Making Metrics Practical in the Development Process - ten fundamental principles for failure, and ten critical software metrics principles for success in the commercial environment.

- By Tom Gilb

- MASTER 2016

Ten fundamental software metrics principles,

- The drunk knew he'd lost his watch down the street in a dark corner,
  - –But it was tempting to look for it under the lamp post
- Determine what is most critical to control,
  - –and then find a way to quantify it - there is always a useful way
  - – then find ways to measure that quantity
    - There are always useful ways
- If you can't imagine the ways to quantify or measure something, the internet can.

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.

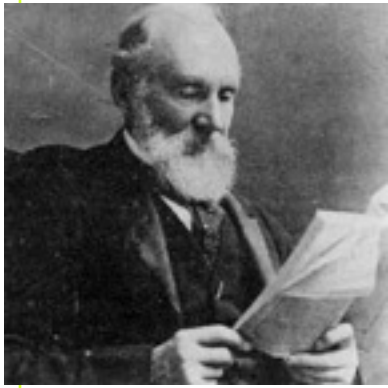
I often say that when you can measure what you are speaking about, and express it in numbers, you know something about it;

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."

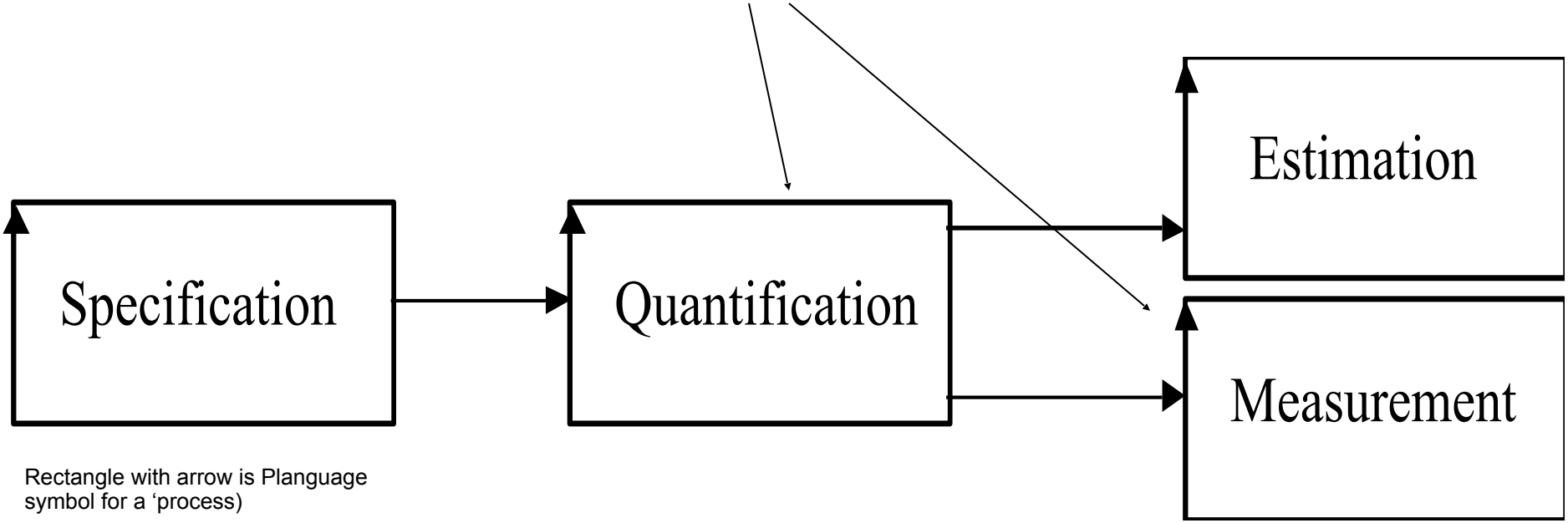
Lord Kelvin, 1893

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



Don't Confuse these Metrics Process Concepts

(Kelvin mentioned them 2x in one sentence!)



Rectangle with arrow is Planguage symbol for a 'process)

- Quantification is useful,
- even without measurement,
- for example!

Scales: **Units of Measure**
(NOT 'measuring method')

Scale ***-/-/-*** ***Concept *132***

A 'Scale' *parameter* is used to define a 'scale of measure'.

All elementary scalar attribute definitions require a defined Scale.

A Scale states the fundamental and precise *operational definition* for a specific scalar attribute.

It is used as the *basis* for expressing many of the *parameters* within the scalar attribute definition (for example, Meter, Goal and Budget):

all scalar estimates or measurements are made with reference to the Scale.
The Scale states the units of measurement, and any required scalar qualifiers.

User Friendly:

Type: Quality Requirement.

Ambition: To consistently exceed Competitor's ease of learning.

Scale: Time to Master
a defined [Task]
by defined [Learner].

Meter: <Use good academic practice, do at least 10 Tasks, with at least 5 Learner Types and at least 50 people>.

Record [Competitor AA, Product XYZ, Task = Dial Out, Learner = Novice]: 2 minutes <- Our current tests.

Goal [Our Company, Product ABC, Task = Dial Out, Learner = Novice]: < 10 seconds <- Marketing Requirement 4.5.7.

Master: Defined as: ability to pass a suitable approved test.

*Meter -|?|- Concept *093*

A Meter parameter is used to
- identify, or specify,
- the definition of a practical measuring device, process, or test
- that has been selected for use in measuring a numeric value (level) on a defined Scale.

> *"... there is nothing more important for the transaction of business than use of operational definitions."*
>
> *W. Edwards Deming, 1986 (Out of the Crisis, MIT Press)*

Repair:
Ambition: Improve the speed of repair of faults substantially, under given conditions.

Scale: Hours to repair or replace, from fault occurrence to when customer can use faultlessly, where they intended.

Meter [Product Acceptance]: A formal test in field with at least 20 representative cases,
[Field Audit]: Unannounced field testing at random.

========== Benchmarks ==========
Past [Product = Phone XYZ, Home Market, Qualified Dealer Shop]: {0.1 hours at Qualified Dealer Shop +
0.9 hours for the Customer to transit to/from Qualified Dealer Shop}.

Record [Competitor Product XX]: 0.5 hours average.
"Because they drive a spare to the customer office."
Trend [USA Market, Large Corporate Users]: 0.3 hours. "As on-site spares for large customers."

========== Targets ==========
Goal [Next New Product Release, Urban Areas, Personal Users]: 0.8 hours in total,
[Next New Product Release, USA Market, Large Corporate Users]: 0.2 hours
<- Marketing Requirement, 3 February This Year.

========== Constraints ==========
Fail [Next New Product Release, Large Corporate Users]: 0.5 hours or less on average
<- Marketing Requirement, 3 February This Year.

2. *If you measure too late, you deserve your fate.*

- you need to measure early, in order to discover
  - –what to measure, what the requirements really are
  - –what measures are useful
  - – what is worth measuring
  - – what numeric levels of requirements *should* be
- Measuring at the end of a project,
  - –Is just too late to learn in time
  - –To convince people they have a solvable problem in time to solve it

Real client example: weekly design impact estimates, and same week measurement,
Weekly Feedback to the development team
about cumulative progress toward critical numeric performance and quality targets

	A	B	C	D	E	F	G	BX	BY	BZ	CA
1											
2		Current Status	Improvements		Goals			Step9 Recoding			
								Estimated impact		Actual impact	
		Units	Units	%	Past	Tolerable	Goal		%	Units	
					Usability.Replacability (feature count)						
		1,00	1,0	50,0	2	1	0				
					Usability.Speed.NewFeaturesImpact (%)						
		5,00	5,0	100,0	0	15	5				
		10,00	10,0	200,0	0	15	5				
		0,00	0,0	0,0	0	30	10				
12					Usability.Intuitiveness (%)						
		0,00	0,0	0,0	0	60	80				
14					Usability.Productivity (minutes)						
		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

Qualities

Priority

Next week

Warning metrics based

Cost

Cumulative weekly progress metric

Benchmark

Constraint

Target

Estimates

Weekly Testing

Confirmit Case Slides http://www.gilb.com/community/tiki-download_file.php?fileId=33

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

Confirmit EVO week
WEEKLY METRICS CONTINUOUSLY, PRIMARY DRIVER

	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 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 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	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

EVO's impact on Confirmit product qualities

IMPRESSIVE QUARTERLY IMPROVEMENT METRICS for Users

Only 5 OF 25 REQUIREMENTS, highlights of the results, 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

3. If you measure too few, then ones you left out, will have all the clout. If you measure too many you will also lose out.

- Limit yourself, at any one level of consideration, to the maximum 'top ten' most critical requirement measures
  - when you have mastered all of them, you might have resources left to turn to the *next* priority requirement.
  - You cannot afford to distract your attention from the top few highest priorities
  - Mastering 10 critical variables, at demanding levels, is a magnificent technical management deed
    - You will be forgiven for failing on the 11th, *for the moment* - it is next on your hit list anyway.

The 25 Critical Improvement Requirements: Progress Report
4 product areas were attacked concurrently, by 4 small teams (3-4 people)

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.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		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		
3,0	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		84

Current Status	Improvements		Reportal - MR Features		
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

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

Code quality – ”green” week
Metrics for ‘Refactoring’, each month

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	100		
Maintainability.Doc.Code									
	100,0	100,0	0	80	100				
InterviewerConsole									
NUnitTests									
	0,0	0,0	0	90	100				
PeerTests									
	100,0	100,0	0	90	100			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	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									

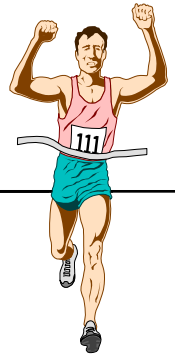
Speed
Maintainability.Doc.Code
NUnit Tests
Peer Tests
FX Cop
Test Director Tests
Robustness.Correctness
Robustness.Boundary Conditions
Speed
Resource Usage.CPU
Maintainability.Doc.Code
N Unit Tests

4. If the metric level is too low, you are in for a sorry blow.

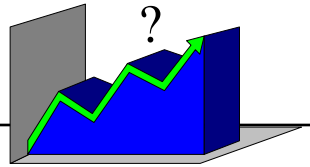
- What is ‘too low’ a requirement level?
- There are several simultaneous variations to consider:
  - too low in relation to a future competitor level (uncompetitive)
  - too low in relation to our current levels (worse product or service)
  - too low in relation to constraints
  - too low at a particular time
  - too low in a particular area
  - too low under specific conditions or events

Some Planguage 'Quantification' Level Concepts

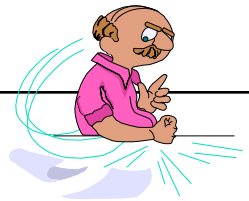
PAST: any useful reference point. Your old product, a competitors organization, a quality achieved in same discipline but different branch of business.



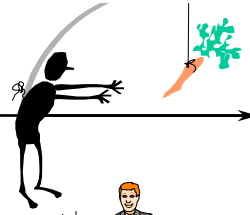
RECORD: best in some class, state of the art. Something to beat. A challenge for you. An extreme PAST.



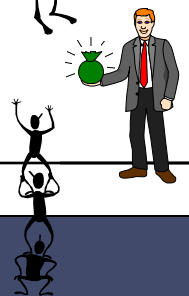
TREND: a future ? guess based on the PAST.



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



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



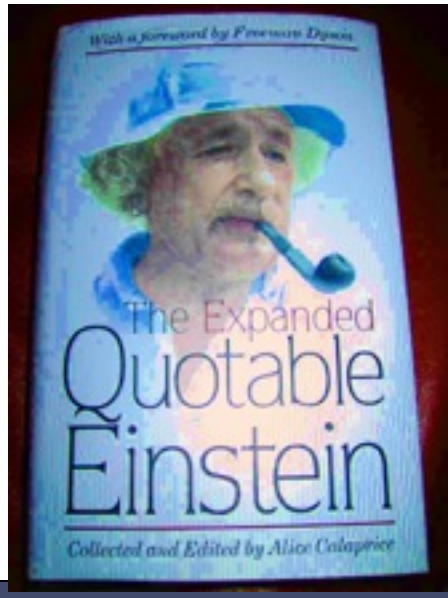
Wish: a level desired by someone, but which might not be feasible. Project is not committed to it.

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)

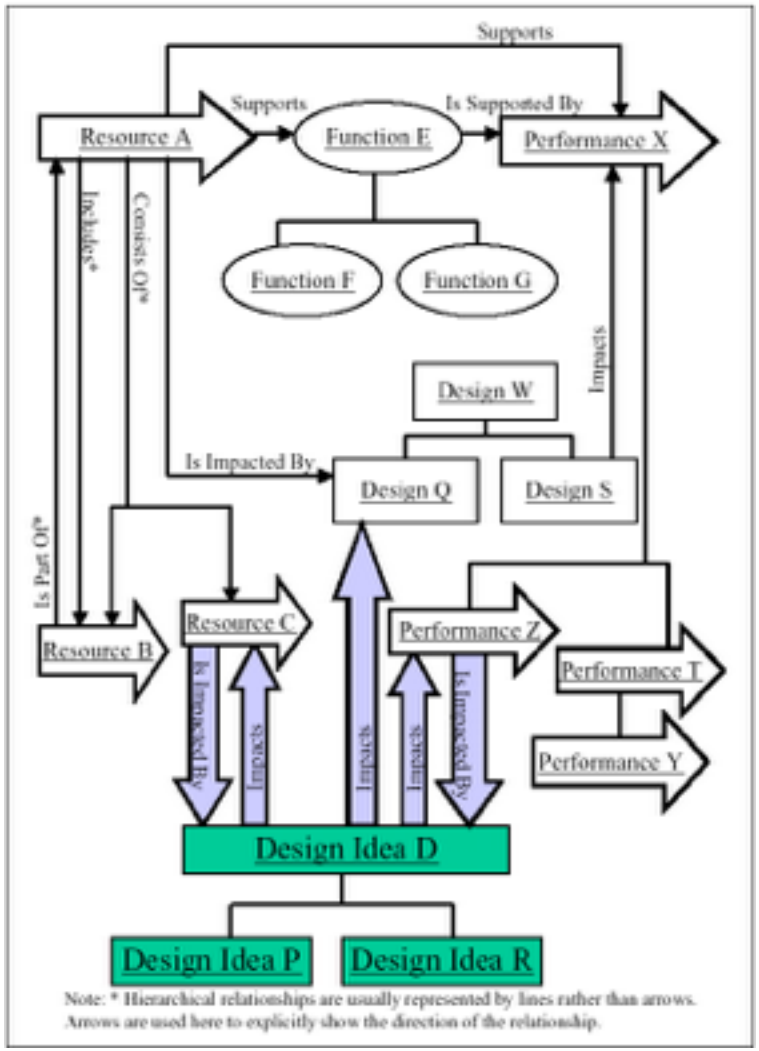


Source detail in notes section of this slide. (Calaprice, 2000)

- A metric lives in a system environment
  - Spaces
    - Geographical, Market Segment, Task Type, …..
  - Time
    - Deadlines
    - Intervals (‘office hours’, ‘weekends’)
    - Obsolete times, irrelevant times, …..
  - Concurrent events and conditions
    - Contracts signed, laws in force, achievements succeeded, …..
  - We need to carefully define that environment

