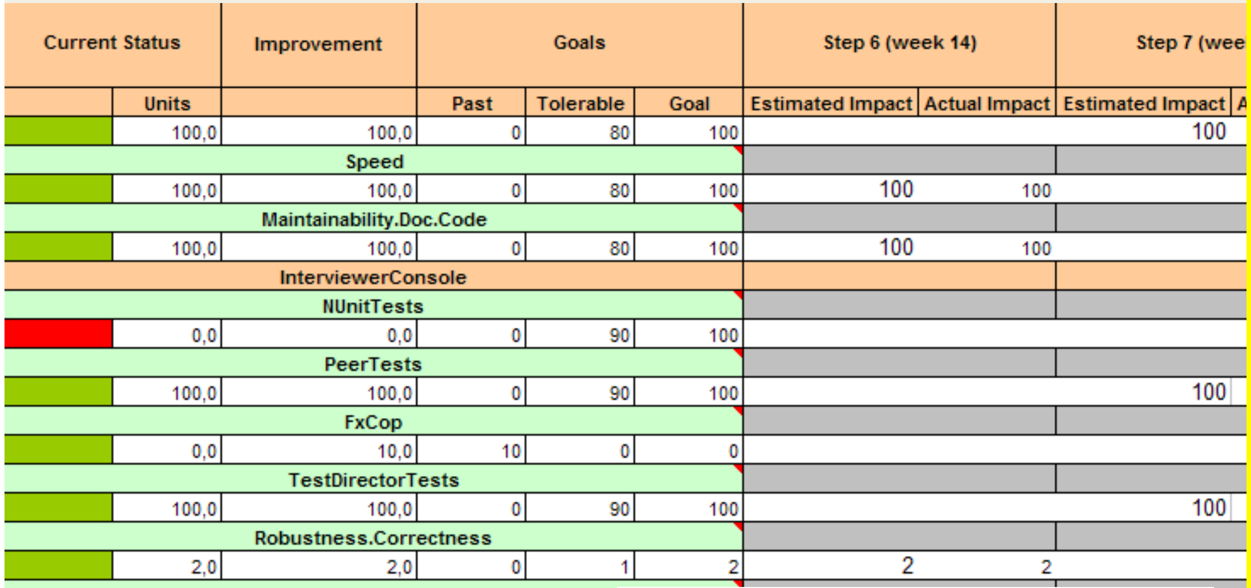
Product quality	Description	Customer value
Productivity	Time (in minutes) to test a defined survey and identify 4 inserted script errors, starting from when the questionnaire is finished to the time testing is complete and is ready for production. (Defined Survey: Complex survey, 60 questions, comprehensive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

Evo's impact on Confirmit 9.0 product qualities
Results from the second quarter of using Evo. 2/2

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%**

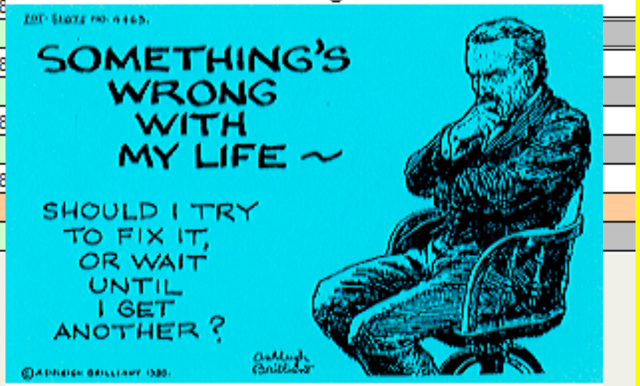
Code quality – ”green” week

- 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.



Current Status		Improvement	Goals			Step 6 (week 14)		Step 7 (wee	
	Units		Past	Tolerable	Goal	Estimated Impact	Actual Impact	Estimated Impact	A
🟩	100,0	100,0	0	80	100			100	
Speed									
🟩	100,0	100,0	0	80	100	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			100	
FxCop									
🟩	0,0	10,0	10	0	0				
TestDirectorTests									
🟩	100,0	100,0	0	90	100			100	
Robustness.Correctness									
🟩	2,0	2,0	0	1	2	2	2		
Robustness.BoundaryConditions									
🟥	0,0	0,0	0	8					
Speed									
🟥	0,0	0,0	0	8					
ResourceUsage.CPU									
🟥	100,0	0,0	100	8					
Maintainability.Doc.Code									
🟩	100,0	100,0	0	8					
SynchronizationStatus									
NUnitTests									

POT-SHOTS — Brilliant Thoughts in 17 words or less


POT-SHOTS NO. 4465.
SOMETHING'S WRONG WITH MY LIFE ~
SHOULD I TRY TO FIX IT, OR WAIT UNTIL I GET ANOTHER ?
Ashleigh Brilliant

© Ashleigh Brilliant www.ashleighbrilliant.com

Speed

Maintainability

Nunit Tests

PeerTests

TestDirectorTests

Robustness.Correctness

Robustness.Boundary Conditions

ResourceUsage.CPU

Maintainability.DocCode

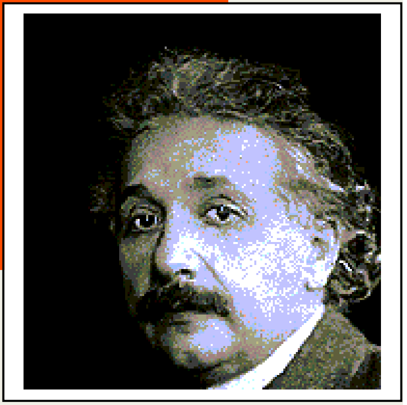
SynchronizationStatus

Al Says

“Not everything that can be counted counts,

and not everything that counts can be counted.”

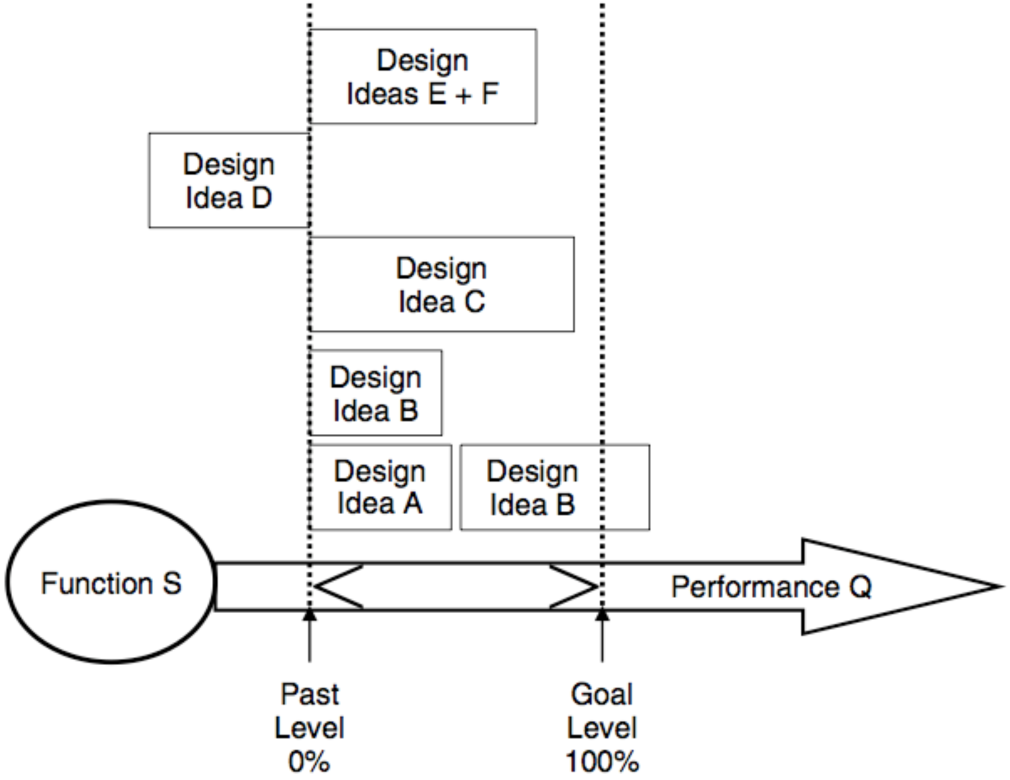
Albert Einstein



I agree.

But, system qualities can be ‘counted’. Tom

How do we evaluate a single quality dimension of design impact?



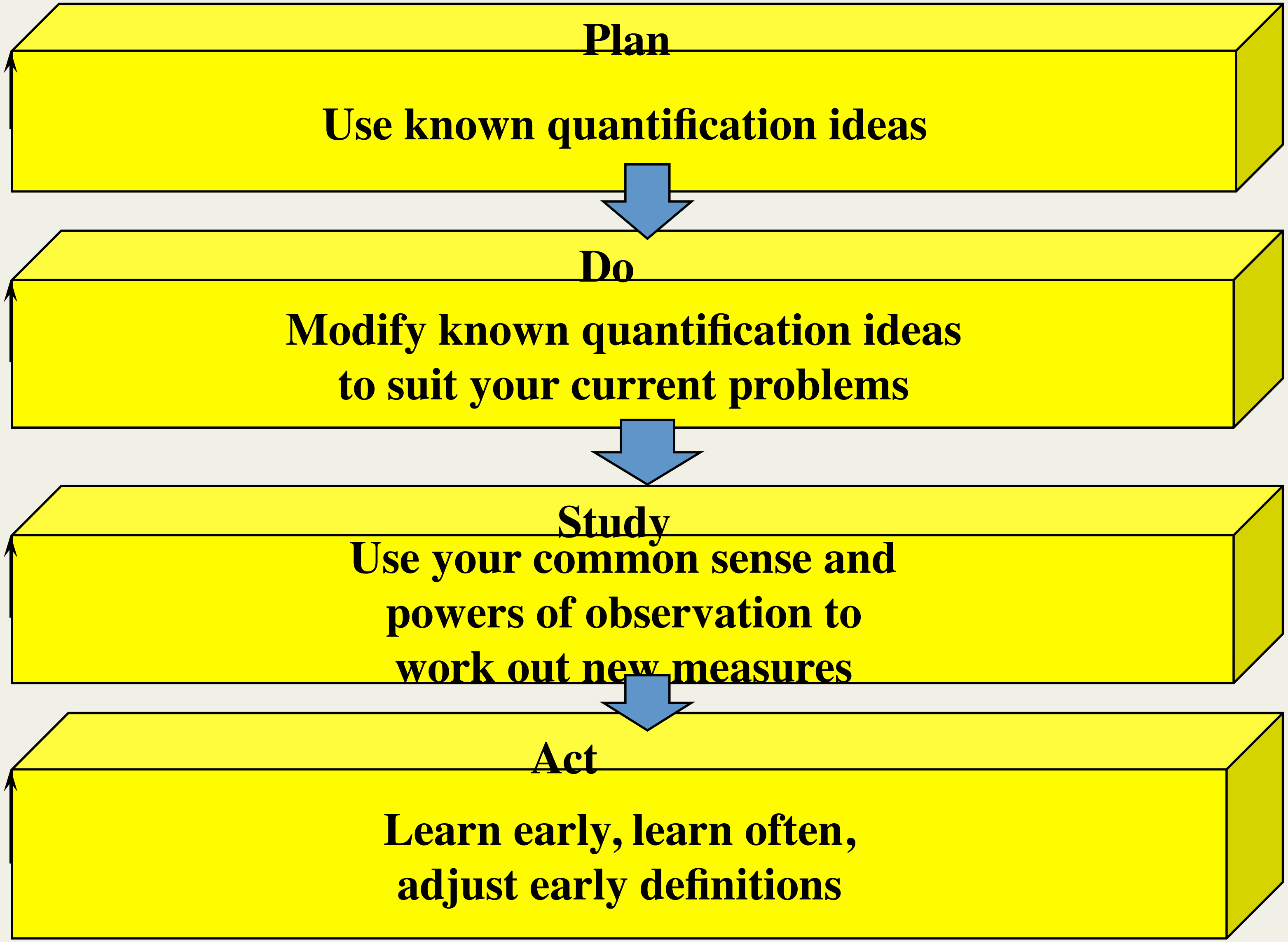
- We must estimate
- (or measure)
- the numeric cumulative impact
- of the design
  - on a defined Scale (units),
  - using a defined Meter (test process),
  - with respect to requirement levels.

