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

Exclusively for UK Software Metrics Association

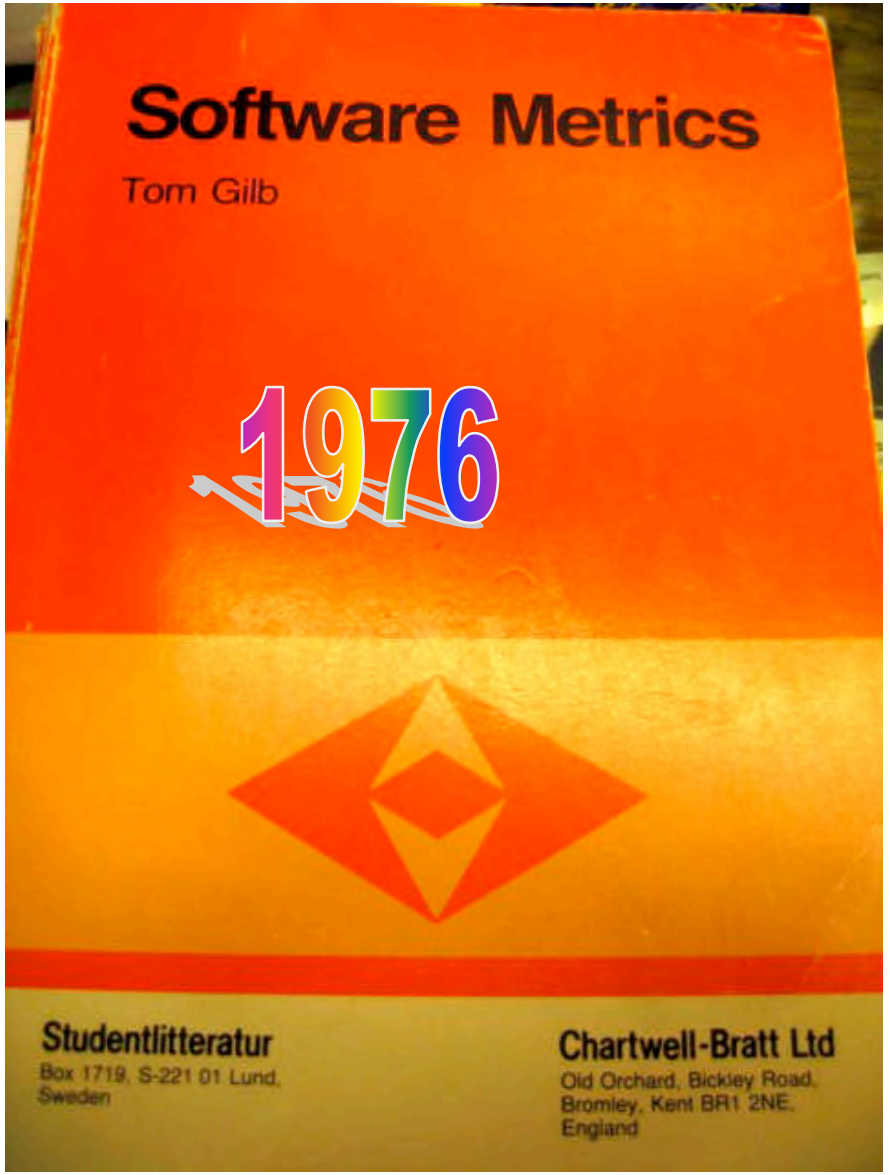
09:45 - 10:30 , 16 October 2007, London

These slides
(96-slide long-version)
are in pdf at
www.gilb.com,
Free Downloads,
Gilb Library.
See last slide here.