Some Planguage parameters which define relationships:

- Authority
- Source
- Owner
- Author
- Implementer
- Impacts
- Supports
- Supported By
- Version
- Derived From
- Sub-component of
- Sub-components {list}
- Dependencies
- Contract
- Test Case
- Scenario
- Model
- And more!
- And 'Qualifiers, like
  - Goal [UK, Teens, 2009] 35%



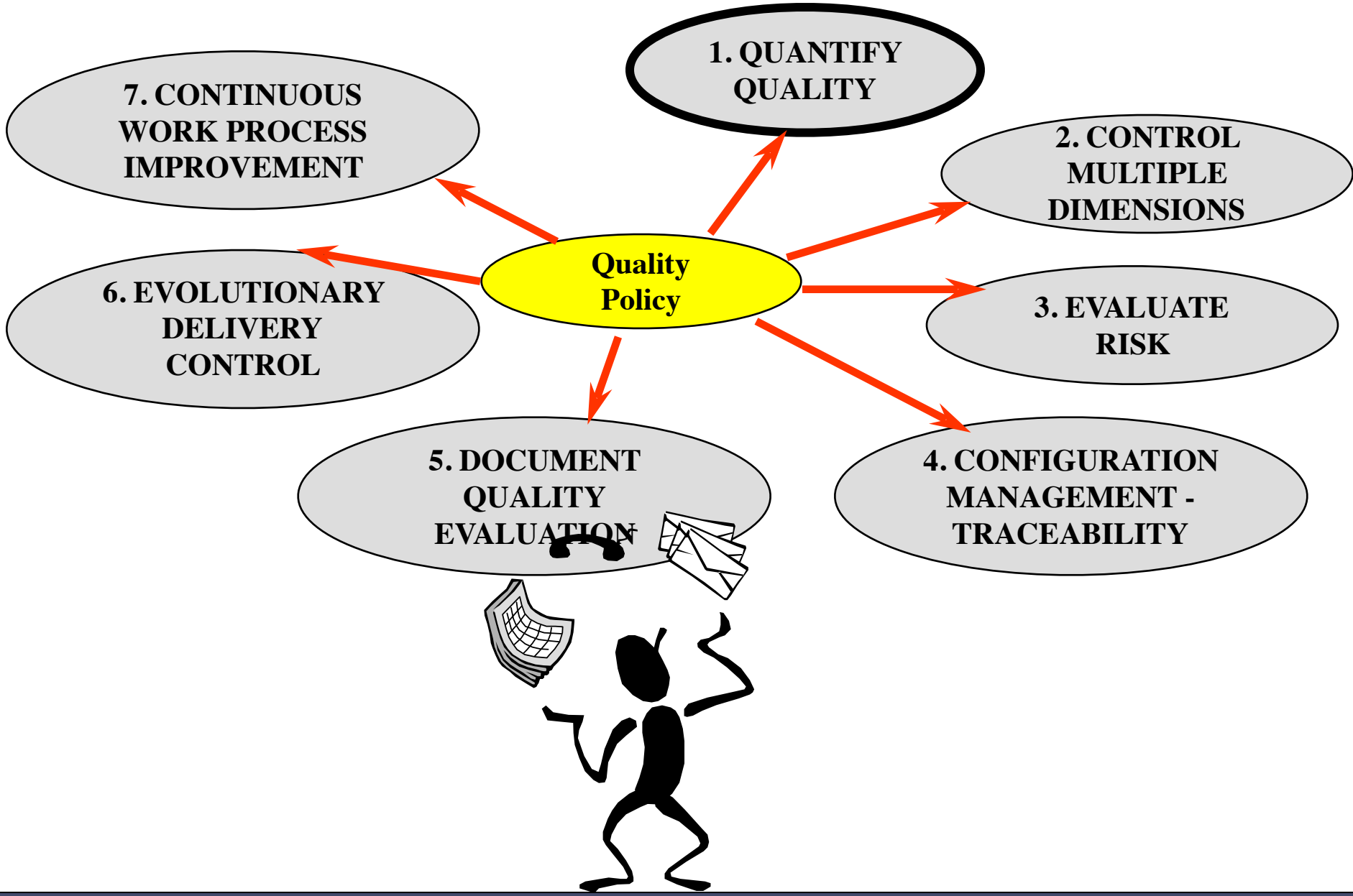
Several Metrics Specs, related to a single requirement

<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'**

- Developers will naturally prioritize quantified requirements that they believe they will be judged on delivering
  - –And quantified constraints (deadline, budget)

- So we need to have a notion of being 'complete' for the quantified critical requirements:
  - – we cannot have some quantified and others equally important in un-quantified formats like
    - "Very User-Friendly", "Highly Secure", "Extremely Adaptable"

A Corporate Quality Policy (Euro Multinational)

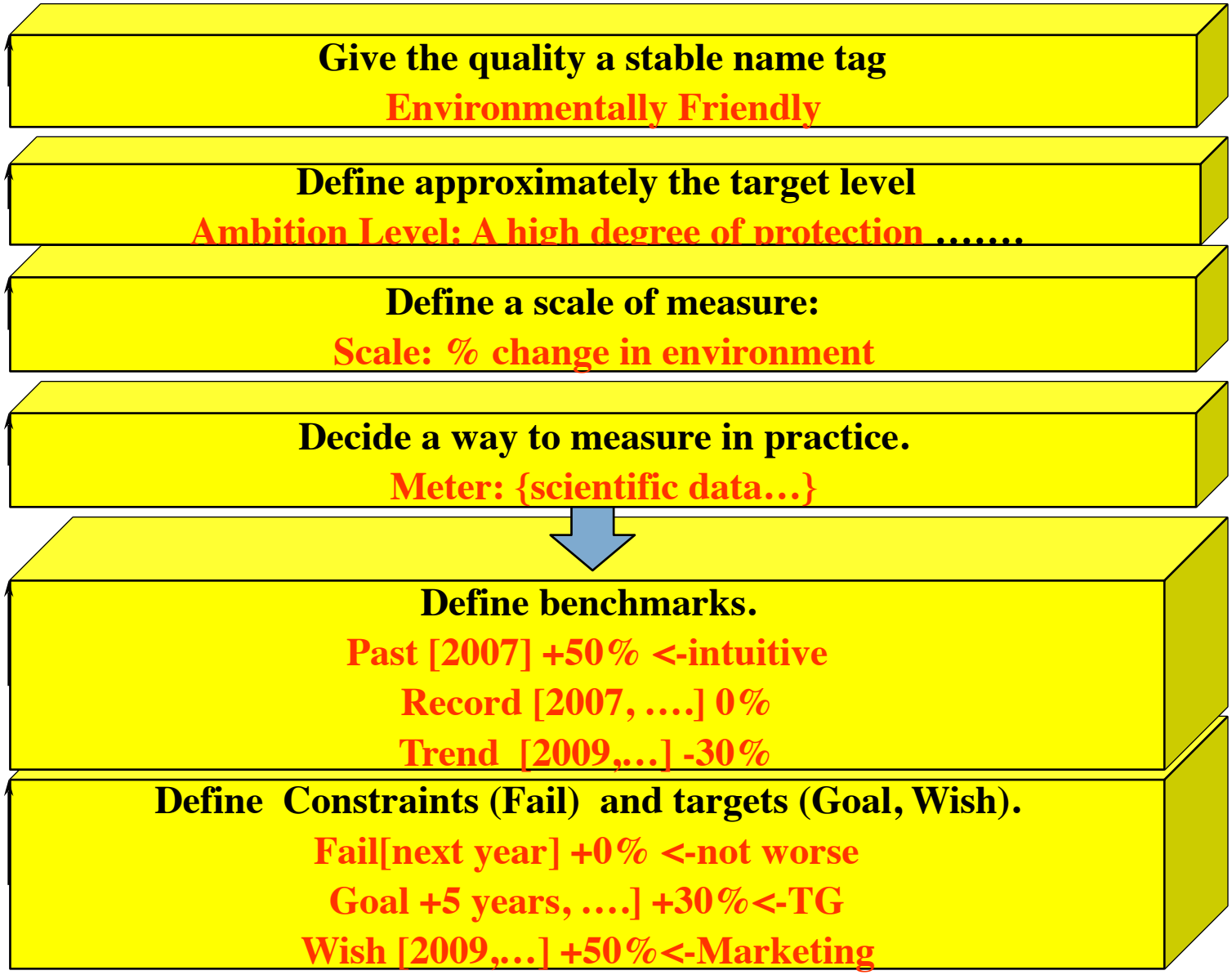


Real Corporate Policy on QUANTIFICATION, CLARIFICATION AND TESTABILITY OF CRITICAL OBJECTIVES:

"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.

‘Environmentally Friendly’ Quantification Example



- Managers have no training or culture in developing quantified and clear metrics for their most critical qualitative ('soft') objectives.
- they love to use a series of popular words, because that is their culture today
- if you guide them into quantifying their wordy objectives,
  - Some of them will love it and learn it
    - The CEO, COO, and CFO types
  - Some of them would rather lose their jobs
    - (the marketing types especially)

Real (NON-CONFIDENTIAL version) example of an initial draft of setting the objectives that engineering processes must meet.

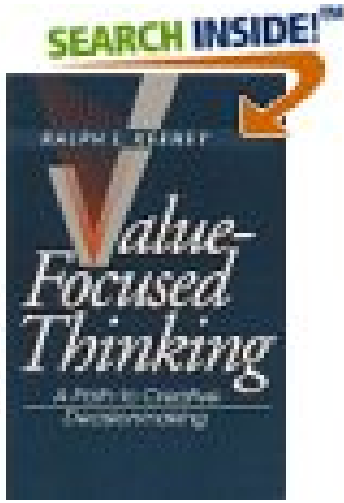
Business objective	Measure	Goal (200X)	Stretch goal ('0X)	Volume	Value	Profit	Cash
Time to market	Normal project time from GT to GT5	<9 mo.	<6 mo.	X		X	X
Mid-range	Min BoM for The Corp phone	<$90	<$30	X		X	X
Platformisation Technology	# of Technology 66 Lic. shipping > 3M/yr	4	6	X		X	X
Interface	Interface units	>11M	>13M	X		X	X
Operator prefere	[illegible] Corp	1	2	X		X	X
Productivity						X	X
Get Torden	Lyn goes for Technology 66 in Sep-04	Yes		X		X	X
Fragmenta n	[illegible]	<10%	<5%		X	X	X
Commoditisation	Switching cost for a UI to another System	>1yr	>2yrs		X	X	X
Duplication	The Corp share of 'in scope' code in best-[illegible]	>90%	>95%		X	X	X
Competitiven	[illegible] MX	Same	Better	X		X	X
User experience	Key use cases superior vs. competition	5	10	X	X	X	X
Downstream cost saving	Project ROI for Licensees	>33%	>66%	X	X	X	X
Platformisation IFace	Number of shipping Lic.	33	55	X		X	X
Japan	Share of of XXXX sales	>50%	>60%	X		X	X
	Numbers are intentionally changed from real ones						

Business Objectives Quantified

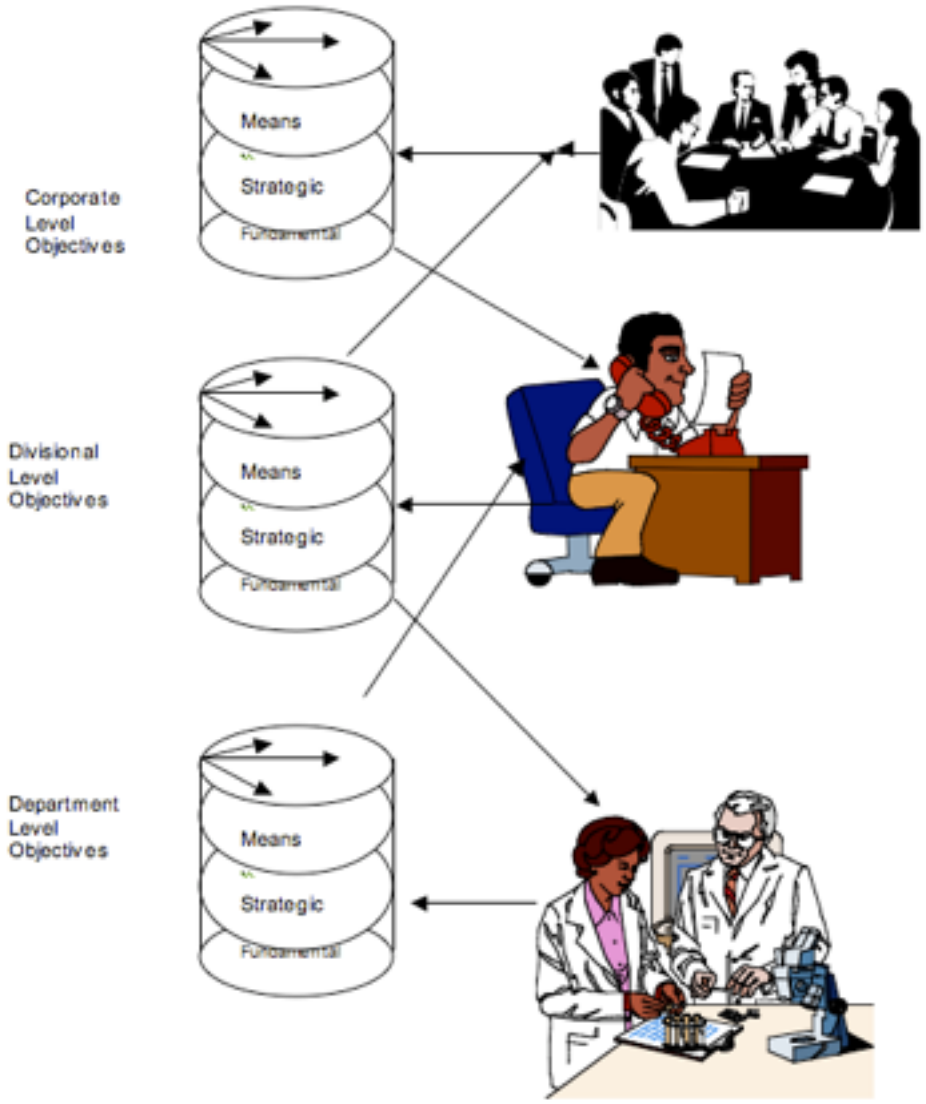
Impact on Top Business Objectives

8. *Some metrics support <u>other</u> metrics. You'd better know which is the star, and which is the supporting role.*

- Ralph Keeney's Levels ('Value-Focused Thinking')
  - Fundamental Objectives
  - Strategic Objectives
  - Means Objectives
- Are all relative to one's level in the organization
  - Fundamental Objectives (Your boss)
  - Strategic Objectives (you)
  - Means Objectives (your staff, and support)



Levels of Perception:
One level's Means objectives become the next level's fundamental objectives



- The varied top ten objectives metrics cannot be directly added to each other, to get a sum of improvements.
  - But the % of progress towards the 10 different Gola levels can be added and averaged to get some idea of progress to date