How can we evaluate all dimensions of impact?
All = {Capacity, Quality, Costs}

Design Ideas **Objectives**	*Central*	*Youth*	*Facts*	*London*	*Diploma*	*Events*	*Discounts*	Total
Participation	80%±50%	60%±70%	0%±50%	0%±50%	30%±50%	20%±50%	30%±50%	220%±370%
Representation	80%±50%	80%±50%	10%±50%	0%±50%	10%±50%	20%±50%	50%±40%	250%±340%
Information	0%±50%	20%±40%	80%±50%	0%±20%	20%±50%	0%±50%	0%±30%	120%±290%
Conviction	0%±10%	20%±50%	60%±30%	80%±50%	10%±50%	80%±50%	0%±50%	250%±290%
Influence	0%±50%	40%±40%	60%±50%	0%±50%	80%±50%	80%±50%	0%±50%	260%±340%
Fun	50%±50%	40%±50%	10%±50%	0%±0%	0%±0%	80%±50%	0%±0%	180%±200%
Total	210% ±260%	260% ±300%	220% ±280%	80% ±220%	150% ±250%	270% ±300%	80% ±220%	
Budgets								
Cost	10%	10%	10%	10%	1%±5%	50%±50%	80%±50%	171%±105%
Benefit–to-Cost Ratio	210%/10%	260%/10%	220%/10%	80%/10%	150/1	270/50	80/80	

- **We can use an Impact (Estimation) Table**

How to Quantify Quality



Plan

Use known quantification ideas

Do

Modify known quantification ideas to suit your current problems

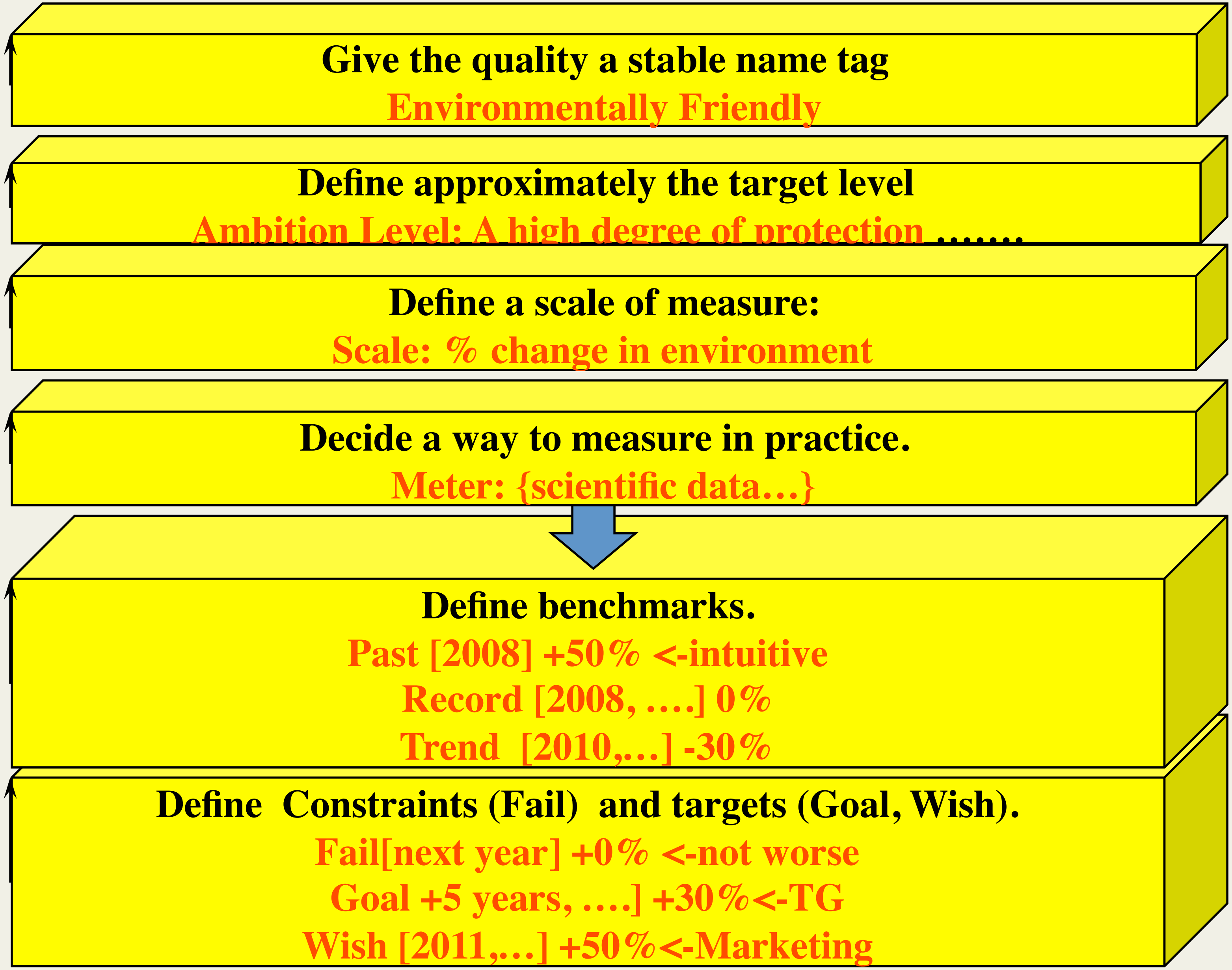
Study

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

Act

Learn early, learn often, adjust early definitions

‘Environmentally Friendly’ Quantification Example



Give the quality a stable name tag

Environmentally Friendly

Define approximately the target level

Ambition Level: A high degree of protection …….

Define a scale of measure:

Scale: % change in environment

Decide a way to measure in practice.

Meter: {scientific data…}

Define benchmarks.

Past [2008] +50% <-intuitive

Record [2008, ….] 0%

Trend [2010,…] -30%

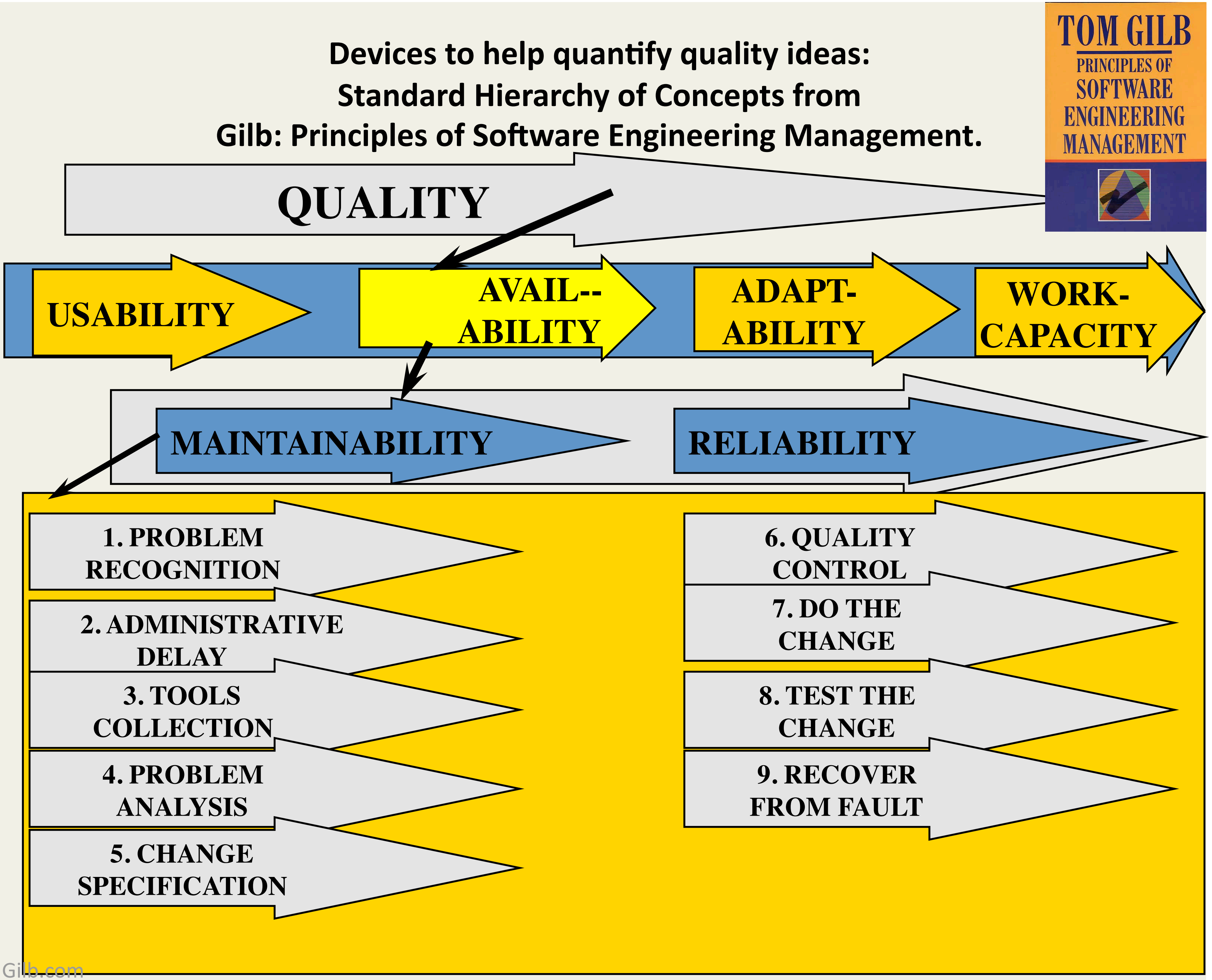
Define Constraints (Fail) and targets (Goal, Wish).