Books 1976, 1977 and 2005

1977 USA

2005



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

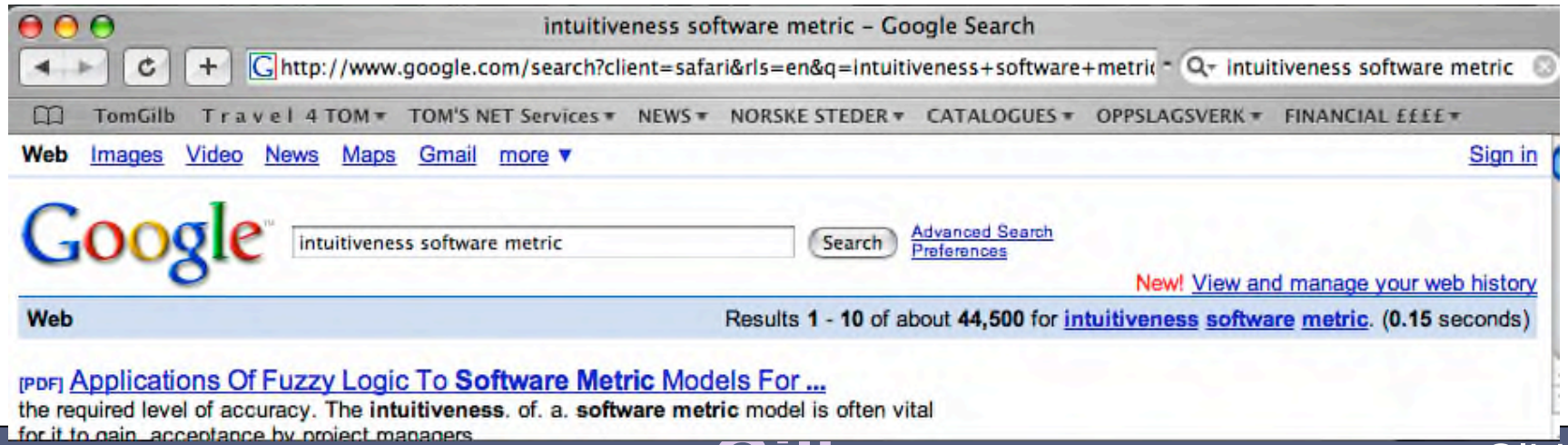
Tom Gilb

Part I of II

Ten fundamental software metrics principles, for failure

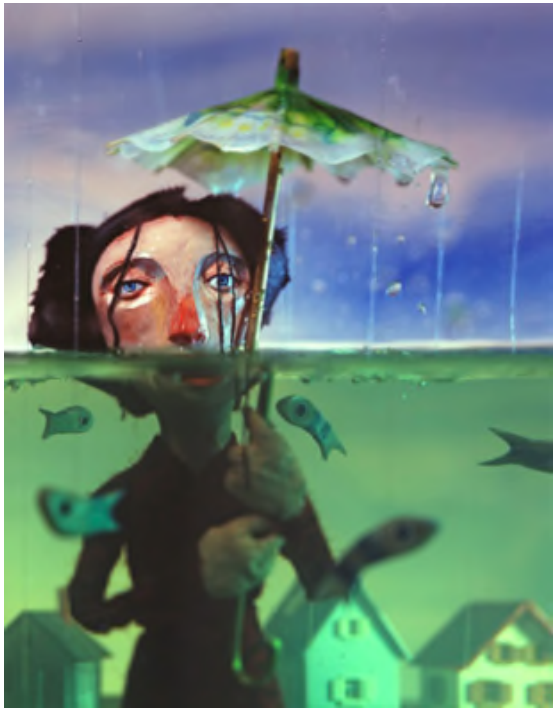
1. If you measure what is easy rather than right , you'll lose the fight.

- 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.



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

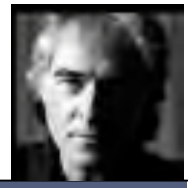
- you need to measure early, in order to discover -
  - *what* to measure, what the requirements really are
  - what measures are *useful*
  - what is *worth* measuring
  - what the actual numeric levels of requirements *should* be
- Measuring at the end of a project,
  - is just too late to learn in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



3. If you measure too few, then the ones you left out, will lack any 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.

Alfred-Gockel



4. If the metric level is too low, your users *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

5. Know the role of your metric, or it can roll over on your project

- 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 - the metric ‘role’ in the ‘play

6. If you fail to quantify a critical variable, it will fail to be what you need

- 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"

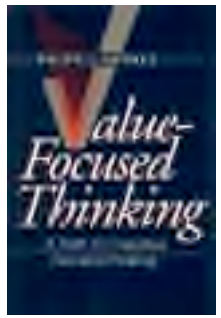
7. Do not trust managers to define the most critical metrics, help them out

- 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)

knowledge base	win-win	24/7	performance indicators	best practice
fast track	empower	risk management	revisit	blame culture
result-driven	value-added	silo	out of the loop	pinch point
at the end of the day	benchmark	core business	touch base	synergy
I hear what you say	networking	differential	the big picture	ballpark