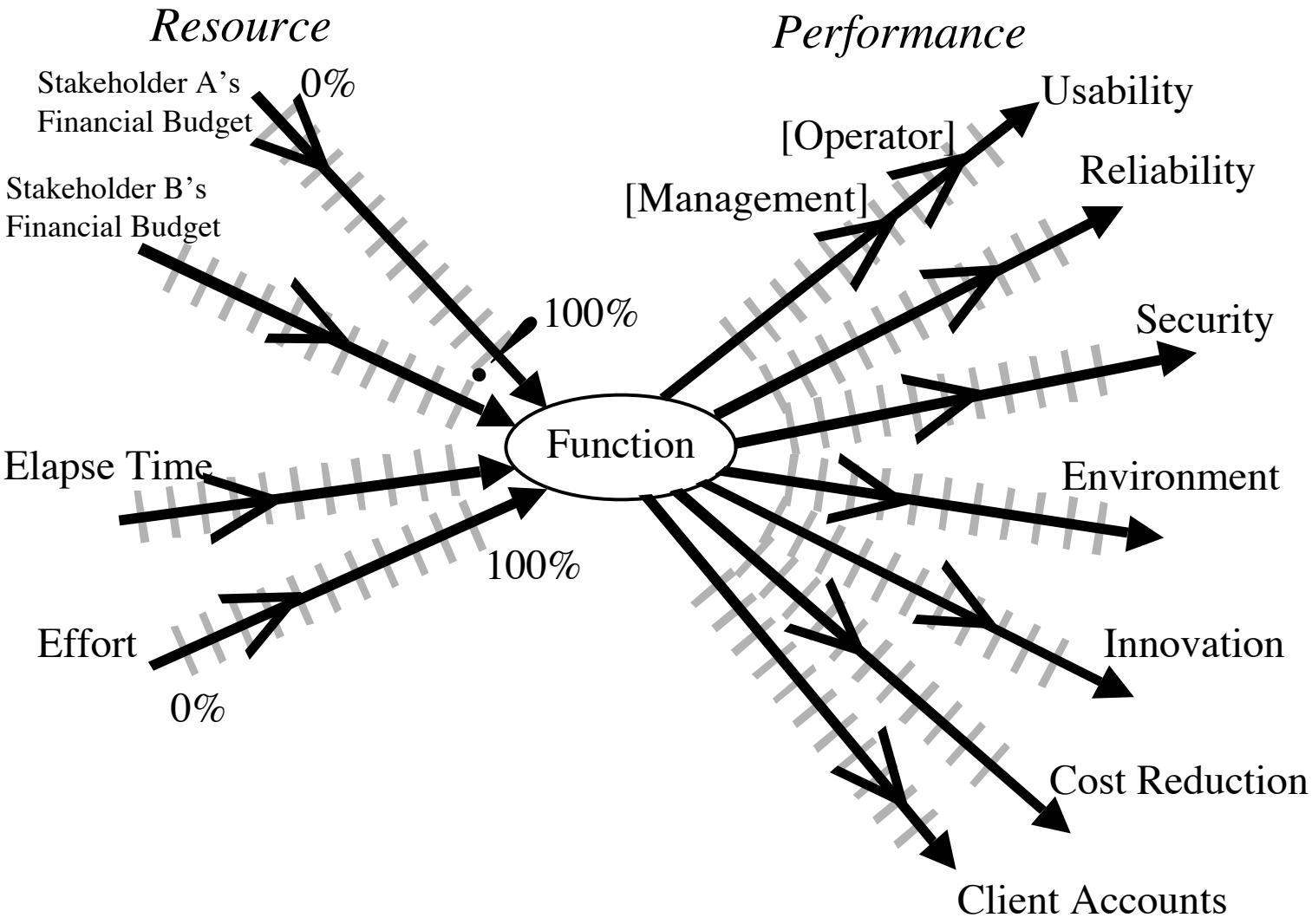
The 25 Critical Improvement Requirements: Progress Report
4 product areas were attacked concurrently, by 4 small teams (3-4 people)
Notice teams are > 75% complete after 9 of 12 weeks to deadline

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.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		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		
3,0	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		84

Current Status	Improvements		Reportal - MR Features		
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

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


Resource
Performance
Stakeholder A's Financial Budget
0%
Stakeholder B's Financial Budget
100%
Elapse Time
Effort
100%
0%
Function
[Operator]
[Management]
Usability
Reliability
Security
Environment
Innovation
Cost Reduction
Client Accounts

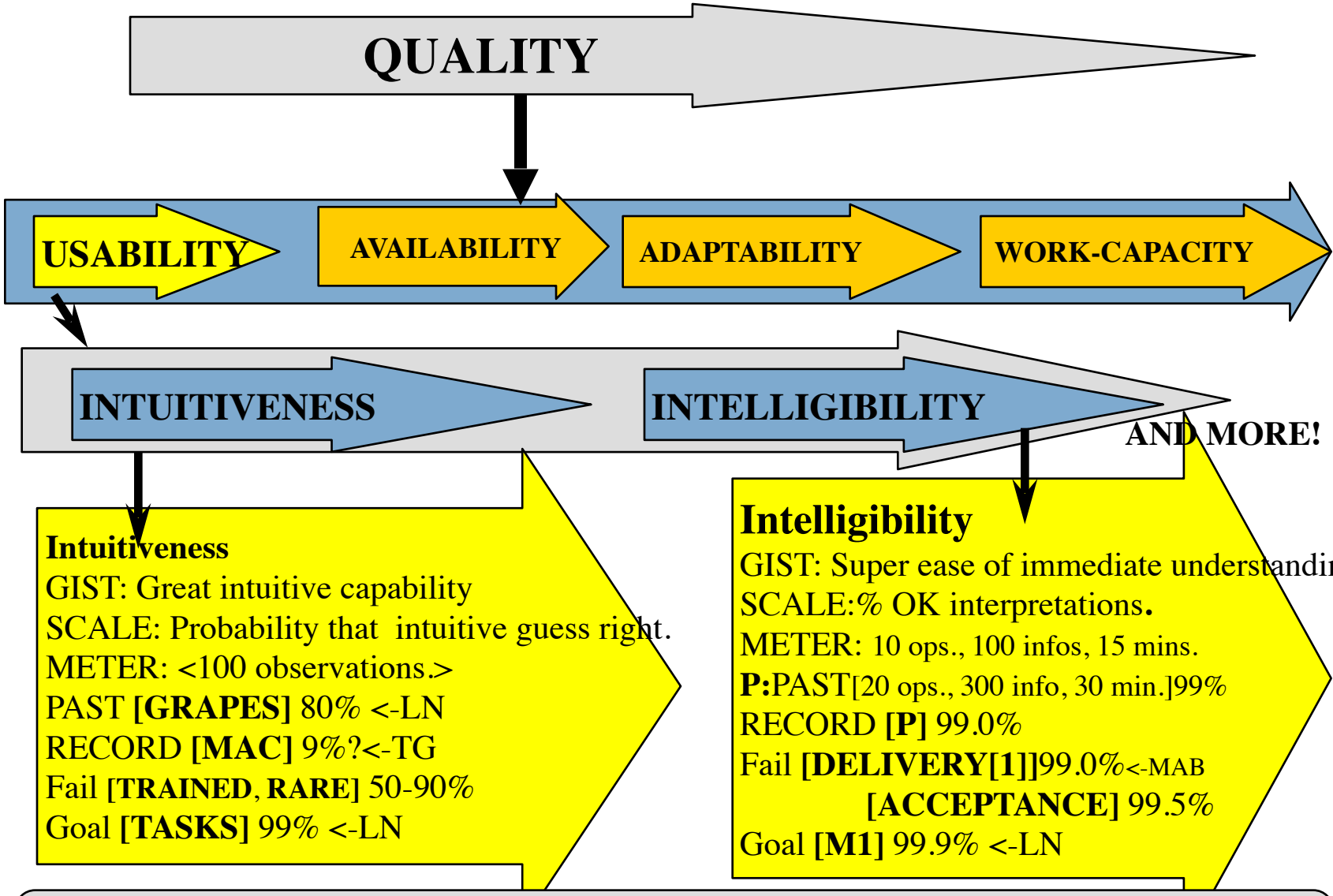
- So we need
  - sound best practice standards
  - training
  - management leadership
  - quality control
  - a constant learning process
- The ideas and practices exist
  - but the sound culture and motivation is not there

Ten critical software metrics usage principles for success in the commercial environment

1. *Develop requirements metrics top down from critical management objectives.*

- The most critical requirements in any project, are
  - The critical few improvements that the project sponsors are hoping for
  - They are 'always' quantifiable!
- All other 'requirements' are in reality supporting requirements for the top ones.
- At the top systems level there are some stakeholder values (quantifiable) - like save time.
  - Software products can have performance/quality requirements to directly support delivery of these values
    - Like: Increase Usability (defined by some Scale) by 50%, by next release

Quantifying Usability (Real C&C System 'Erieye')



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.

2. Connect metrics with metrics.

- there are many types and levels of metrics
- And you should make their relationships and connections clear and documented

System benchmarks are integrated with future requirements

Adaptability:
Type: Quality Requirement.
Scale: The calendar time in hours needed to re-configure the defined [Base Configuration] to any other defined [Target Configuration] using defined [Methods] and defined [Reconfiguration Staff].

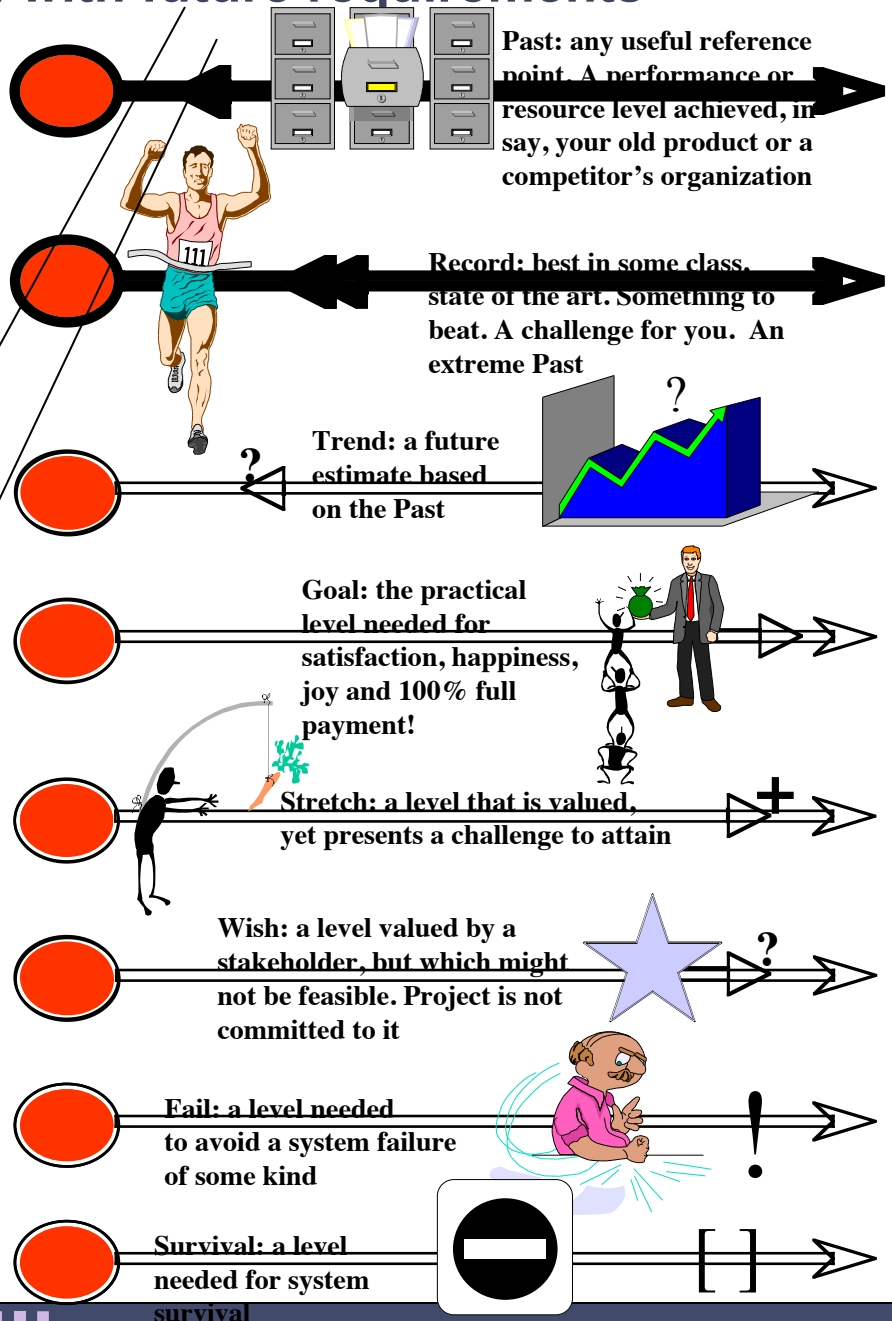
Expert Reconfiguration: Defined As:
{Base Configuration = Novice Setup,
Target Configuration = Expert Setup,
Methods = Selection of Library Reconfiguration Process,
Reconfiguration Staff = Qualified Expert}.

======== Benchmarks ================================
Past [Expert Reconfiguration, Version 0.3, Asian Market]: < 1 hour.

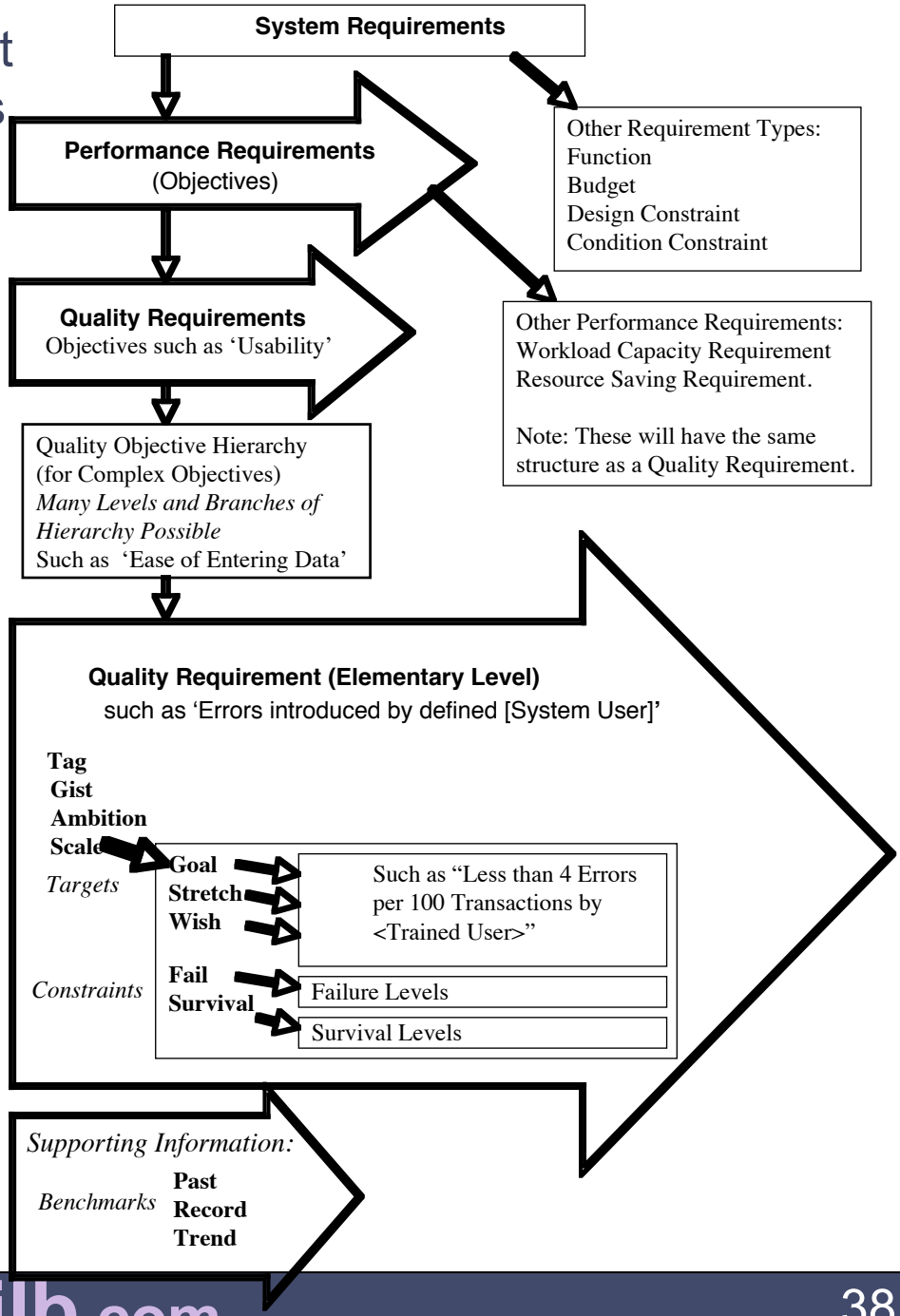
======== Goals (Performance Targets) ==================
Authority [Goals]: Federal Drug Administration.
Goal [Expert Reconfiguration, Deadline = Version 1.0]: < 0.5 hours.
Goal [Expert Reconfiguration, Deadline = Version 2.0]: < 0.1 hours.