Fail[next year] +0% <-not worse

Goal +5 years, ….] +30%<-TG

Wish [2011,…] +50%<-Marketing

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



QUALITY

USABILITY

AVAIL--ABILITY

ADAPT-ABILITY

WORK-CAPACITY

MAINTAINABILITY

RELIABILITY

1. PROBLEM RECOGNITION

2. ADMINISTRATIVE DELAY

3. TOOLS COLLECTION

4. PROBLEM ANALYSIS

5. CHANGE SPECIFICATION

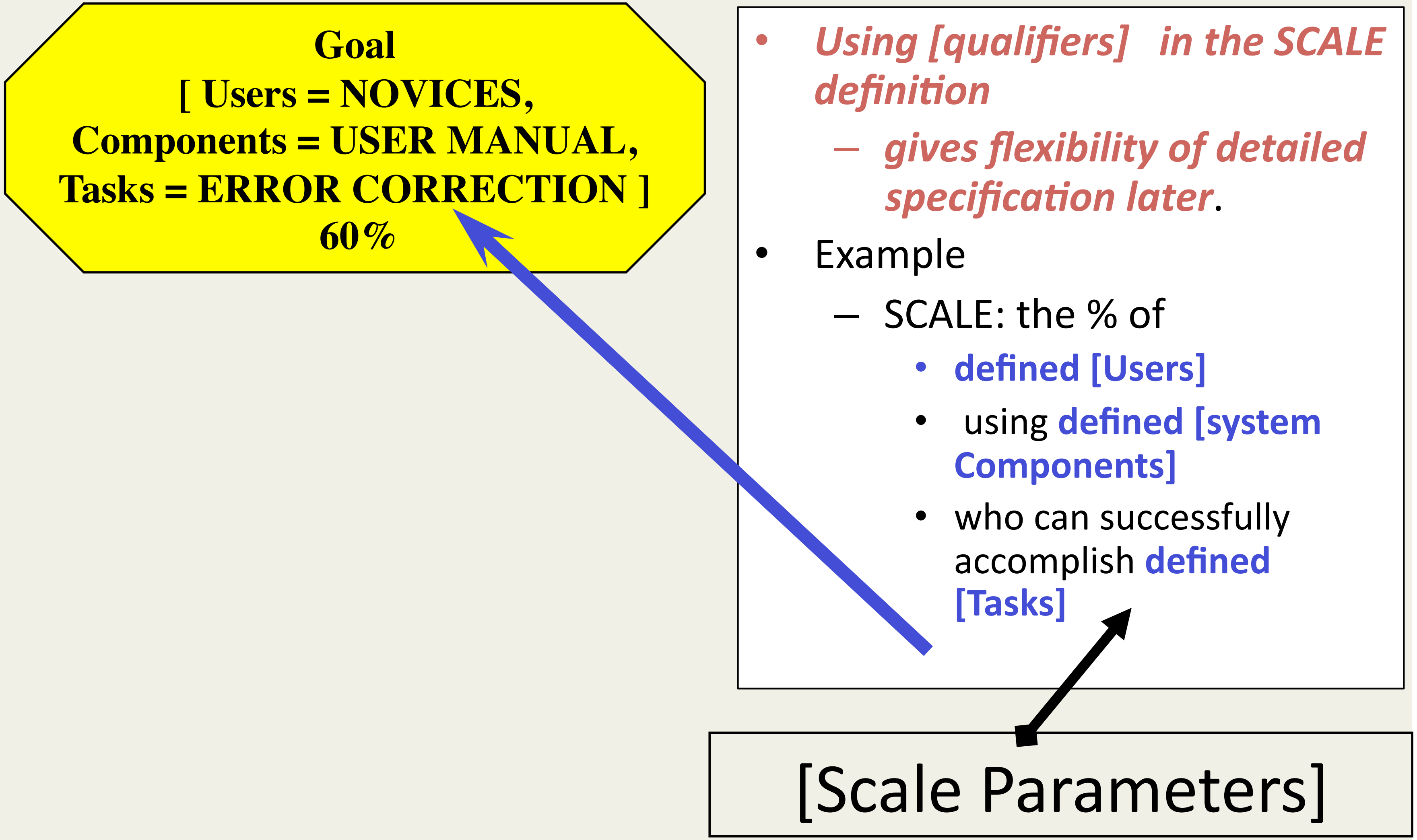
6. QUALITY CONTROL

7. DO THE CHANGE

8. TEST THE CHANGE

9. RECOVER FROM FAULT

Using ‘Parameters’ when defining a Scale of Measure



Goal
[Users = NOVICES,
Components = USER MANUAL,
Tasks = ERROR CORRECTION]
60%

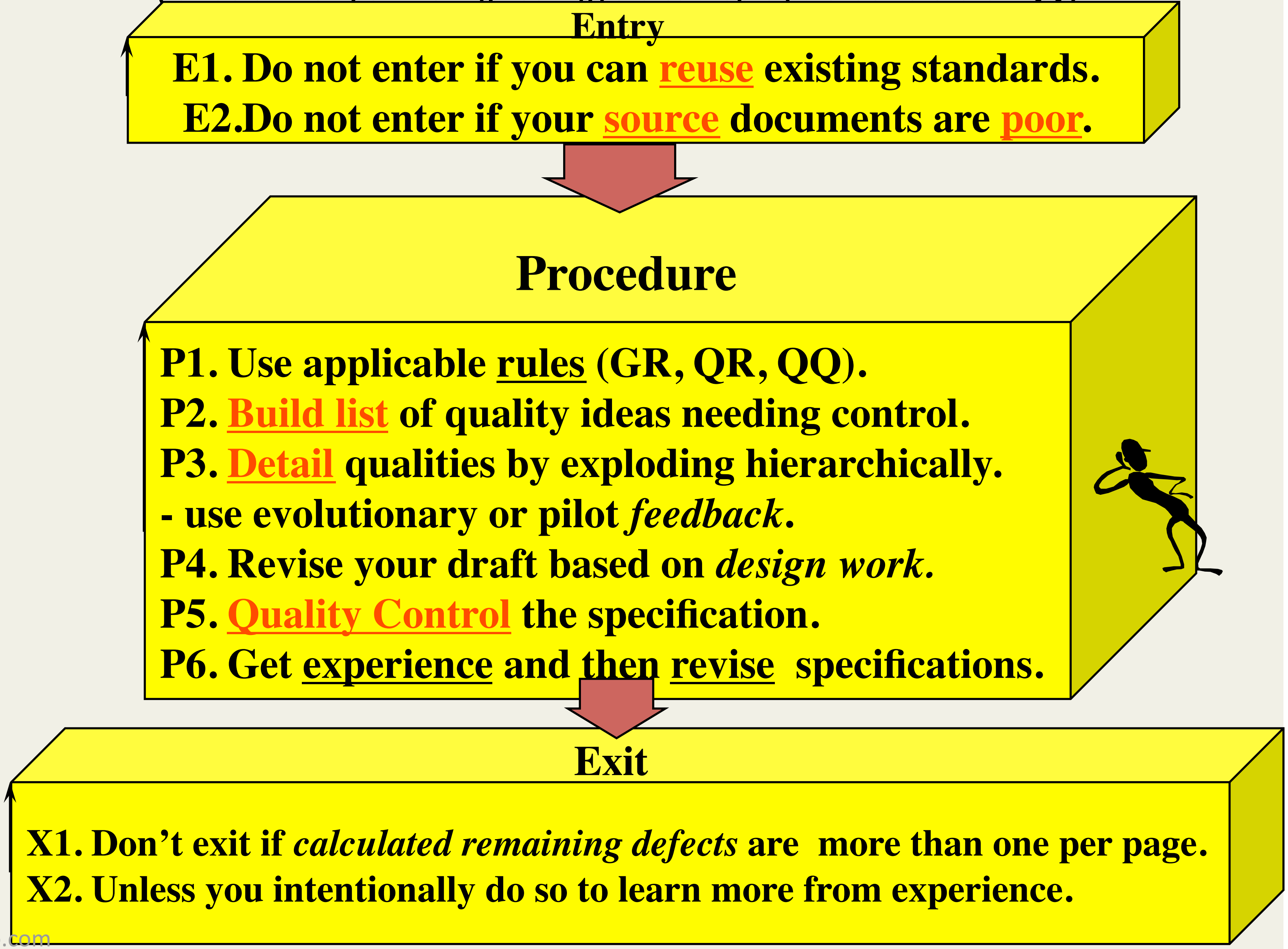
- *Using [qualifiers] in the SCALE definition*
  - *gives flexibility of detailed specification later*.
- Example
  - SCALE: the % of
    - **defined [Users]**
    - using **defined [system Components]**
    - who can successfully accomplish **defined [Tasks]**

[Scale Parameters]

© Tom@Gilb.com www.gilb.com

Quality Quantification Process

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



Entry

E1. Do not enter if you can <u>reuse</u> existing standards.

E2.Do not enter if your <u>source</u> documents are <u>poor</u>.

Procedure

P1. Use applicable <u>rules</u> (GR, QR, QQ).

P2. <u>Build list</u> of quality ideas needing control.

P3. <u>Detail</u> qualities by exploding hierarchically.

- use evolutionary or pilot *feedback*.

P4. Revise your draft based on *design work.*

P5. <u>Quality Control</u> the specification.

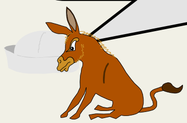
P6. Get <u>experience</u> and <u>then</u> <u>revise</u> specifications.

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.

A ‘Quality Quantification’ Principle

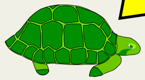


He had a lot of hats.
He wants to be best in hatmanship.

0. *THE PRINCIPLE OF*

'BAD NUMBERS BEAT GOOD WORDS'

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



Scale: hats on his head.
Past:3
Goal: 13

General Hatmanship:

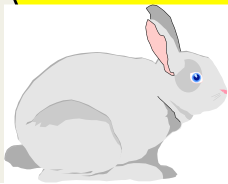
Ambition Level: improve ability to have hats on head and nearby

Hatmanship On Head:

SCALE: hats on top of persons head
PAST [Me, This year] 10 <- Guess
RECORD [2009, UK] 15 <- GB Record
WISH [Guinness Record, April] 20 <- Tom

Hatmanship Nearby:

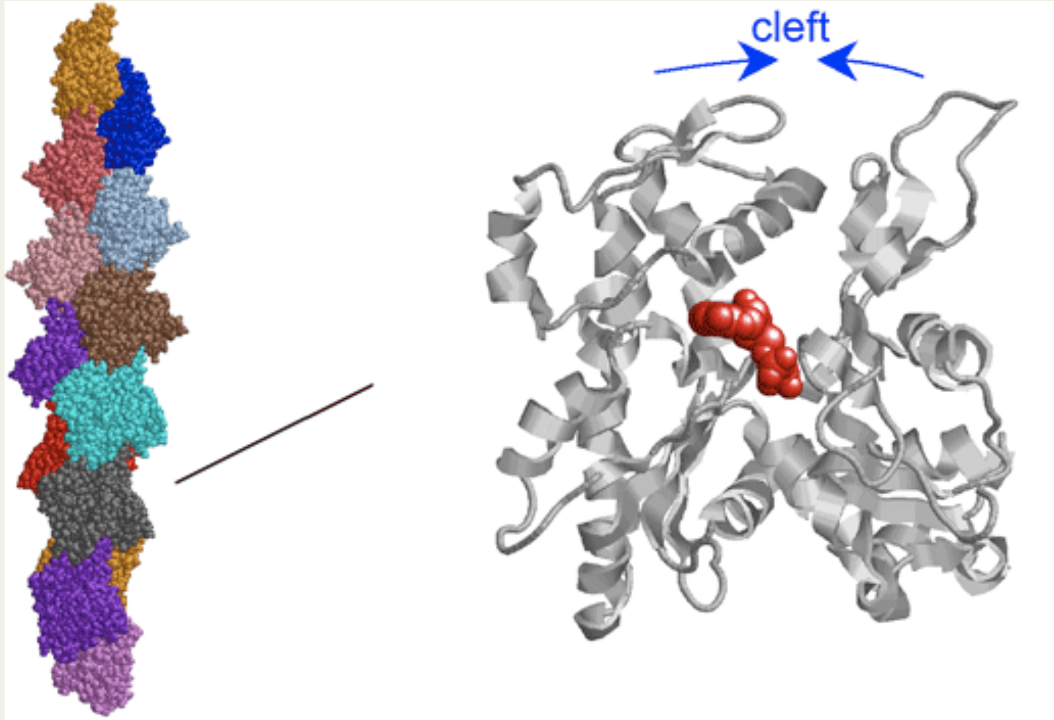
SCALE: hats not on head, but on, or near, body;within 10 meter radius.
Past…. Goal……..etc.



Quantify for realistic judgements

- *← R. H. Hayes et al*
- *"To leave [soft considerations] out of the analysis*
  - *simply because they are* ***not readily quantifiable***
  - *or to avoid introducing "personal judgments,"*
  - *clearly <u>biases decisions against</u> 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"*
- *"Dynamic Manufacturing", p. 77*
- *in MINTZBERG94: page124*

Principles for Quality Quantification.



The Decomposition Principle

- Some hopefully deep and useful guidelines
- to help you quantify quality ideas

© Tom@Gilb.com www.gilb.com

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

- *Poor* quantification is more useful than none;   66±8
- at least it can be improved systematically. 65±2

State of the Art Flexibility

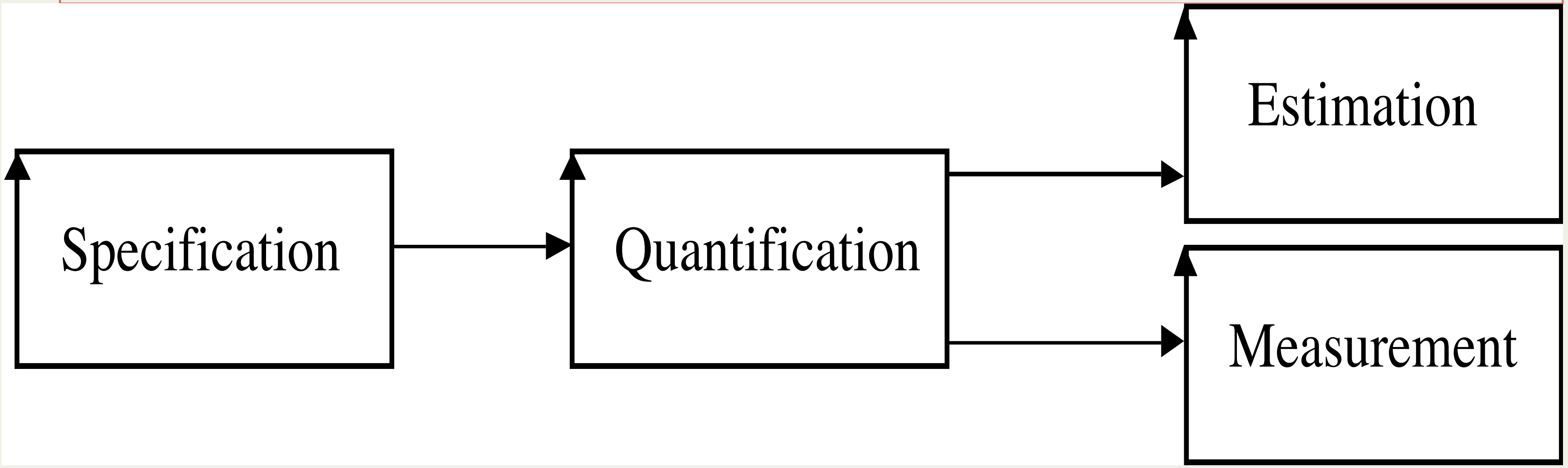
Not Clear!