======== Constraints ================================
Fail [All USA Products]: < 0.7 hours.
Fail [Expert Reconfiguration, Deadline = Version 2.0]: < 0.5 hours.

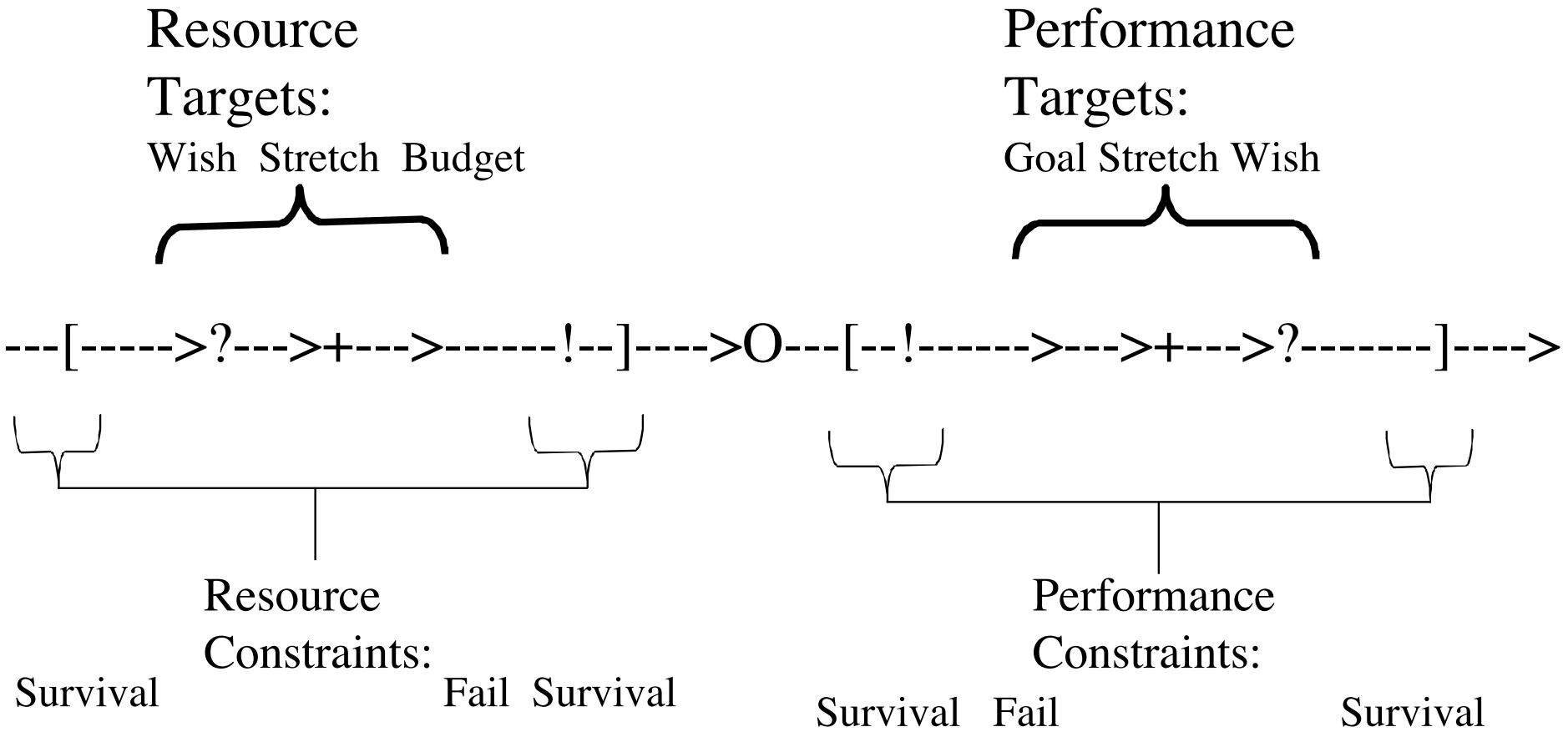
Survival [Expert Reconfiguration, European Market]: < 1 working day.



Examples of connecting requirements at different levels of perception, and levels of detail and specialization



Targets and Constraints
Different levels of Performance Requirements



Benchmarks: Past, Record & Trend

Past: A relevant benchmark level already achieved by an existing system (our own, competitive, or any other system) that is worth consideration.

Record: A 'Past', which is the best known result [in some defined area]. A 'state-of-the-art' value.

Trend: An extrapolation of past data, trends and emerging technology to a defined [time and place].

- Aside from our own project's plans to improve this level, what future levels are likely to be achieved by others?
- What will we be competing with?

Usability [New Product Line, Major Markets]:

Ambition: To achieve a low average time-to-learn to use our telephone answerer, under various conditions.

Scale: Average number of minutes for defined [representative user and all their household family members over 5 years old] to learn to use defined [basic daily use functions] correctly.

Meter [Product Acceptance]: A formal test in field with at least 20 representative cases,
[Field Audit]: Unannounced field testing at random.

========= Benchmarks ==========
Past [Product XYZ, Home Market, People between 30 and 40 years old, in homes in Urban Areas, <For one explanation & demo>]: 10 minutes.

Record [Competitor Product XX, Field Trials]: < 5 minutes?> <- one single case reported.

Trend [USA Market, S Corporation, By Initial Release]: 10 seconds <- Public Market Intelligence Report.

======== Constraints ==========
Fail [Next New Product Release, Children over 10]: 5 minutes <- Marketing Requirements 3 February Last Year.

========= Targets ==========
Goal [Next New Product Release, Urban Areas, Personal Users]: 5 minutes total,
[Next New Product Release, USA Market, Large Corporate Users]: 5 minutes <- Marketing Requirements 3 February Last Year.

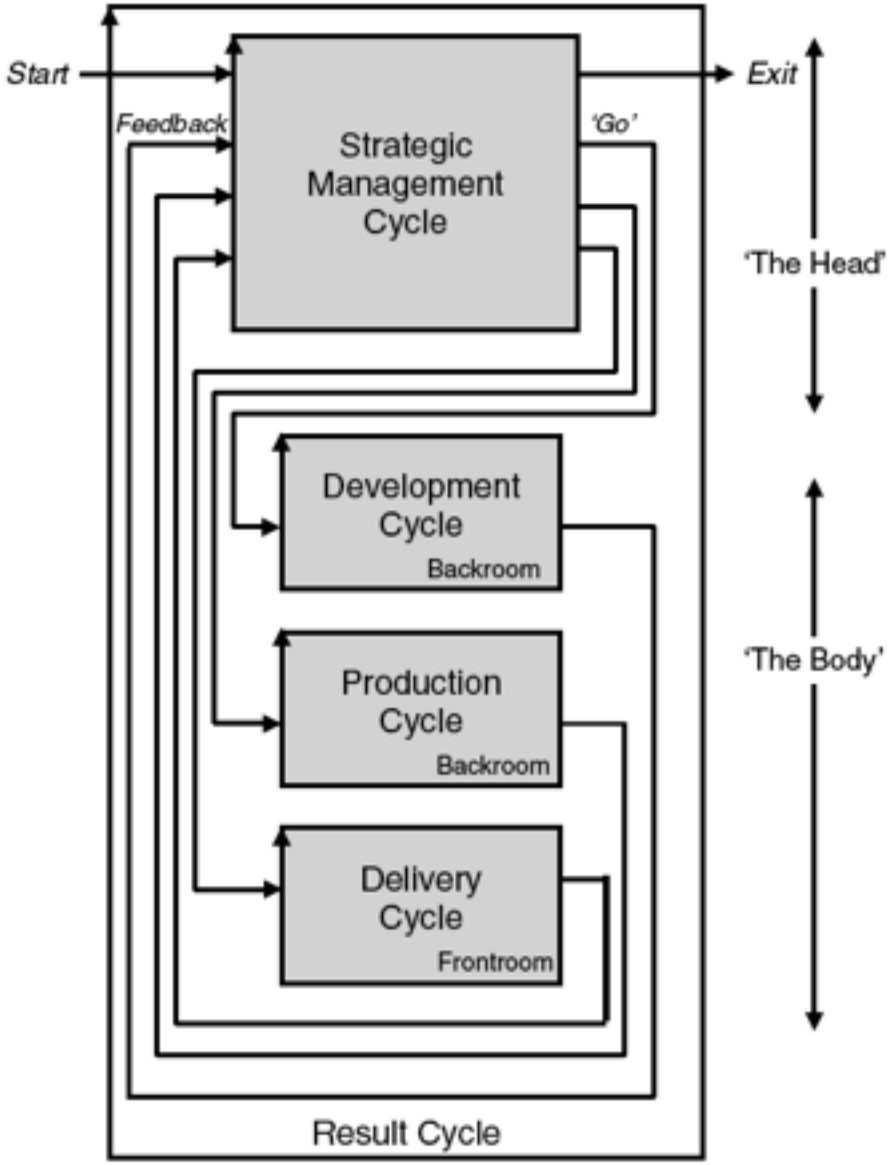
Stretch [Next Year]: (Record - 10%).

- You will be trying to get to a few numeric long term goal levels - of performance/quality.
- We believe the smartest way to the long term is to try to move towards them in early, frequent, small 'weekly' steps.
- The metrics are estimated, then measured, then evaluated against estimates, to learn.
  - this gets real results for stakeholders
  - This makes sure your entire development process works
  - this makes it impossible to fail big - just stop if you are failing in the small increments
    - The metrics will remind you that you do not know what you are doing!

The Result Cycle for an Evo Step

It is all about feedback and learning,

And real forward motion - proven by the metrics



Philips Evo Pilot May 2001

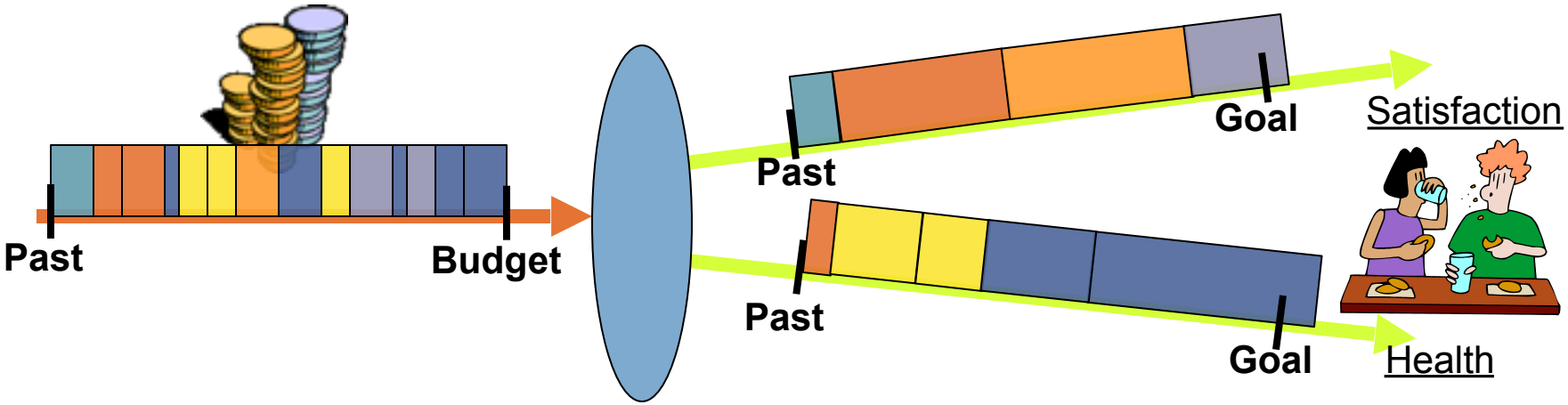
An example Of frequent Weekly Result Delivery steps

# Jobs	Week	[- 5%,+10%]	[-10%,+20%]	[-15%,+30%]	out of range
6	wk 8	1	5		
11	wk 9	3	1	7	
19	wk 10	6	3	7	3
25	wk 11	6	4	6	9
25	wk 12	17	3	5	
42	wk 13	31	3	2	6
55	wk 14	37	11	1	6
55	wk 15	39	9	1	6
55	wk 16	48	4	1	2
55	wk 17	50	4		1

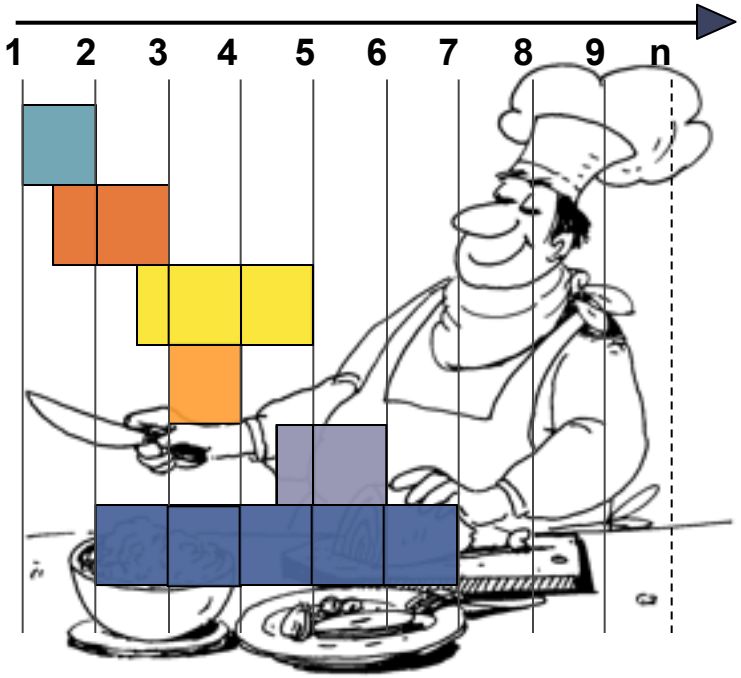
Frank van Latum, The Manager

The GxxLine PXX Optimizer EVO team proudly presents the success of the Timing Prediction Improvement EVO steps.
Shown are the results of the test set used to monitor the improvement process.
The size of the test set has grown, as can be seen in the first column. (In the second column the week number is shown.)
We measured the quality of the timing prediction in percentages, in which –5% means that the prediction by the optimizer is 5% too optimistic.
Excellent quality (–5% to +10%) is given the color green, very good quality quality is yellow, good quality is orange, & the rest is red.
The results are for the ToXXXz X(i) and EXXX X(i), and are accomplished by thorough analysis of the machines, and appropriate adaptation of the software.
The GXXline Optimiser Team presented the word document below to the Business Creation Process review team.
The results were received with great applause. The graphics are based on the timing accuracy scale of measure that was defined with Jan verbakel. Classification: Unclassified

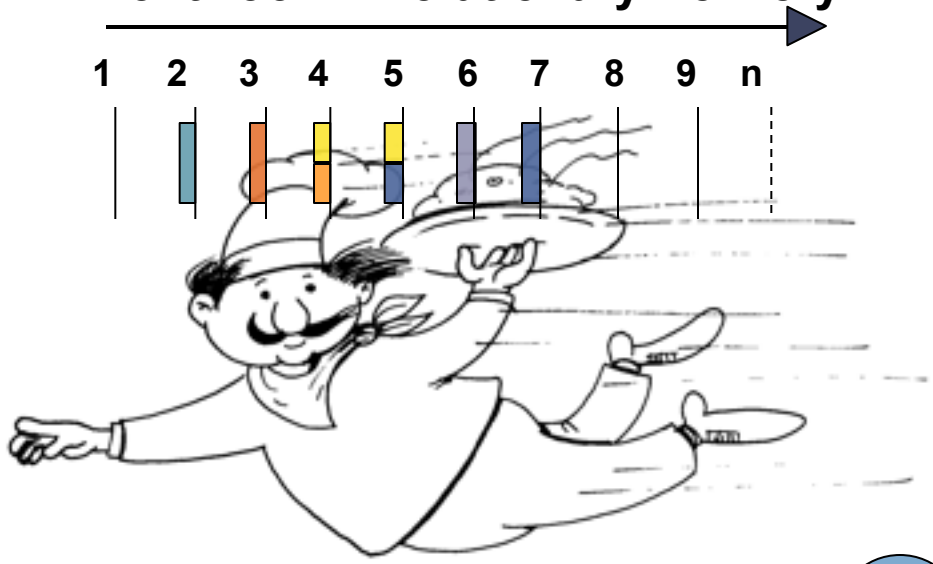
Costs / Effects in measurable increments



Back-room Design Development



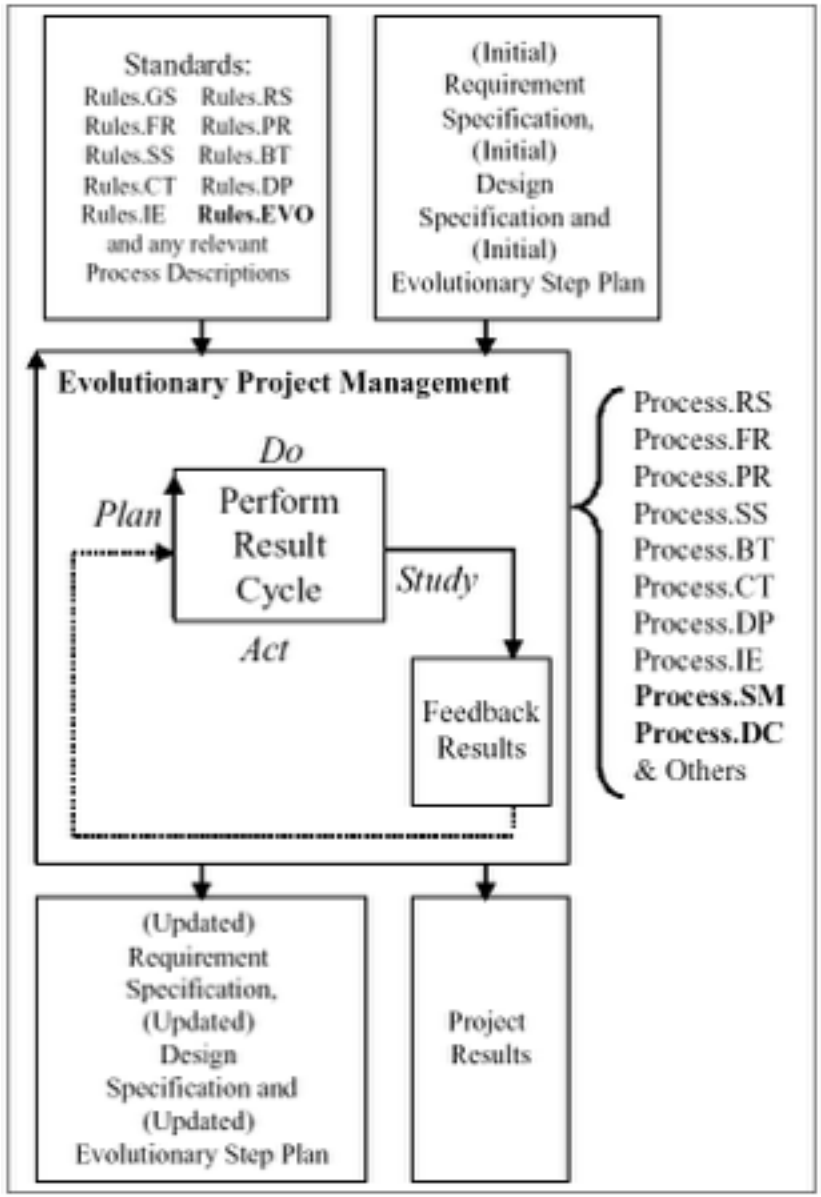
Front-room Evolutionary Delivery



How does Planguage Specification support Evo?

Quantified metrics requirements are the *project management*
- –result delivery targets and
- –Constraints

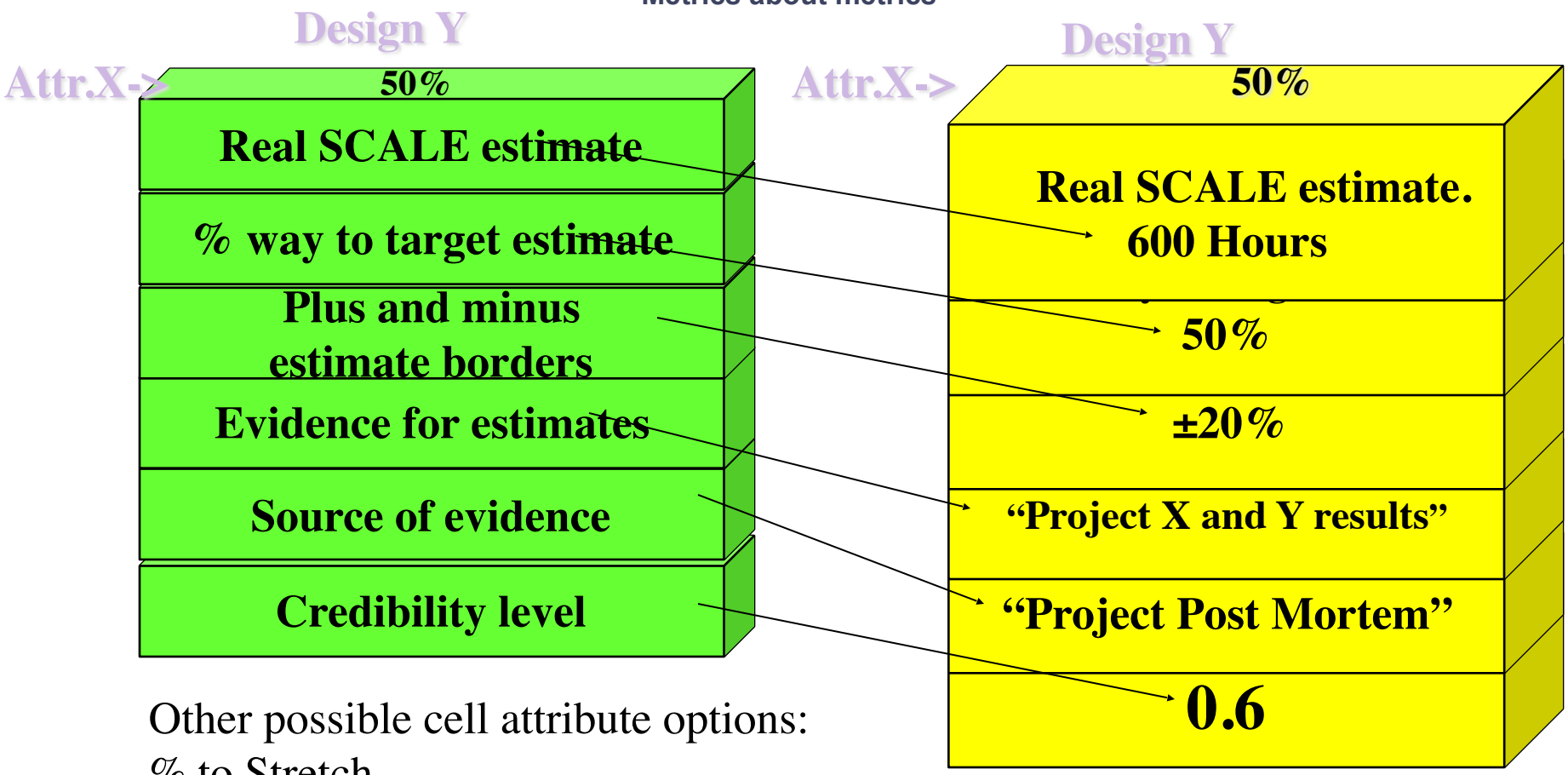
designs, and corresponding quantified *impact estimates* help *control*
- –the delivery and
- –implementation process



- a Metric has attributes,
  - their qualities -
    - like accuracy, credibility, relevance, impact
  - and costs
    - Like learning cost, test setup cost, test process costs, test analysis costs

- We can use metrics to describe and understand our primary metrics
  - And to better select both scales of measure, and corresponding measurement processes.

Impact Estimation: Cell Depth: Metrics about metrics



Other possible cell attribute options:
% to Stretch
% to Goal [other qualifier]
Owner of estimate. “Tom”
Version: 1.01
Date of Estimate: May 9, 2004

Impact Estimation Analyzes Requirement |-| Design relationships across systems if necessary.

	On-line Support	On-line Help	Picture Handbook	On-line Help + Access Index
Learning Past: 60minutes <-> Goal: 10minutes				
Scale Impact	5 min.	10 min.	30 min.	8 min.
Scale Uncertainty	±3min.	±5 min.	±10min.	±5 min.
Percentage Impact	110%	100%	60%	104%
Percentage Uncertainty	±6% (3 of 50 minutes)	±10%	±20%?	±10%
Evidence	Project Ajax: 7 minutes	Other Systems	Guess	Other Systems + Guess
Source	Ajax Report, p.6	World Report, p.17	John B	World Report, p.17 + John B
Credibility	0.7	0.8	0.2	0.6
Development Cost	120K	25K	10K	26K
Performance to Cost Ratio	110/120 = 0.92	100/25 = 4.0	60/10 = 6.0	104/26 = 4.0
Credibility-adjusted Performance to Cost Ratio (to 1 decimal place)	0.92*0.7 = 0.6	4.0*0.8 = 3.2	6.0*0.2 = 1.2	4.0*0.6 = 2.4
Notes: Time Period is two years.	Longer timescale to develop			

- Source Competitive Engineering Fig 9.5

Credibility (of Evidence and Source!) Rating Scale (CE p.274, fig. 93.)

Credibility Rating	Meaning
0.0	Wild guess, no credibility
0.1	We know it has been done somewhere
0.2	We have one measurement somewhere
0.3	There are several measurements in the estimated range
0.4	The measurements are relevant to our case
0.5	The method of measurement is considered reliable
0.6	We have used the method in-house
0.7	We have reliable measurements in-house
0.8	Reliable in-house measurements correlate to independent external measurements
0.9	We have used the idea on this project and measured it
1.0	Perfect credibility, we have rock solid, contract-guaranteed, long-term, credible experience with this idea on this project and, the results are unlikely to disappear

Evidence - by Thomas and John

- "The most formidable weapon against errors of every kind is reason."
- --Thomas Paine

- "Facts are stubborn things; and whatever may be our wishes, our inclinations, or the dictates of our passions, they cannot alter the state of *facts and evidence*."
- --John Adams

- all ‘designs’ have multiple performance/quality/cost attributes,
  - That define ‘how well’ the designs satisfy our requirements.
- ‘software’ as a craft is not yet at the engineering stage of maturity
  - Because then we would more systematically be matching up numeric design attributes , to numeric requirements.
  - today we match
    - ambiguous words (‘enterprise architecture’)
    - with other ambiguous words (‘IT system flexibility’)
  - (software witchcraft, not software engineering)

Design Specification Template <with Hints>

Tag: <Tag name for the design idea>.
Type: {Design Idea, Design Constraint}.
========== Basic Information ==========
Version: <Date or version number>.
Status: <{Draft, SQC Exited, Approved}>.
Quality Level: <Maximum remaining major defects/page, sample size, date>.
Owner: < Role/e-mail/name of person responsible for changes and updates>.
Expert: < Name and contact information for a technical expert, in our organization or otherwise available to us, on this design idea>.
Authority: <Name and contact information for the leading authorities, in our organization or elsewhere, on this technology or strategy. This can include references to papers, books and websites>.
Source: <Source references for the information in this specification. Could include people>.
Gist: <Brief description>.
Description: <Describe the design idea in sufficient detail to support the estimated impacts and costs given below>.
Stakeholders: <Prime stakeholders concerned with this design>.
========== Design Relationships ==========
Reuse of Other Design: <If a currently available component or design is specified, then give its tag or reference code here to indicate that a known component is being reused>.
Reuse of This Design: <If this design is used elsewhere in another system or used several times in this system, then capture the information here>.
Design Constraints: <If this design is a reflection of attempting to adhere to any known design constraints, then that should be noted here with reference one or more of the constraint tags or identities>.
Sub-Designs: <Name tags of any designs, which are subsets of this one, if any>.
========== Impacts Relationships ==========
Impacts [Functions]: <list of functions and subsystems which this design impacts attributes of>.
Impacts [Intended]: <Give a list of the performance requirements that this design idea will impact in a major way, good or bad. The positive impacts are the main justification for the existence of the design idea!>.
Impacts [Side Effects]: <Give a list of the performance requirements that this design idea will impact in a more minor way, good or bad>.
Impacts [Cost]: <Give a list of the budgets that this design idea will impact in a major way>.
Impacts [Other Designs]: <Does this design have any consequences with respect to other designs? Name them at least>.
Value: <Name or quantify value produced, and stakeholders affected by this design. Use Qualifiers>
========== Impact Estimation/Feedback ==========
For each Scalar Requirement in Impacts [Intended] (see above):
Tag: <Tag of a scalar requirement listed in Impacts [Intended]>.
Scale: <Scale for the scalar requirement>.
Scale Impact: <Give estimated or real impact, when implemented, using the defined Scale. That is, given current baseline numeric value, what numeric value will implementing this design idea achieve or what numeric value has been achieved?>.
Scale Uncertainty: <Give estimated optimistic/pessimistic or real ± error margins>.
Percentage Impact: <Convert Scale Impact to Percentage Impact. That is, what percentage of the way to the planned target, relative to the baseline and the planned target will implementing this design idea achieve. Has been achieved? 100% means meeting the defined Plan level on time>.
Percentage Uncertainty: <Convert Scale Uncertainty to Percentage Uncertainty. Plan deviations>.
Evidence: <Give the observed numeric value, dates, places and other relevant information where you have tapped previous experience using this design idea>.
Source: <Give the person or attribute source of evidence>
Credibility: <Credibility 0.0 low to 1.0 high. Rate the credibility of your estimates, based on the evidence and its source>.
========== Priority and Risk Management ==========
Assumptions: <Any assumptions that have been made>.
Dependencies: <State any dependencies for this design idea>.
Risks: <Name or refer to tags of any factors, which could threaten your estimated impacts>.
Priority: <List the tag names of any design ideas that must be implemented before or after this design idea>.
Issues: <Unresolved concerns or problems in the specification or the system>.
========== Implementation Control ==========
Supplier: < Name actual supplier or list supplier requirements>
Responsible: <Who in or organization is responsible for managing the supplier relation?>
Contract: <Refer to the contract if any, or the contract template>
Test Plan: <Refer to specific test pan for this design>
Implementation Process: <Name any special needs during implementation>
Location of Master Specification: <Give the intranet web location of this master specification>.