Enhanced Usability

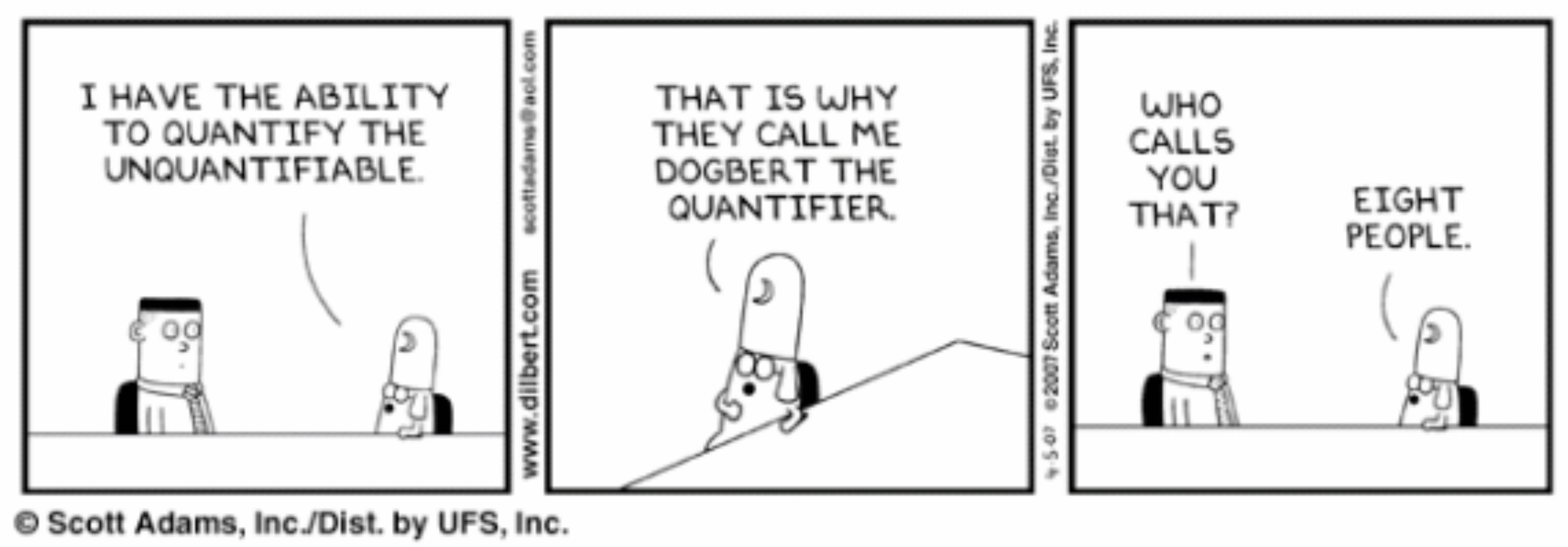
Improved Performance

1. THE PRINCIPLE OF 'QUALITY QUANTIFICATION'

- All qualities can be expressed **quant**itatively,
- *'qualitative'* does *not* mean **un**measurable.



Dogbert the Quantifier


I HAVE THE ABILITY TO QUANTIFY THE UNQUANTIFIABLE.
scottadams@aol.com
www.dilbert.com
THAT IS WHY THEY CALL ME DOGBERT THE QUANTIFIER.
© 2007 Scott Adams, Inc./Dist. by UFS, Inc.
9-5-07
WHO CALLS YOU THAT?
EIGHT PEOPLE.

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

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;

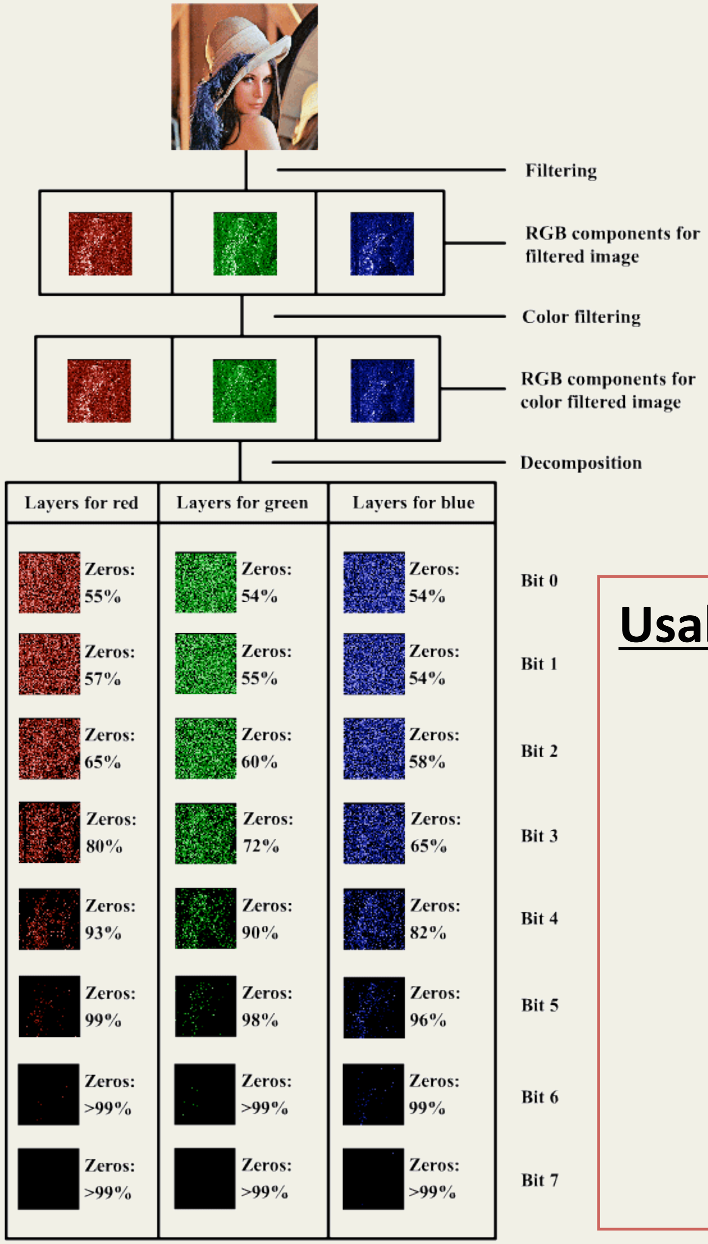
it may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the state of Science, whatever the matter may be."

Lord Kelvin, 1893

from

http://zapatopi.net/kelvin/quotes.html

2. THE PRINCIPLE OF 'MANY SPLENDORED THINGS’

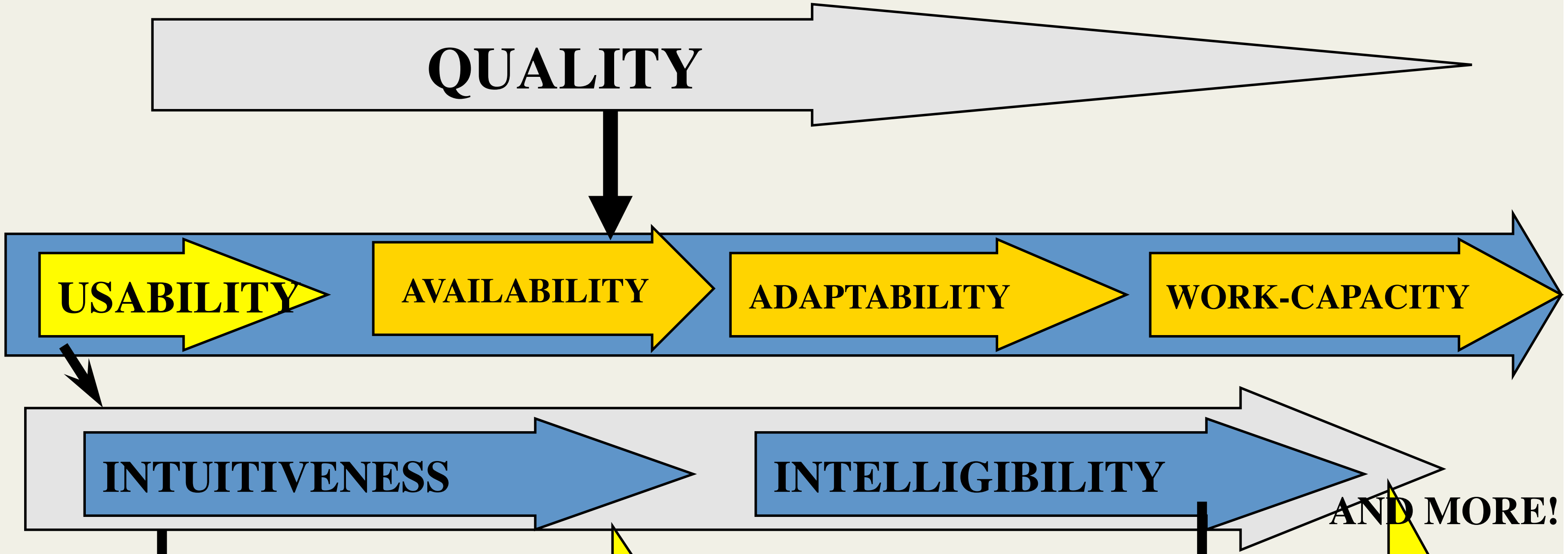


Most quality ideas are usefully broken down into *several* measures of goodness.

Usability: Includes:

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

Quantifying Usability (Erieye C&C System)



QUALITY

USABILITY

AVAILABILITY

ADAPTABILITY

WORK-CAPACITY

INTUITIVENESS

INTELLIGIBILITY

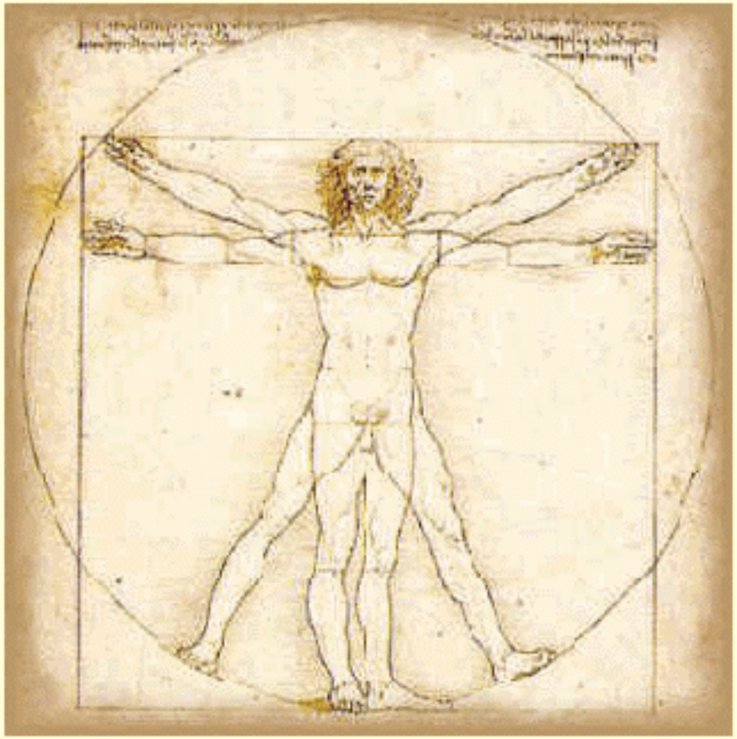
AND MORE!

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

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

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.

3. THE PRINCIPLE OF 'SCALAR DEFINITION'



Leonardo da Vinci

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: Constraints, Targets

<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>**

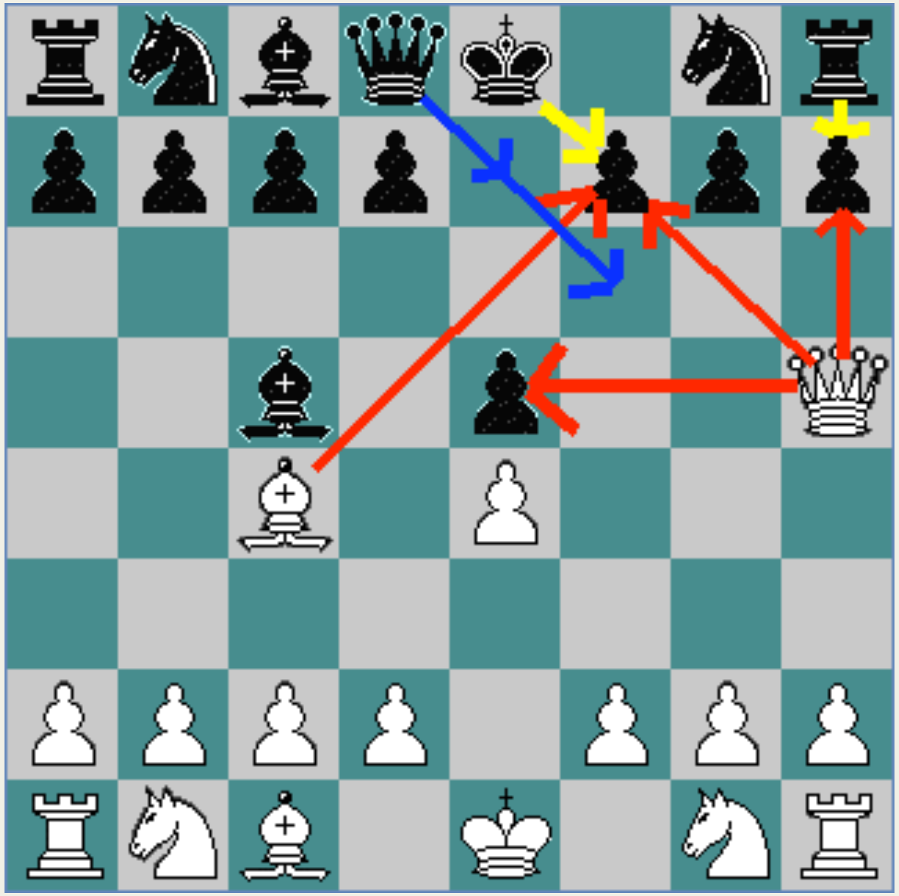
Stretch **[   ] <motivating ambition level>   <-- <source of level>**

========== Constraints ========================

Fail **[   ]   <-- <source>   'Failure Point'**

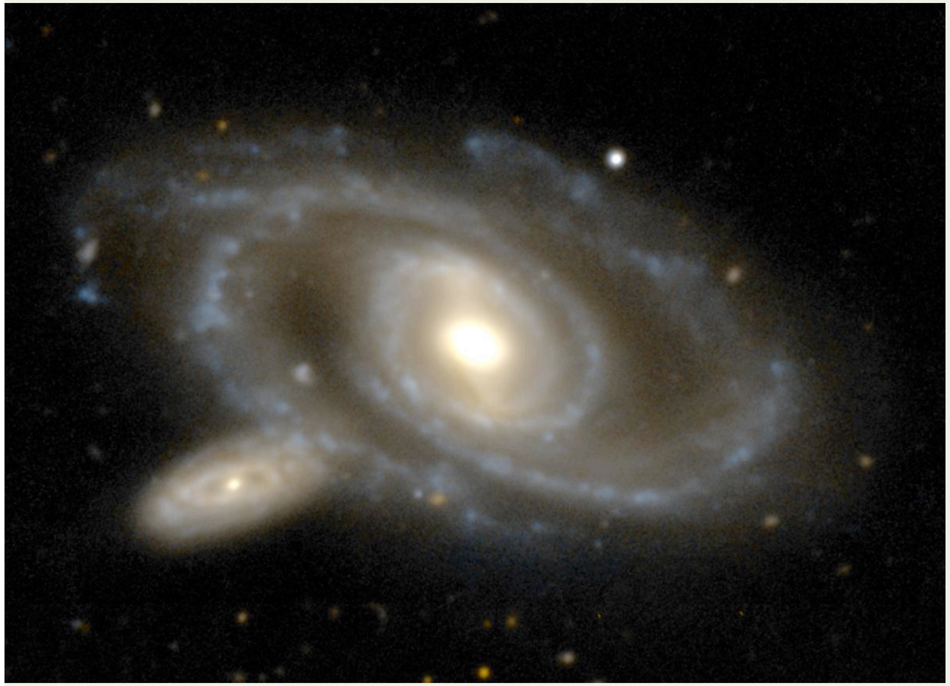
Survival **[   ] <- <source of limit>   'Survival Point'**

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'

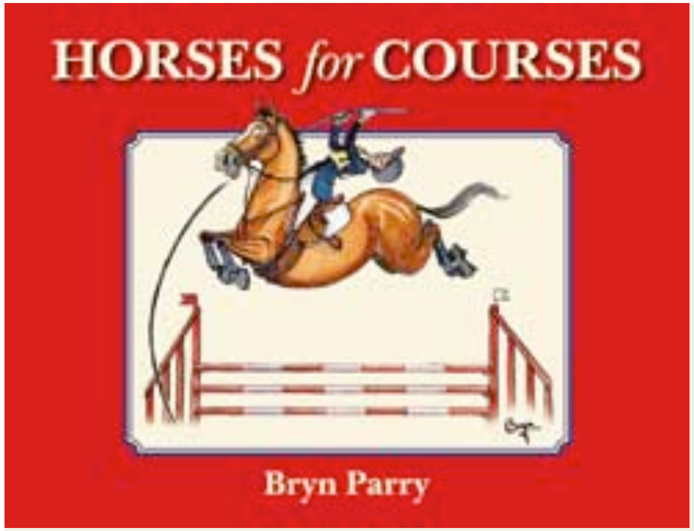
Practical measuring instruments improve the *practical understanding* and *application* of ‘Scales of measure’.

Portability:

Scale: Cost to convert/Module

Meter [Data] measure/1,000 words converted

Meter [Logic] measure/1,000 Function Points Converted



7. THE PRINCIPLE OF 'HORSES FOR COURSES'

Different quality-Scale *measuring processes*

will be necessary

for different *points in time*, different *events* and different *places*.

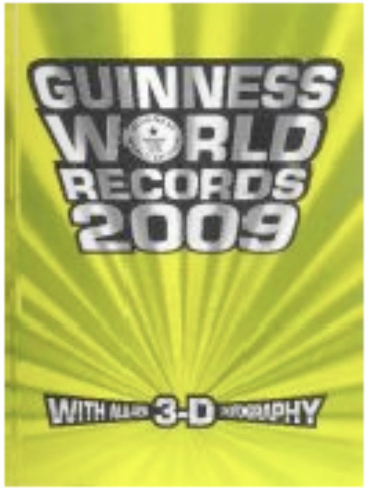
Availability:

Scale: % Uptime for System

Meter *[USA, 2011] Test X*

Meter *[UK, 2012] 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, 2008] 30,000 Hours

Trend [Nato Allies, 2012] 50,000 Hours

improve

Goal [UK MOD, 2011] 60,000 Hours

9. THE PRINCIPLE OF 'NUMERIC FUTURE'

Numeric future requirement levels

{Wish, Stretch, Goal, Ideal}

complete a clear quality definition of relative terms like '**improved**'.

Usability:

Scale: Time to learn average task.

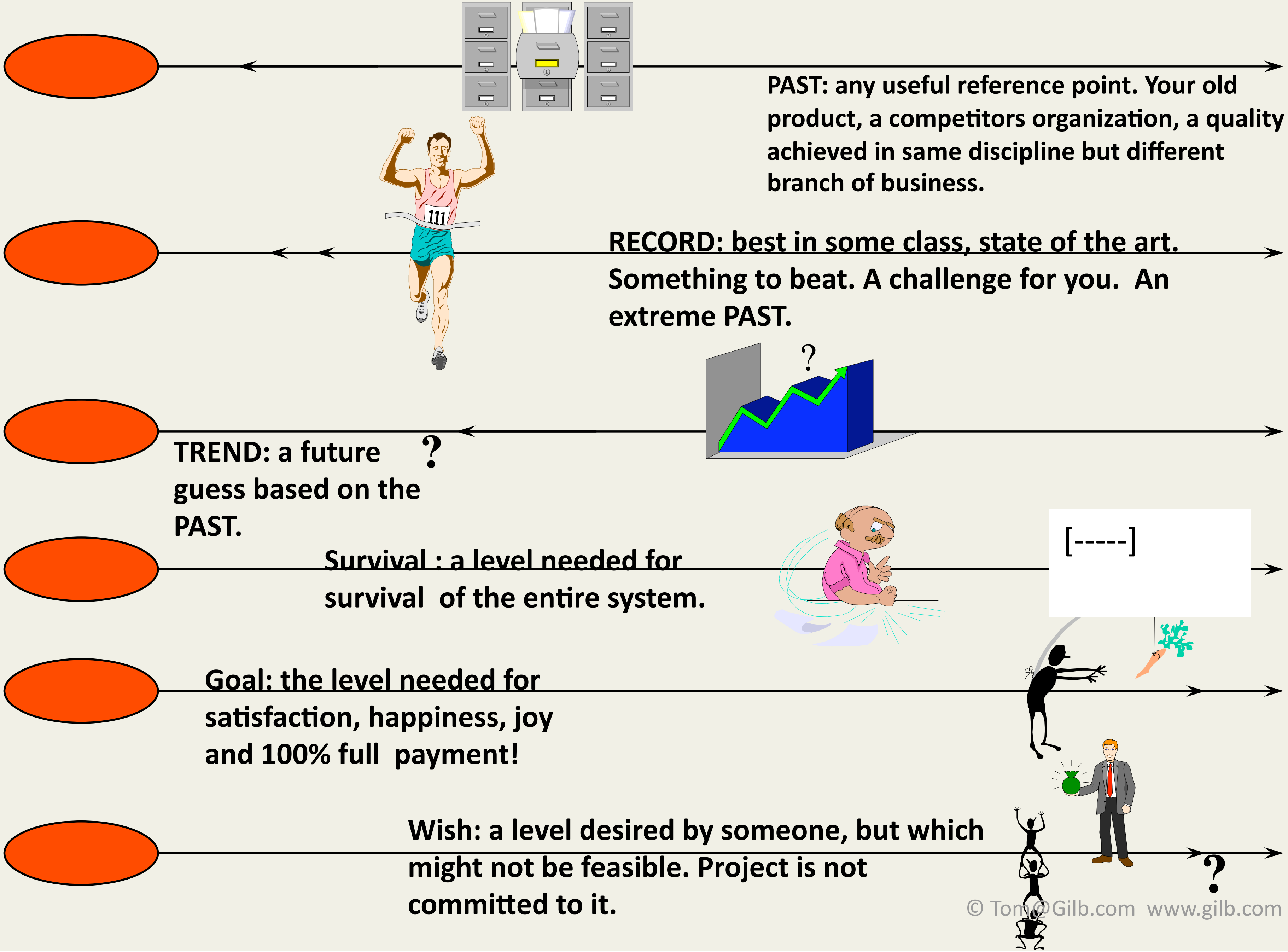
Past [Old product, 2008] 20 minutes

Wish [New product, 2011] 1 minute

Stretch [End 2012, Students] 2 minutes

Goal [End 2013, Teachers] 5 minutes

Some Planguage ‘Quality Quantification’ Concepts



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

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

TREND: a future guess based on the PAST.