See next slide for

Metrics for a Design Spec
Enlargement of Full Design Spec Template

========== Impacts Relationships ==========

Impacts [Functions]: <list of functions and subsystems which this design impacts attributes of>.

Impacts [Intended]: <Give a list of the performance requirements that this design idea will impact in a major way, good or bad. The positive impacts are the main justification for the existence of the design idea!>.

Impacts [Side Effects]: <Give a list of the performance requirements that this design idea will impact in a more minor way, good or bad>.

Impacts [Cost]: <Give a list of the budgets that this design idea will impact in a major way>.

Impacts [Other Designs]: <Does this design have any consequences with respect to other designs? Name them at least>.

Value: <Name or quantify value produced, and stakeholders affected by this design. Use Qualifiers>

========== Impact Estimation/Feedback ==========

For each Scalar Requirement in Impacts [Intended] (see above):

Tag: <Tag of a scalar requirement listed in Impacts [Intended]>.

Scale: <Scale for the scalar requirement>.

Scale Impact: <Give estimated or real impact, when implemented, using the defined Scale. That is, given current baseline numeric value, what numeric value will implementing this design idea achieve or what numeric value has been achieved? >.

Scale Uncertainty: <Give estimated optimistic/pessimistic or real ± error margins>.

Percentage Impact: <Convert Scale Impact to Percentage Impact. That is, what percentage of the way to the planned target, relative to the baseline and the planned target will implementing this design idea achieve or, has been achieved? 100% means meeting the defined Plan level on time>.

Percentage Uncertainty: <Convert Scale Uncertainty to Percentage Uncertainty ± deviations>.

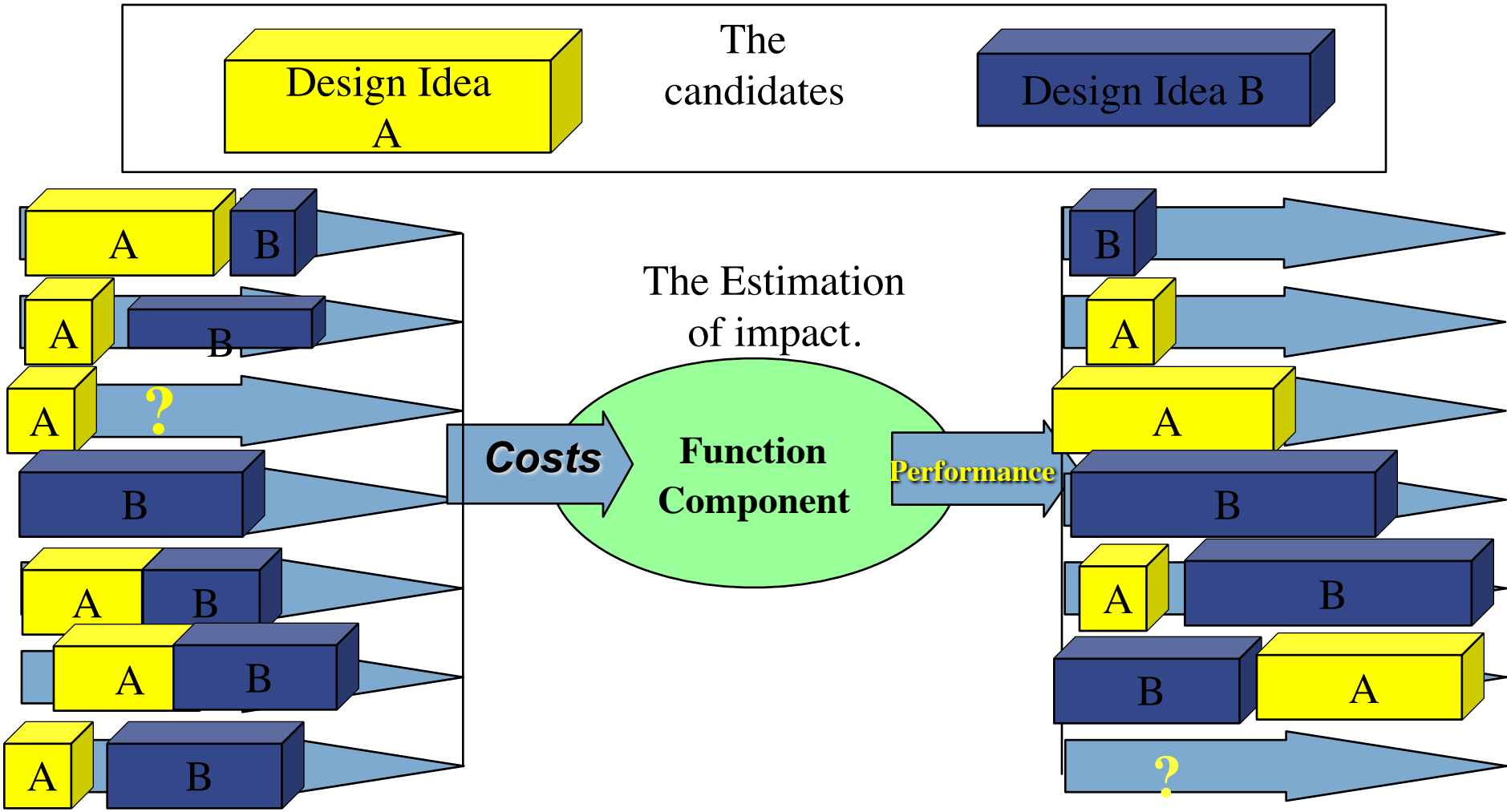
Evidence: <Give the observed numeric values, dates, places and other relevant information where you have data about previous experience of using this design idea>.

Source: <Give the person or written source of your evidence>.

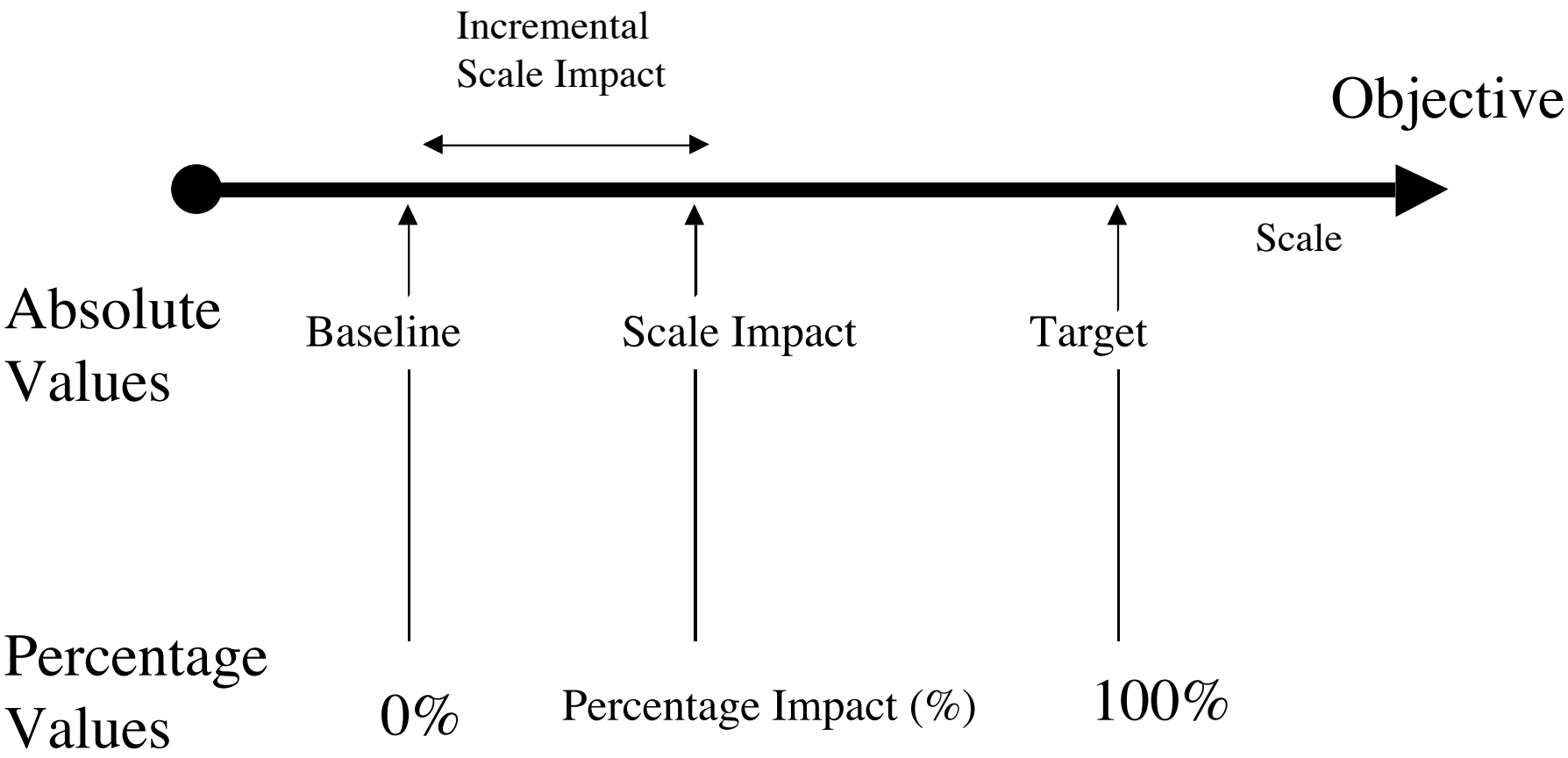
Credibility: <Credibility 0.0 low to 1.0 high. Rate the credibility of your estimates, based on the evidence and its source>.

Impact Estimation:

How much do designs impact all critical cost and quality attributes?

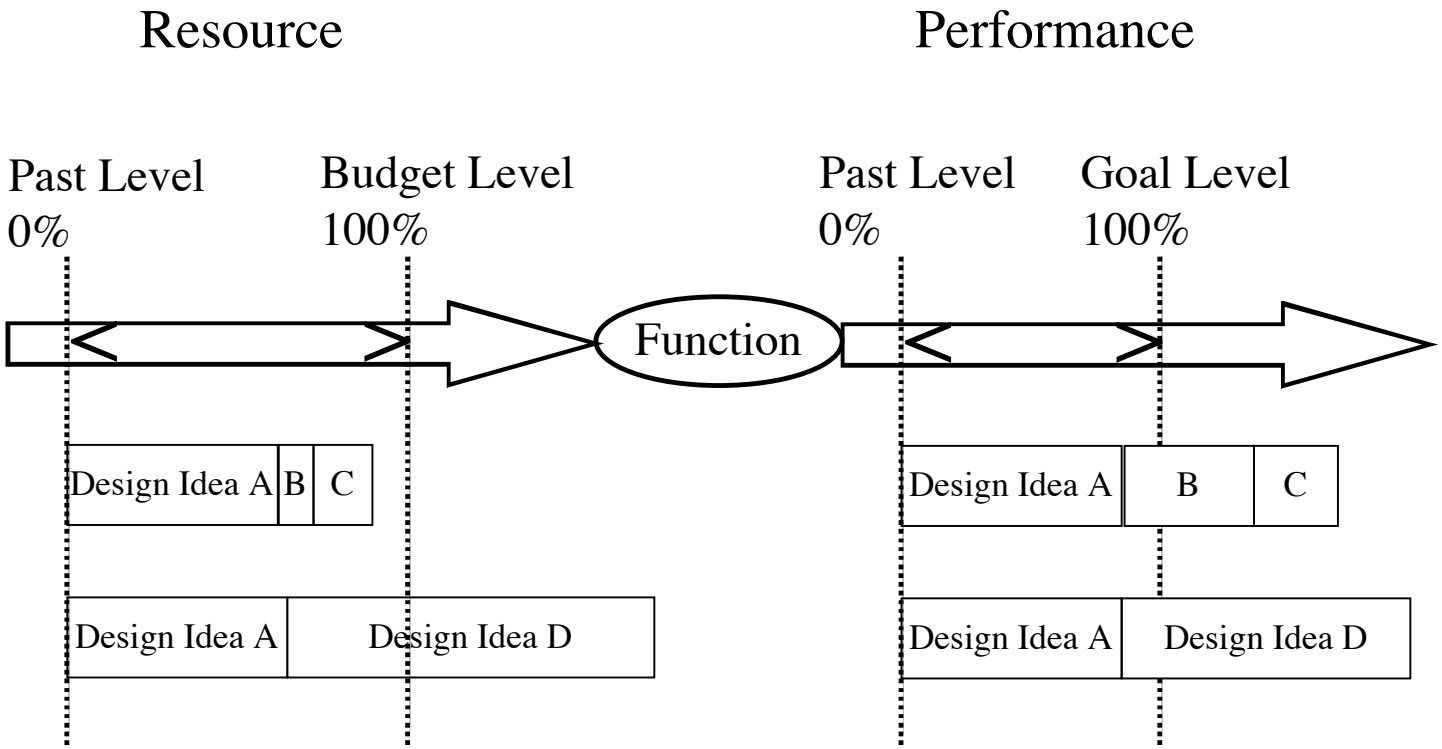


Impact Estimation Basic Concepts



- Source: Lindsey Brodie, Editor of Competitive Engineering May 2000

How do we evaluate a single dimension of impact?