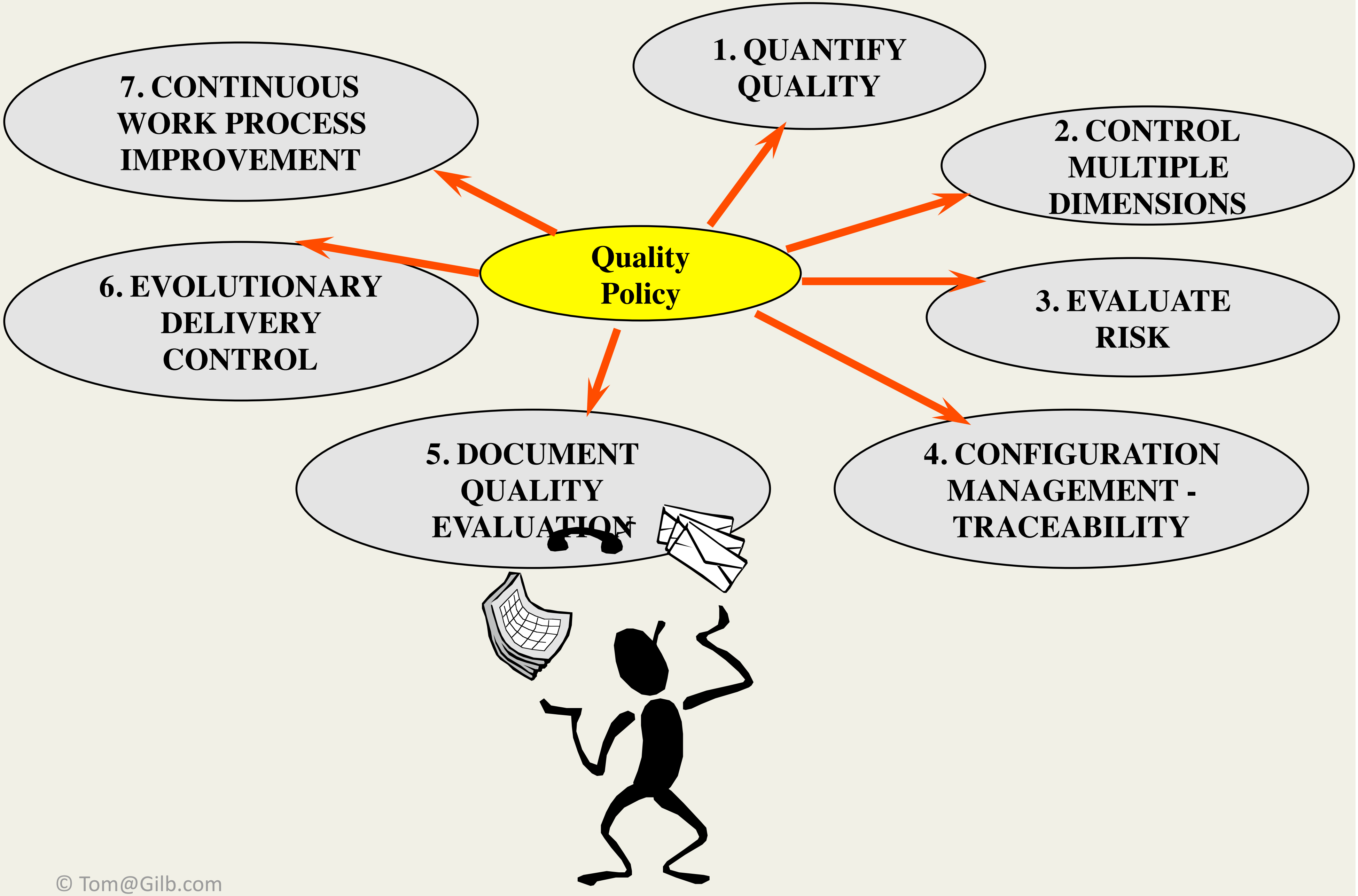
Survival : a level needed for survival of the entire system.

[-----]

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

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

A Corporate Quality Policy (Euro Multinational)



Quality Policy

1. QUANTIFY QUALITY

2. CONTROL MULTIPLE DIMENSIONS

3. EVALUATE RISK

4. CONFIGURATION MANAGEMENT - TRACEABILITY

5. DOCUMENT QUALITY EVALUATION

6. EVOLUTIONARY DELIVERY CONTROL

7. CONTINUOUS WORK PROCESS IMPROVEMENT

Policy on QUANTIFICATION, CLARIFICATION AND TESTABILITY OF CRITICAL OBJECTIVES:

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

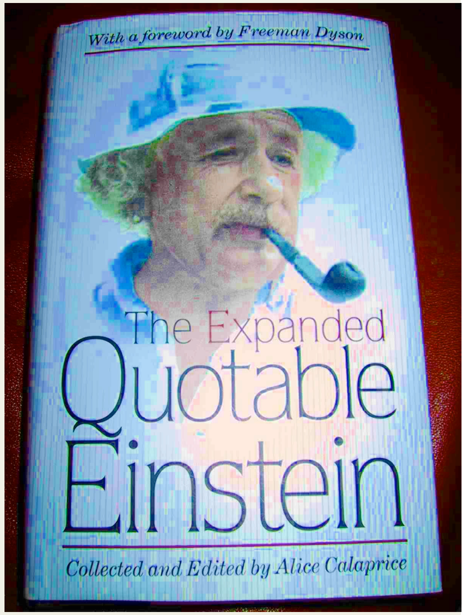
<- (Quality Manual Source is) 5.2.2, 4.1.2, 4.1.5, 5.1.1, 6.1, 6.4.1, 7.1.1, 7.3 and many others.

Einstein on Stretching

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

"We have to do the best we can.

This is our sacred human responsibility" (1940)


With a foreword by Freeman Dyson

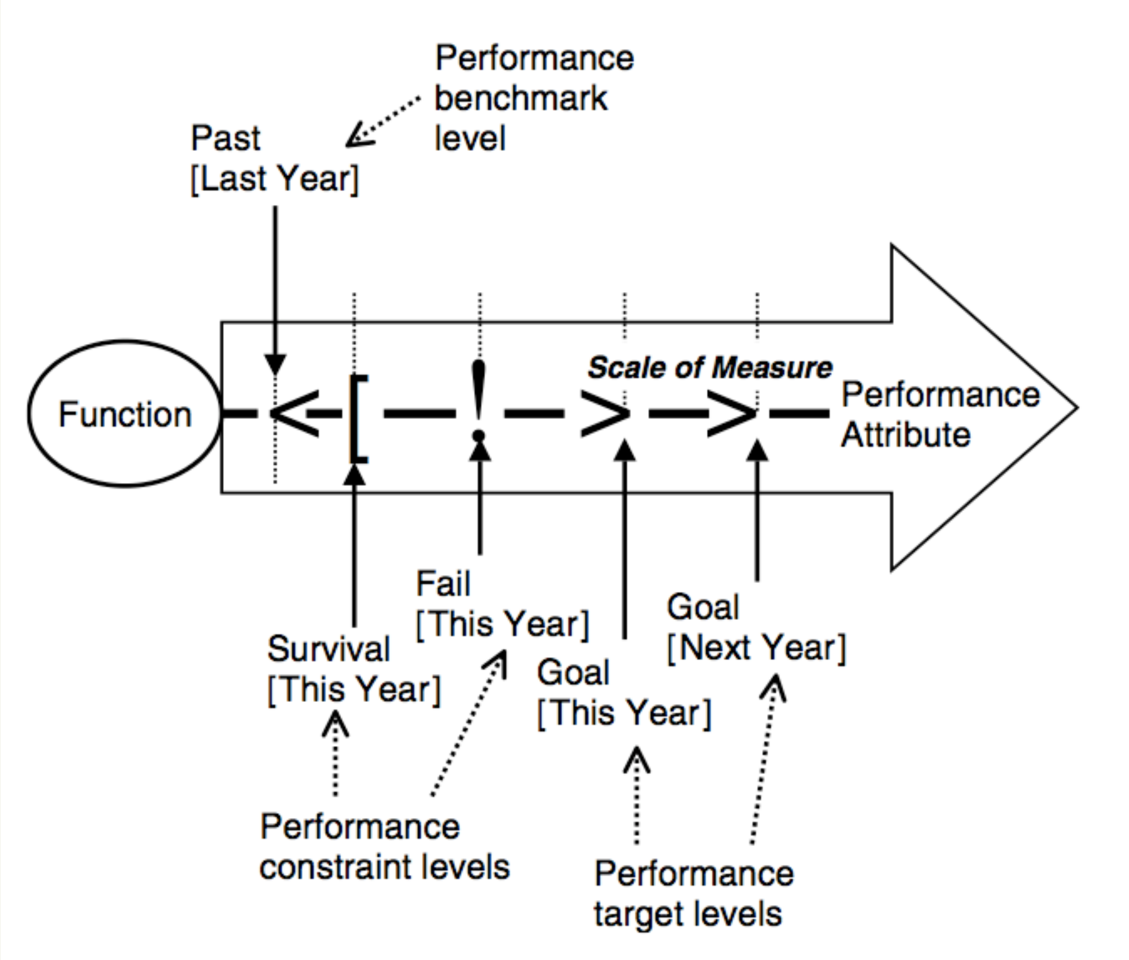
The Expanded Quotable Einstein

Collected and Edited by Alice Calaprice

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

Priority Management

- **Priority is**
  - **Claim on scarce or limited resources**
- **Is a function of**
  - **Constraint type (Survival, ..)**
  - **Target type (Goal, ..)**
  - **Remaining gap to constraint or target level & [qualifiers]**
  - **Remaining budgeted resources; and their constraint and target levels**
- **Priority is dynamically computable!**
- **Priority is also related to other specification parameters such as**
  - **Authority**
  - **Sponsor**
  - **Source**

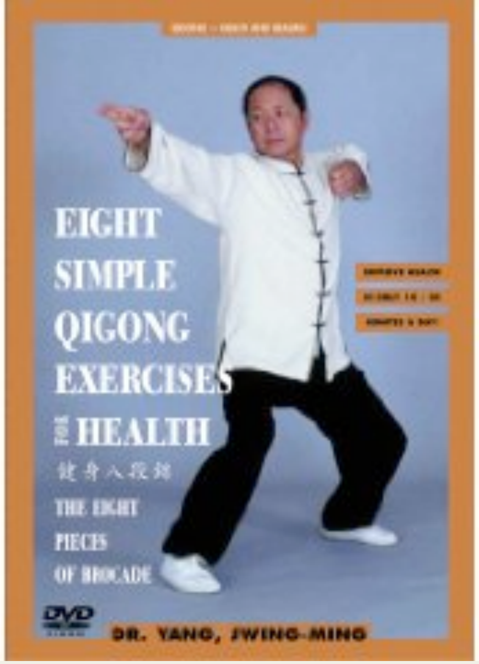


Choice and P…	http://www.gilb.com/tiki-download_file.php?fileId=48

Mng Priorities	http://www.gilb.com/tiki-download_file.php?fileId=60

Quantification Book

- Tom Gilb,
  - Competitive Engineering:
    - A Handbook For Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage
  - ISBN 0750665076 Publisher:
  - Elsevier Butterworth-Heinemann
  - 2 free sample chapters at Gilb.com



1400-1550

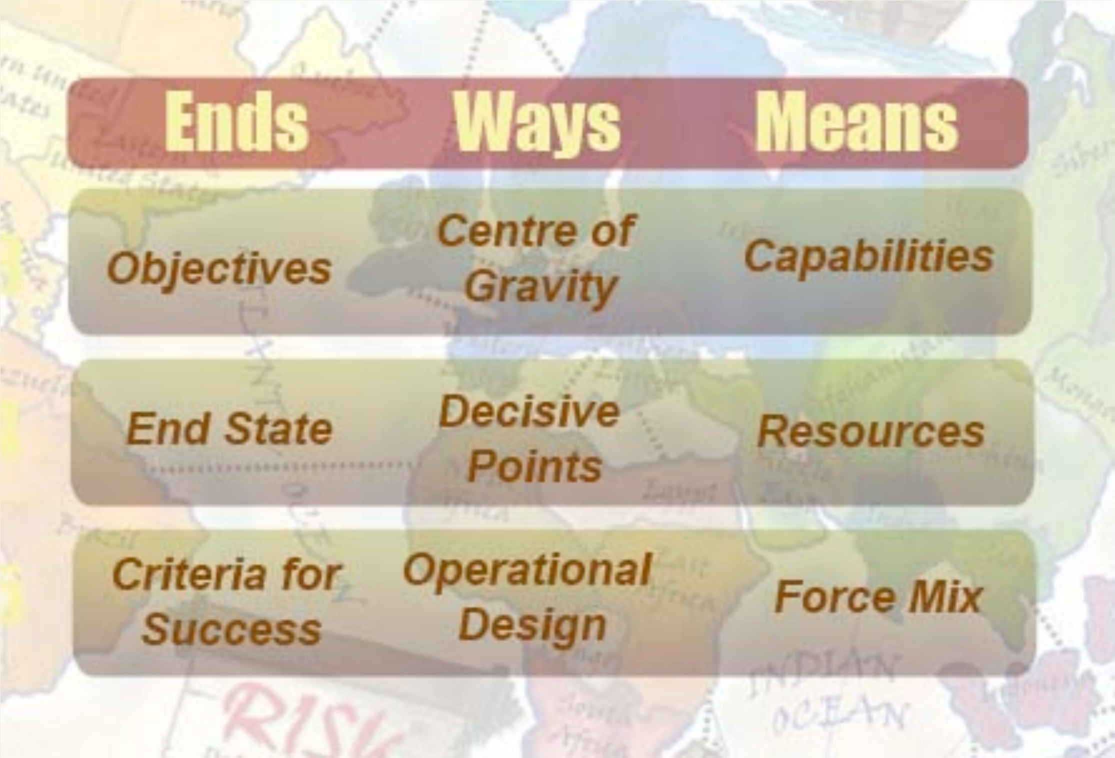
Simple workshop exercises

1. identifying quality requirements

- Write down some requirements ideas
- The most critical ones for your project
- What is expected by managers who gave your project a budget?
- 4 minutes