We must estimate or measure the numeric cumulative impact of the design on a defined Scale:

- using a defined Meter (or estimates)
- with respect to target (Goal, Stretch, Wish) and possible constraint levels (Fail, Survival, Tolerable, Worst Case)

Nordic Road Building Software IE
"Look for high impact numbers"
to identify promising Evo steps

	Road Design Functions						Road Data Model		Drawing Production		
	Road Standard (Requirements)	Road Network	Alignment Design	Road modelling	Intersection modelling (3D!)	Analyse the Design	Storage of road model	Storage of Alignments	Drawing Functions	Drawing Factory	CAD Compon
Product Qualities											
Efficiency.Design,	5%	30%	20%	40%	15%	20%	10%	15%	30%	20%	0%
Efficiency.Construction	0%	5%	0%	40%	20%	10%	10%	0%	0%	0%	0%
Efficiency. Facility management	0%	20%	0%	10%	5%	0%	10%	10%	0%	0%	0%
Efficient.Localisation	-20%	0%	0%	0%	15%	-5%	10%	0%	30%	20%	0%
Quality.Localisation	-20%	0%	0%	0%	0%	0%	10%	0%	20%	15%	0%
Usability.Learnability	0%	10%	30%	30%	15%	-5%	5%	10%	10%	10%	0%
Usability.Intuitive	-5%	10%	20%	30%	15%	-5%	10%	10%	10%	10%	0%
Usability.Fun	10%	10%	20%	20%	10%	5%	5%	0%	15%	15%	0%
Usability.Workflow	20%	40%	10%	20%	15%	0%	5%	10%	10%	10%	0%
Availability.Reliability	0%	-10%	-10%	-10%	-10%	0%	10%	0%	5%	5%	0%
Availability.Maintainability	0%	-10%	-10%	-10%	-10%	0%	10%	0%	5%	5%	0%
Availability.Scaleability	0%	-10%	-10%	-10%	20%	0%	20%	0%	10%	10%	0%
Portability	0%	0%	0%	0%	20%	0%	15%	10%	10%	10%	0%
Identity. Novapoint	30%	30%	30%	0%	10%	15%	30%	10%	5%	5%	0%
	20%	**125%**	**100%**	**160%**	**140%**	**35%**	**160%**	**75%**	**160%**	**135%**	**0%**
Engineers.Innhouse											
15,000	300	1000	80	1000	1000	100	2500	100	0		
Engineers.External											
Thai	300								1000		
Vietnam						300					
Partners		300	200		1000			80			
Sweden										800	
Denmark											
Finland											
Others											
Total Development Resources	600	1300	280	1000	2000	400	2500	180	1000	800	
Benefit / Dev. Resources	0.03%	0.10%	**0.36%**	**0.16%**	0.07%	0.09%	0.06%	**0.42%**	**0.16%**	**0.17%**	0
			2	3				1	3	4	

US Army Example: PERSINSCOM

STRATEGIES ➔ OBJECTIVES	Technology Investment	Business Practices	People	Empow-erment	Principles of IMA Management	Business Process Re-engineering	SUM
Customer Service ?➔0 Violation of agreement	50%	10%	5%	5%	5%	60%	185%
Availability 90% ➔ 99.5% Up time	50%	5%	5-10%	0	0	200%	265%
Usability 200 ➔ 60 Requests by Users	50%	5-10%	5-10%	50%	0	10%	130%
Responsiveness 70% ➔ ECP's on time	50%	10%	90%	25%	5%	50%	180%
Productivity 3:1 Return on Investment	45%	60%	10%	35%	100%	53%	303%
Morale 72 ➔ 60 per mo. Sick Leave	50%	5%	75%	45%	15%	61%	251%
Data Integrity 88% ➔ 97% Data Error %	42%	10%	25%	5%	70%	25%	177%
Technology Adaptability 75% Adapt Technology	5%	30%	5%	60%	0	60%	160%
Requirement Adaptability ? ➔ 2.6% Adapt to Change	80%	20%	60%	75%	20%	5%	260%
Resource Adaptability 2.1M ➔ ? Resource Change	10%	80%	5%	50%	50%	75%	270%
Cost Reduction FADS ➔ 30% Total Funding	50%	40%	10%	40%	50%	50%	240%
SUM IMPACT FOR EACH SOLUTION	***482%***	***280%***	***305%***	***390%***	***315%***	***649%***	
Money % of total budget	15%	4%	3%	4%	6%	4%	
Time % total work months/year	15%	15%	20%	10%	20%	18%	
SUM RESOURCES	***30***	***19***	***23***	***14***	***26***	***22***	
BENEFIT/RESOURCES RATIO	**16:1**	**14:7**	**13:3**	**27:9**	**12:1**	**29:5**	

ImpactEstimation

A set of 12 proposed engineering processes

Defined slide 26

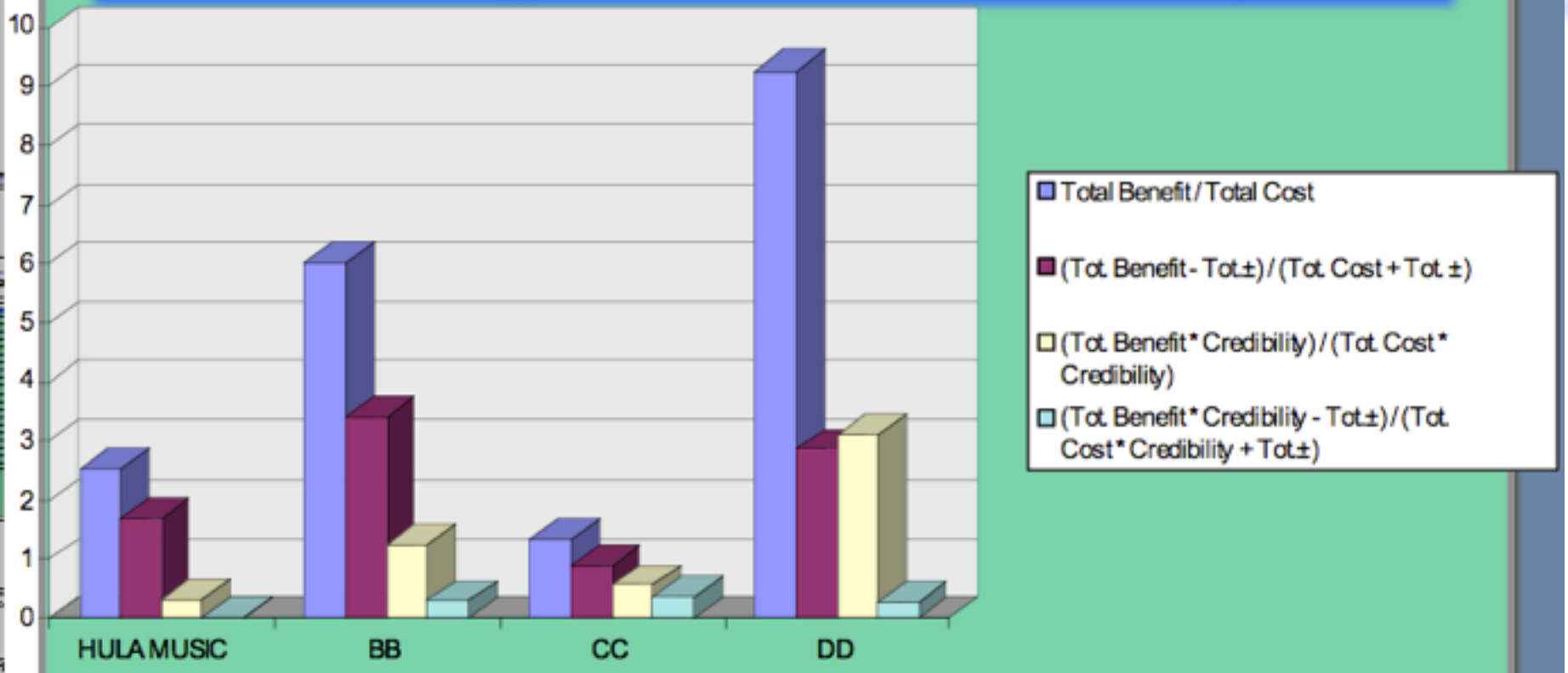
	Deliverables						
	Telephony	Modularity	Tools	User Experience	GUI & Graphics	Security	Enterprise
Business Objective							
Time to Market	10%	10%	15%	0%	0%	0%	5%
Product Range	0%	30%	5%	10%	5%	5%	0%
Platform Technology	10%	0%	0%	5%	0%	10%	5%
Units	15%	5%	5%	0%	0%	10%	10%
Operator Preference	10%	5%	5%	10%	10%	20%	10%
Commoditization	10%	-20%	15%	0%	0%	5%	5%
Duplication	10%	0%	0%	0%	0%	5%	5%
Competitiveness	15%	10%	10%	10%	20%	10%	10%
User Experience	0%	20%	0%	30%	10%	0%	0%
Downstream Cost Saving	5%	10%	0%	10%	0%	0%	5%
Other Country	5%	10%	0%	10%	5%	0%	0%
Total Contribution	90%	80%	55%	85%	50%	65%	55%
Cost (£M)	0.49	1.92	0.81	1.21	2.68	0.79	0.60
Contribution to Cost Ratio	**184**	42	68	70	19	82	92

- A set of 12 proposed engineering Deliverables, for about $100,000,000 of investment projected over time, are evaluated theoretically for their impact on 13 Business Objectives (as defined in previous slide).
- This real example is altered substantially to protect confidentiality. It appropriately ignited the imagination of top management to really plan their engineering business in a quantified manner.
- Notice the overall impact to cost ratio (ROI Index) is estimated for each process. The actual definitions of the strategy deliverables are elsewhere, and are confidential. But that detail would be needed to estimate and to check these estimates

6. Use multiple metrics to compare alternatives

- one way to compare any set of alternatives is
  - –To compare their quality and cost attributes
  - –In relation to your needs (requirements)

Benefit to Cost ratios with regard to risk and credibility



How does Impact Estimation relate to Planguage Specification?

Learning:

Ambition: Make it substantially easier for our users to learn tasks <- Marketing.

Scale: Average time for a defined [User Type: default UK telesales trainee] to learn a defined [User Task: default Response] using <our product's instructional aids>.

Response: Task: Give correct answer to simple request.

Past [last year]: **60** minutes.

GN: Goal [By start of next year]: 20 minutes.

GA: Goal [By start of year after next]: **10** minutes.

	On-line Support	On-line Help		Picture Handbook	On-line Help + Access Index
Learning Past: 60min. <<-> Plan: 10min.					
Scale Impact	5 min.	10 min.		30 min.	8 min.
Scale Uncertainty	±3min.	±5 min.		±10min.	±5 min.
Percentage Impact	110%	100%		67% (2/3)	104%
Percentage Uncertainty	±6% (3 of 50 minutes)	±10%		±20%?	±10%
Evidence	Project Ajax, 1996, 7 min.	Other Systems		Guess	Other Systems + Guess
Source	Ajax report, p.6	World Report p.17		John B.	World Report p.17 + John B.
Credibility	0.7	0.8		0.2	0.6
Development Cost	120K	25K		10K	26K
Benefit-To-Cost Ratio	110/120 = 0.92	100/25 = 4.0		67/10 = 6.7	104/26 = 4.0
Credibility-adjusted B/C Ratio (to 1 decimal place)	0.92*0.7 = 0.6	4.0*0.8 = 3.2		6.7*0.2 = 1.3	4.0*0.6 = 2.4
Notes: Time Period is two years.	Longer timescale to develop				

Picture Handbook: Gist: Produce a radically changed handbook that uses pictures and concrete examples to *instruct*, without the need for *any* other text.

7. Measure critical variables, but with sufficient qualities and lowest costs

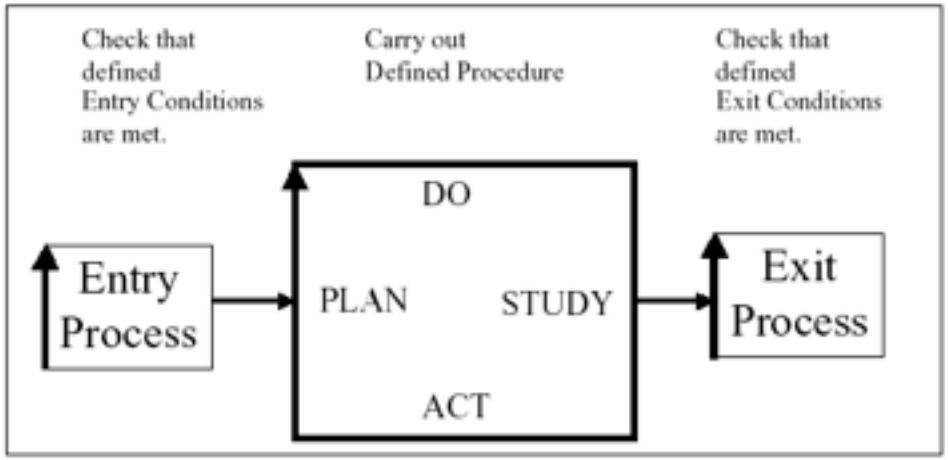
- Quantification *seems* exact: 5.0, 3.14
  - even though it is an approximation.
- Measurement is
  - determining where we really are
  - along a scale of measure,
  - in relation to benchmark level, constraint levels, and target levels.
- Measurement cannot be perfect.
  - Perfect measurement has infinite cost
  - Measurement needs to be sufficient for purpose
  - at the lowest costs for that purpose
  - Measurement processes can be 'designed' to fit a set of numeric qualities, costs, and constraints

8. Use metrics to review specifications

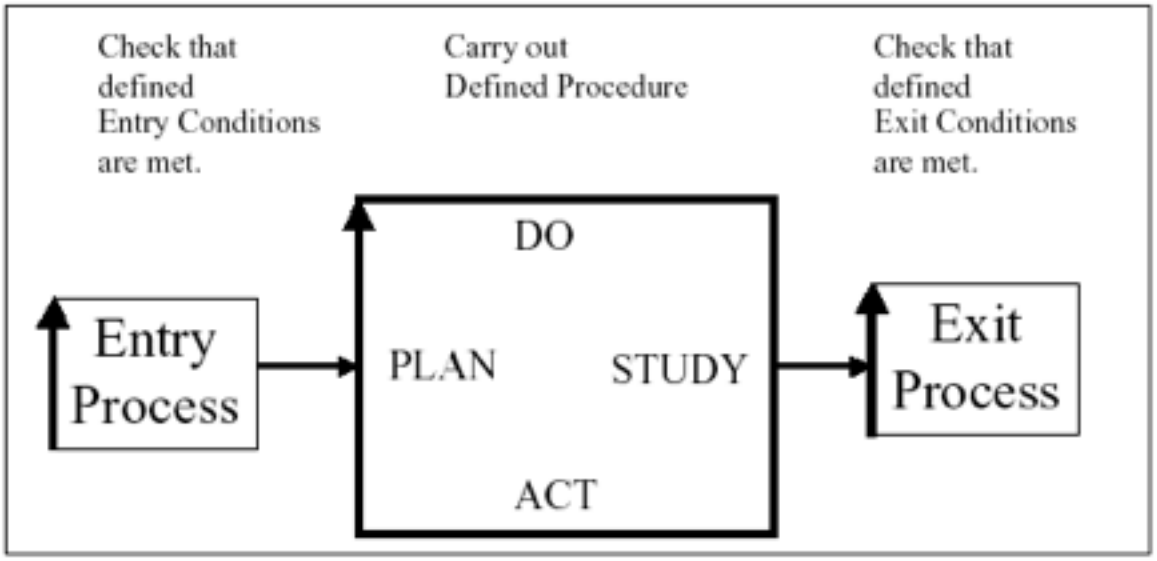
- basic metric: major defects per 300 words
  - –Major: can threaten to hurt the system
  - –Defects: deviations from our standards for how to write the specs
    - Examples (see CE book for many Rules)
      - – The spec must be unambiguous to the intended readership
      - – All qualities must be quantified
      - – All design impacts must be estimated

The process format used for Planguage process descriptions consists of three basic elements

- **Entry** Conditions : to determine whether it is wise to start the procedure.
- **Procedure** : specifying for a task what work needs to be done and how best to do it.
- **Exit** Conditions : to help determine if the work is 'truly finished'.



The quantified Exit and Entry controls



- Entry and Exit Condition example:
- Maximum estimated 1.0 Major defects per logical page remaining.
- This was the MOST important lesson IBM learned about software processes (source Ron Radice, co-inventor Inspections, Inventor of CMM)

Entry Exit Control

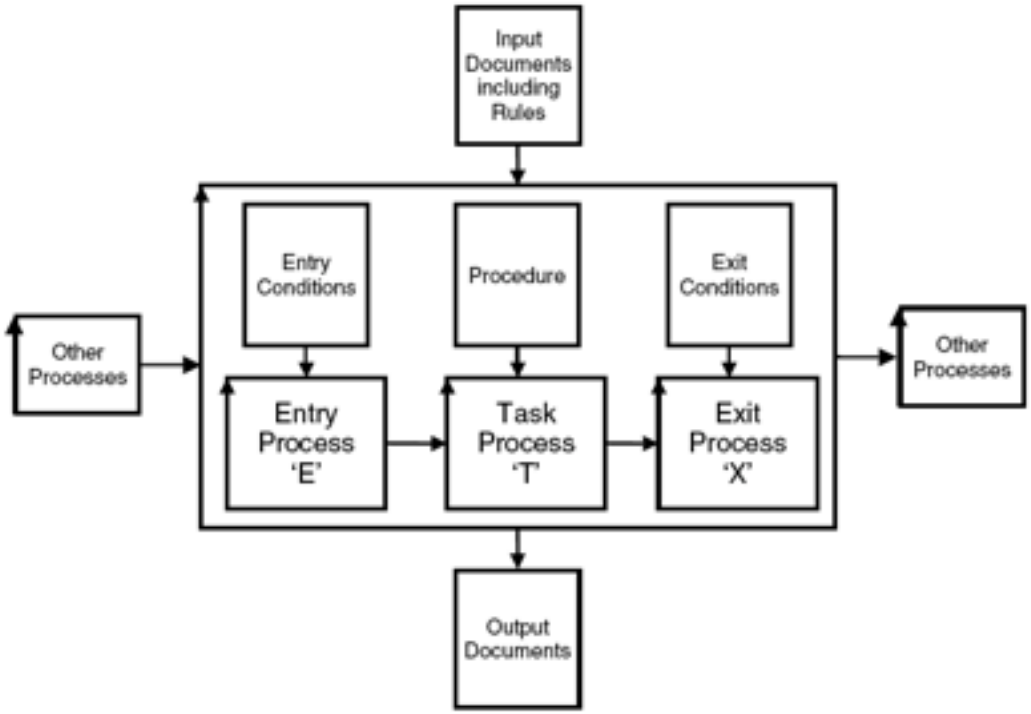
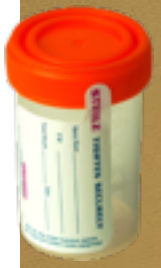


Figure 1.4

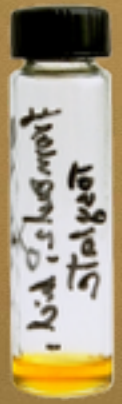
- Diagram of a simple process showing its sub-processes and its relationship to other processes and documents.
- The input documents for each process include the rules, the entry conditions, the procedure and the exit conditions.
- The diagram also shows how the 'ETX' concept for a process is derived.
  - A rectangle is the symbol for a 'written document.'
  - A rectangle with arrow is a 'process' symbol.
    - An example of such a process could be 'Requirement Specification.' <- CE, figure 1.4

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.

---

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) MM-

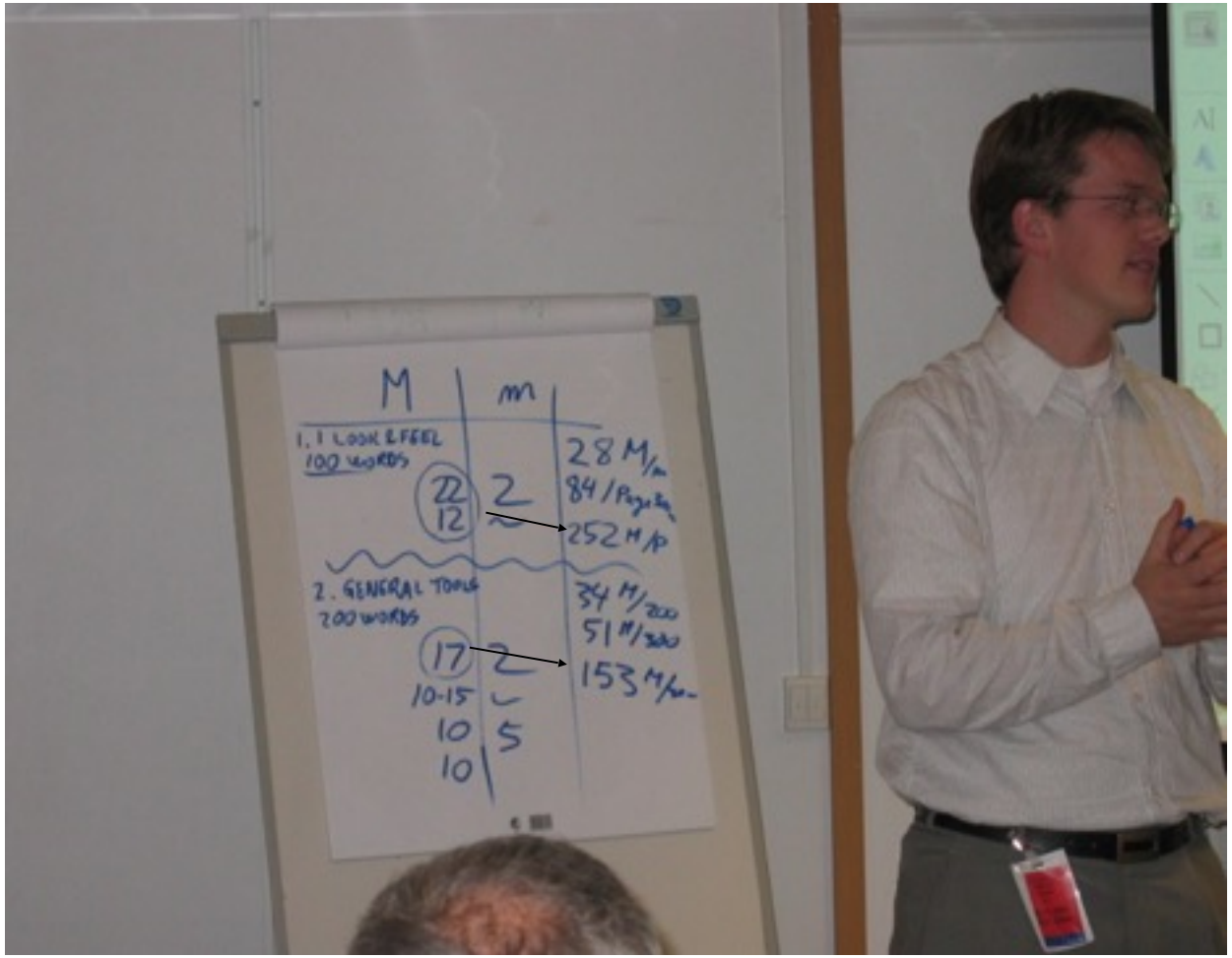
- Upon starting up the display the mapped area will correspond to the one displayed upon last use, or the entire project area if no previous area default is available.
- Upon starting up the display the layer settings (which layers are visible, annotation setting etc.) will be retained.

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.

<- See rewrite of this on later slide

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

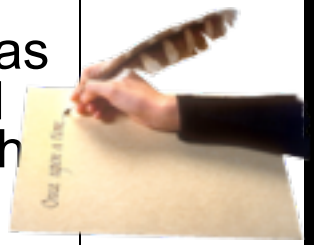


Rewrite of a real Defective ‘Requirement at (Norway, 2004)

- 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)

Analysis

Usability.Intuitive
Ambition: after initial training, (one week course, two week field) the user shall not have to refer to 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 perceived by the Boss”*
Record [Product = 408] ??%

Past [Elements = Finding a menu option, User = Beginner, 2004] 40%$^{\pm 20}$?? <- Will 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

The 'Real' Requirement in Planguage

9. Use metrics to prioritize, and determine priorities

- I argue that traditional weighting metrics are a very bad way of communicating priorities for requirements
  - what are your weights for eating, breathing, drinking?
- I would argue that the natural and logical way to understand priorities is in terms of
  - quantified requirements, and
  - repeated continuous measurement of the satisfaction
  - the more satisfied a requirement,
    - The lower the priority

See detailed papers at www.gilb.com,
Choice and Priority Using Planguage: http://www.gilb.com/community/tiki-download_file.php?fileId=48
Managing Priorities: http://www.gilb.com/community/tiki-download_file.php?fileId=60

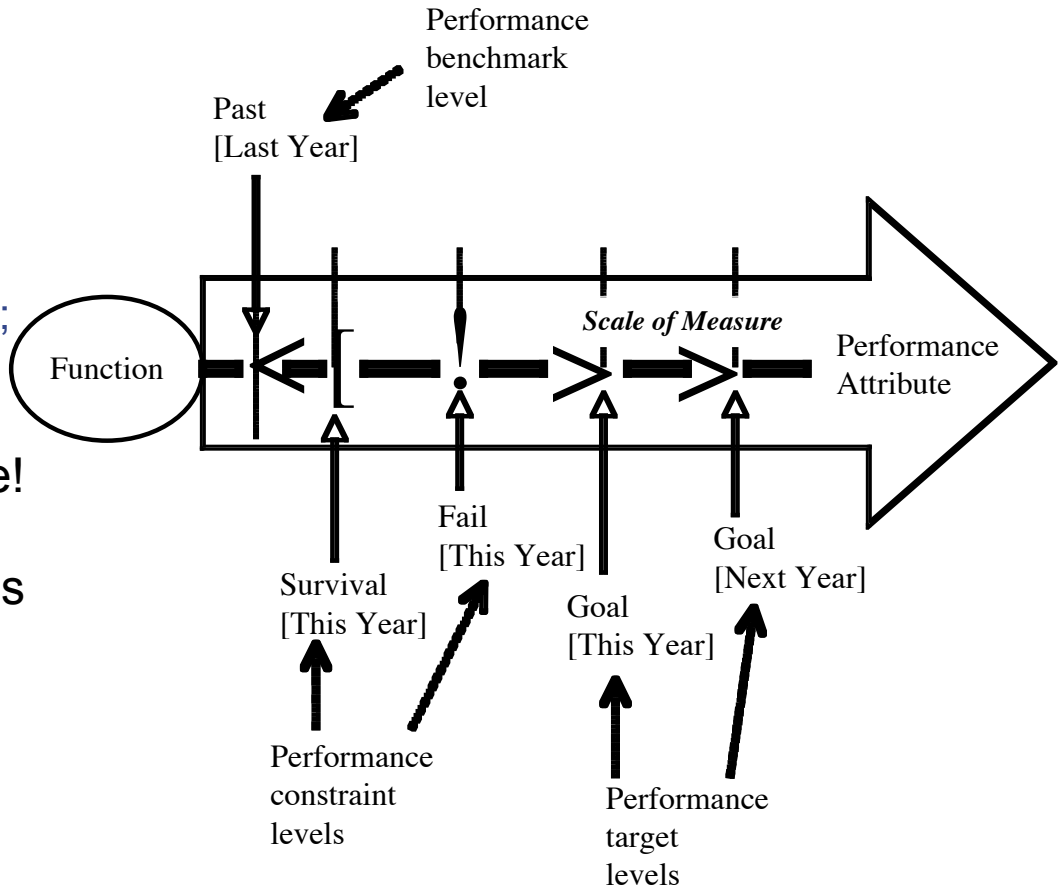
Priority Management

Priority is
- Claim on scarce or limited resources

It 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



- 6.0 is a much clearer notion than ‘very much’
- If we agree to ‘extremely good X’
  - How much have we agreed to?

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

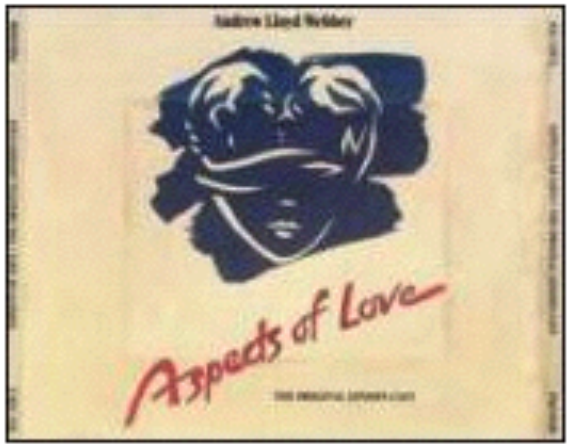
- Make inventory of love's many aspects
- Quantify one requirements for love

- Duration: 6 minutes

See note for Sutra

- 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, Boeing

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

real case

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

real case

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	Stuart Papworth		**Assigned to**		
Stakeholders					
Ambition	R[illegible] 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	[1st 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				

real case

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		Stuart Papworth	**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 c 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		St	**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. *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.* *Note 2: exceptions, short circuit? – cost implications, under investigation. <- Tho*			
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 Detail on next slide

real case

real case

Detail of Scale for 'System Overhead Time' requirement

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. *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.* *Note 2: exceptions, short circuit? – cost implications, under investigation. <- T*

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, 1st Release] 5 minutes? <- BN [Central System Hardware, 1st 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}.				

Quality Requirement: Recoverability

- Notice:
  - –multiple Goal Levels
  - –Parameterized Scale

4.1.1. Readiness

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		S[illegible]		**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		[illegible] 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[redacted]		**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				

real case

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						

Developed by BN

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.

Source: Hughes, 1998, 'Rescuing Prometheus', p164.

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.]

- Move from software engineering to systems engineering

Summary - Final Slide

– Metrics give us a powerful tool to describe, communicate, and exercise management control over software and systems development

–Planguage is a specific defined and free tool for expressing metrics ideas about software and systems components.