2. indirect identification (means ->ends)

- Ask Why? For each requirement!
- If you get an answer , you might have indirectly found the ‘real’ requirement
- 5 minutes



Ends	Ways	Means
Objectives	Centre of Gravity	Capabilities
End State	Decisive Points	Resources
Criteria for Success	Operational Design	Force Mix

3. Naming – the tag –

- Give short distinctive descriptive name for each requirement
- Do NOT include words like ‘increased’, ‘improve’, ‘reduce’
- Get to the core idea:
  - Reliability
  - Usability
  - Security

Christening for

Allira

Amy May

Please join us for the Christening of our daughter
Sunday 22nd April 2007 @ 10am to be held at
Little Dance Ave, Little Dance
RSVP Tammy
on 9754 4559
by 15th April

NC0712

4. Ambition level

- Summarize the requirement in 5 to 20 words
- Like
  - "radically improve usability"
  - "Make the system totally impenetrable"
- Quote from management slides!

5. Ambiguity Test

- xx

- Let us pick a key term in your requirement, and see how clear it is, for the project:
- Everybody will write down their best interpretation
- And we will share the interpretations

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

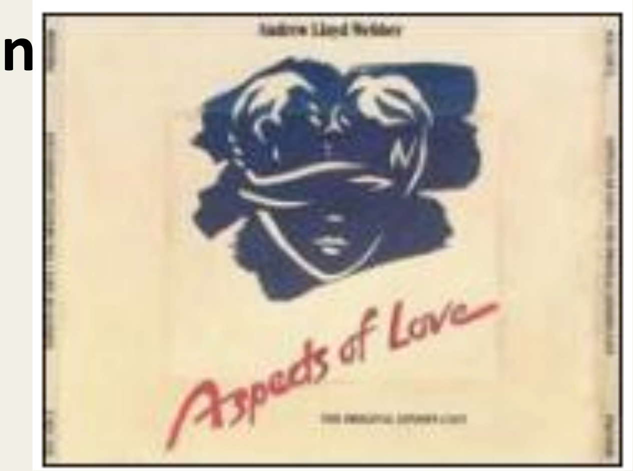
- Make a list of of love’s many aspects
- Quantify a requirement for one of those aspects

See note for Sutra

Love Attributes:
Brainstormed By Dutch Engineers

- **Kissed-ness**
- **Care**
- **Sharing**
- **Respect**
- **Comfort**
- **Friendship**
- **Sex**
- **Understandin**
- **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:**
  - **Broken Agreements**
  - **Late Appointments**
  - **Late delivery**
  - **Gossiping to Others**

“Camaraderie” quantified (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 : Bishop L Day, Boeing

The biblical citation (Book of First Corinthians I) **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)

11. rejoiceth in the truth

12. Beareth all things

13. believeth all things

14. hopeth all things

15. endureth all things

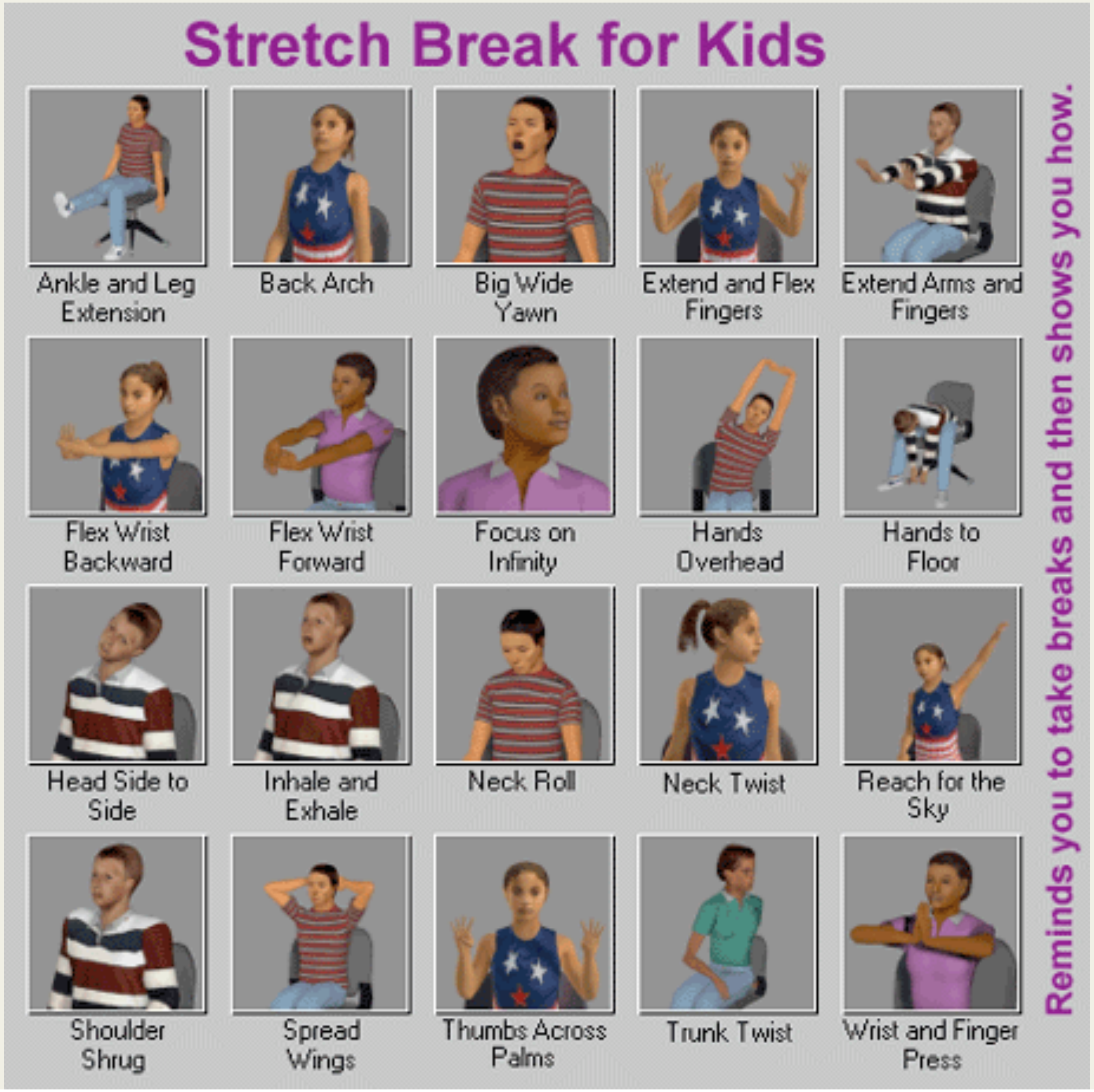
16. never faileth

6. Stakeholders Viewpoint: *they* have quality requirements

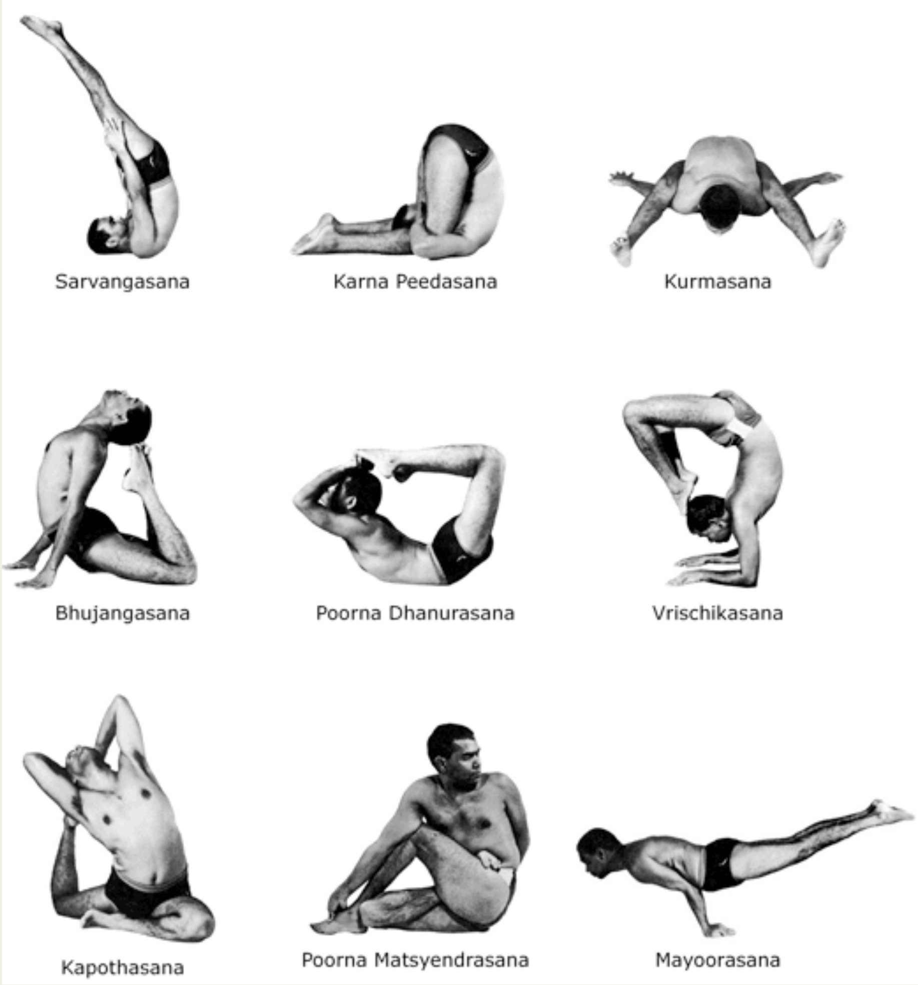
- Make a list of about 20 project/product stakeholders
  - 10 internal stakeholders
    - developers, support, sales
  - 10 external stakeholders
    - Users, buyers, reviewers, regulators

GOVERNMENT
MULTI STAKEHOLDER INITIATIVES
INVESTORS & STAKEHOLDERS
NGOS
CUSTOMERS
& OTHER BUSINESS PARTNERS
SUPPLIERS
EMPLOYEES
COMMUNITIES
CONSUMERS

Break


Stretch Break for Kids
Ankle and Leg Extension
Back Arch
Big Wide Yawn
Extend and Flex Fingers
Extend Arms and Fingers
Flex Wrist Backward
Flex Wrist Forward
Focus on Infinity
Hands Overhead
Hands to Floor
Head Side to Side
Inhale and Exhale
Neck Roll
Neck Twist
Reach for the Sky
Shoulder Shrug
Spread Wings
Thumbs Across Palms
Trunk Twist
Wrist and Finger Press
Reminds you to take breaks and then shows you how.

1600-1700 Advanced Workshop Exercises



Sarvangasana
Karna Peedasana
Kurmasana
Bhujangasana
Poorna Dhanurasana
Vrischikasana
Kapothasana
Poorna Matsyendrasana
Mayoorasana

7. Scale definition – 30 minutes?

- For 'qualities' (= how good the system will become)
- Define a scale of measure.
- If necessary: break down into sub-levels
- (like love aspects)

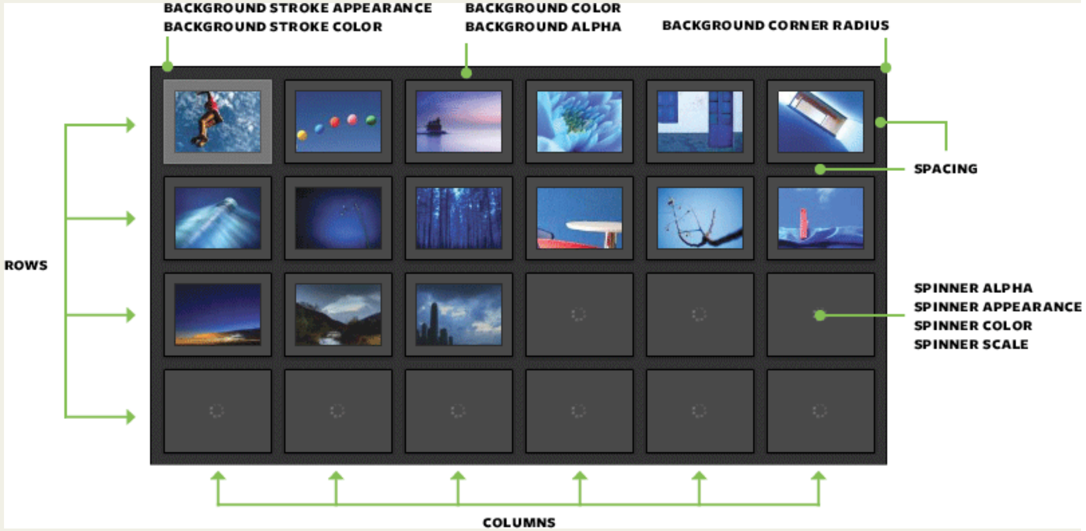
- Scale: ?

8. Meter definition

- Sketch, in a few words, some useful and reasonable way to test *where you are* along each scale of measure
  - **Scale**: Km/hour
  - **Meter**: Speedometer

9. Scale parameters: learning to generalize for later specificity

- **Rewrite** at least one scale to include 2 or more [general parameters]
- **Scale**: seconds from start to end for doing the job correctly.
- **Scale**: [Time] from [Starting Point] to [End Point] for doing a [Job] to a [Proficiency].

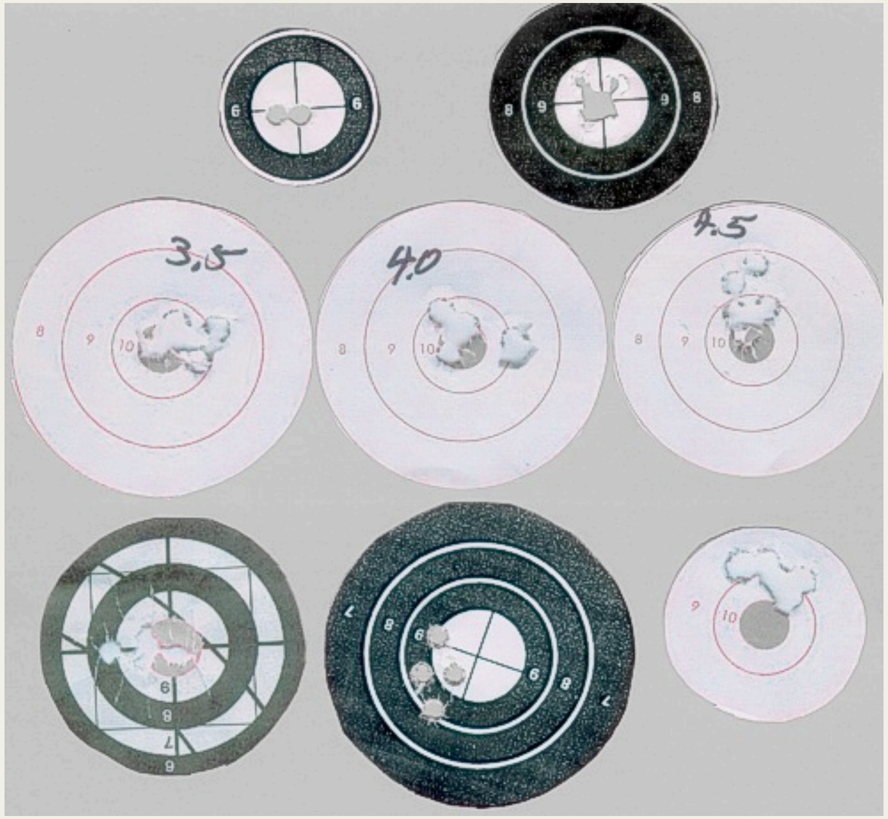


10. Scalar level expressions: (Qualifiers, level, uncertainty, source, justification)

- **Scale**: [Time] from [Starting Point] to [End Point] for doing a [Job] to a [Proficiency].
- **Goal**: [Time = Seconds, Starting Point= Input Return, End Point = Message Understood, Job= , Proficiency =]. 60 ±15?? <- TG Guess
  - Justification: Competitiveness

11. Quality Benchmarks

- Analytical Data, the departure point for improvement
- **Past**: 100 seconds
- **Trend** [Next Release] 120 seconds



12. Quality Constraints

- Restrictive borders
  - Keep away, dangers

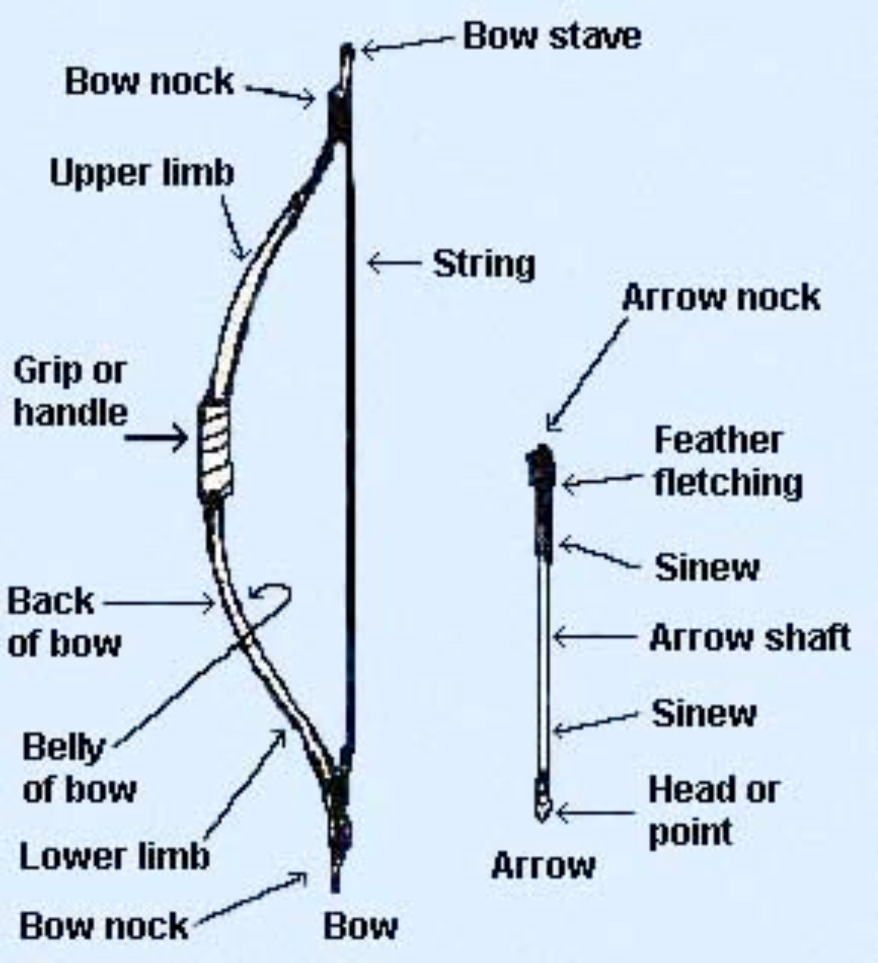
- Fail [1st release] ?
- Catastrophe [Long Term] ?

13. Quality Targets

- Where we want to be
  - Where there is value
  - Levels that may pay off
- Goal
- Stretch

14. more if time ??

– Impacted By <design>



DETAILED BACKUP SLIDES

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 their basis. Only through design work can you know about the available technology and its costs.

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

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

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

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

EXIT: (EXIT.QQ)

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

2. or exit condition “1.” above is waived

with the intent of getting experience or opinions

so as to refine it

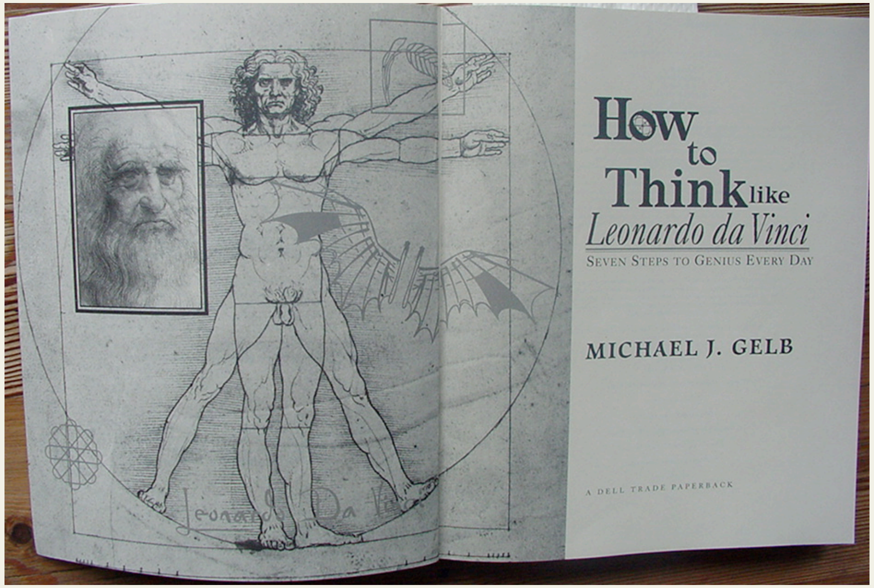
for official exit and more-serious use.

Specific Rules for Quality Quantification (QQ)

- 4.3. Rules: Quality Quantification. (RULES.QQ)
- The following rules would be
  - appropriate for a culture which was intent on raising quality specifications to a high level
  - and to systematically learn as a group,
  - in the long term,
  - from the experiences of themselves and others.
- The rules are guidance to the any writer or maintainer of quality specifications.
- Violations of these rules would be classed as ***'defects'*** in a quality control process on the document.

Da Vinci on Rules

- **“these rules will enable you to have a free and sound judgment:**
- **since good judgment is born of clear understanding,**
- **and a clear understanding comes of reasons derived from sound rules,**
- **and sound rules are the issue of sound experience –**
- **the common mother of all sciences and arts.”**
- ***The Notebooks of Leonardo da Vinci. 18.***


How to Think like Leonardo da Vinci
Seven Steps to Genius Every Day
MICHAEL J. GELB
A DELL TRADE PAPERBACK

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

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

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

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

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

4: Meter : a practical and economic Meter or set of 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

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.

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 or Ambition statement.
  6:QUALIFY: When any statement depends on a specific time, place or event being in force then this shall be specified by means of the [qualifier square brackets].
  7:FUZZY: When any element of a statement is unclear then it shall be marked, for later clarification, by the <fuzzy angle brackets>.
  8: COMMENT: any text which is secondary to a specification, and where no defect could result in a costly problem later, shall be written in *italic text statements, or/ and headed by suitable warning (NOTE, RATIONALE, COMMENT) or moved to footnotes*. Non-commentary specification shall be in plain text *Italic* can be used for emphasis of single terms in non-commentary statements. Readers shall be able to *visually* distinguish critical from not critical specification.
  9: UNIQUE: requirements and design specifications shall be made one single time only. Then they shall be re-used by cross reference to their identity tag. Duplication is strongly discouraged.

In addition to the general rules,
we can specify some special rules
for the specific types of statement
we are dealing with.

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

REQUIREMENTS SPECIFICATION RULES. SPECIFIC RULES.**SR**

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