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

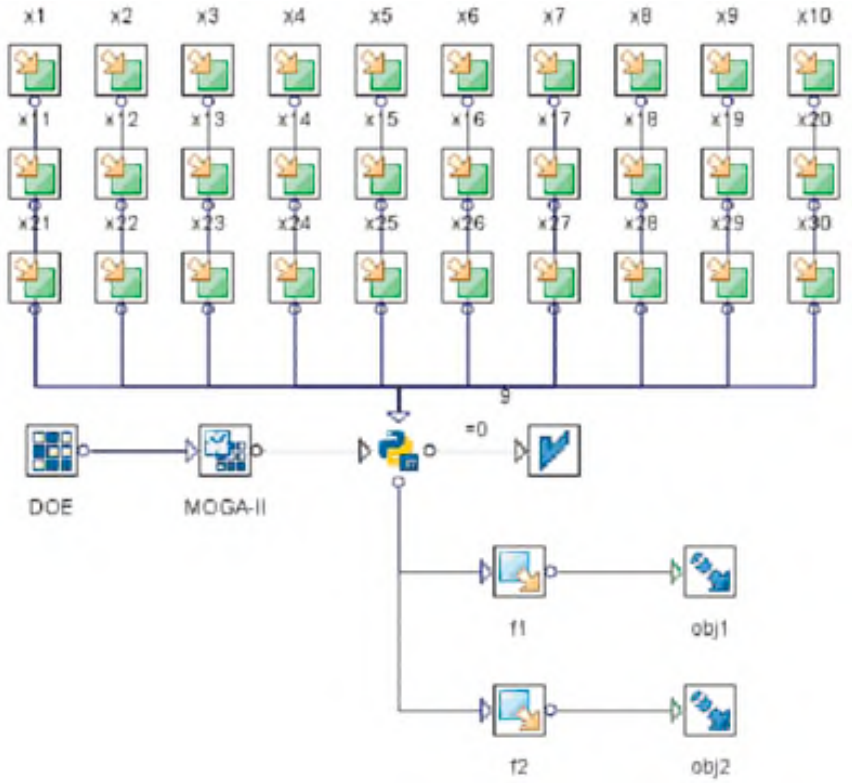
- Ralph Keeney's Levels ('Value-Focused Thinking')
  - Fundamental Objectives
  - Strategic Objectives
  - Means Objectives



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

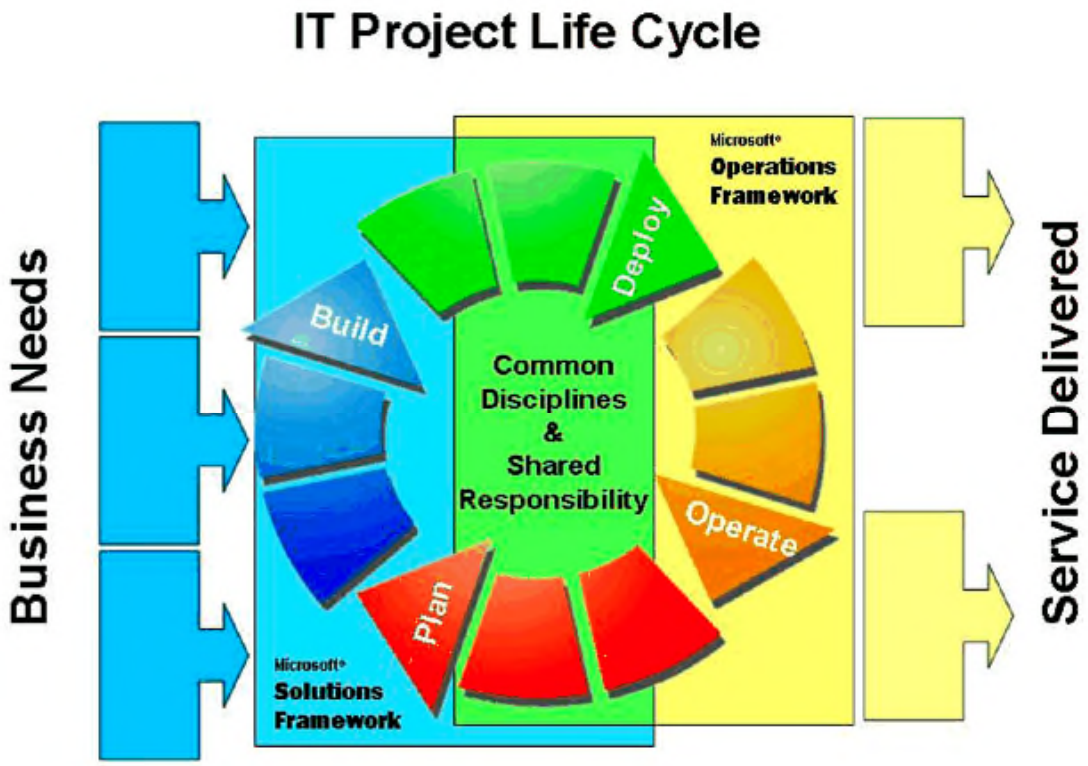
9. Metrics don't add up, but you need to understand the set of them

- 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

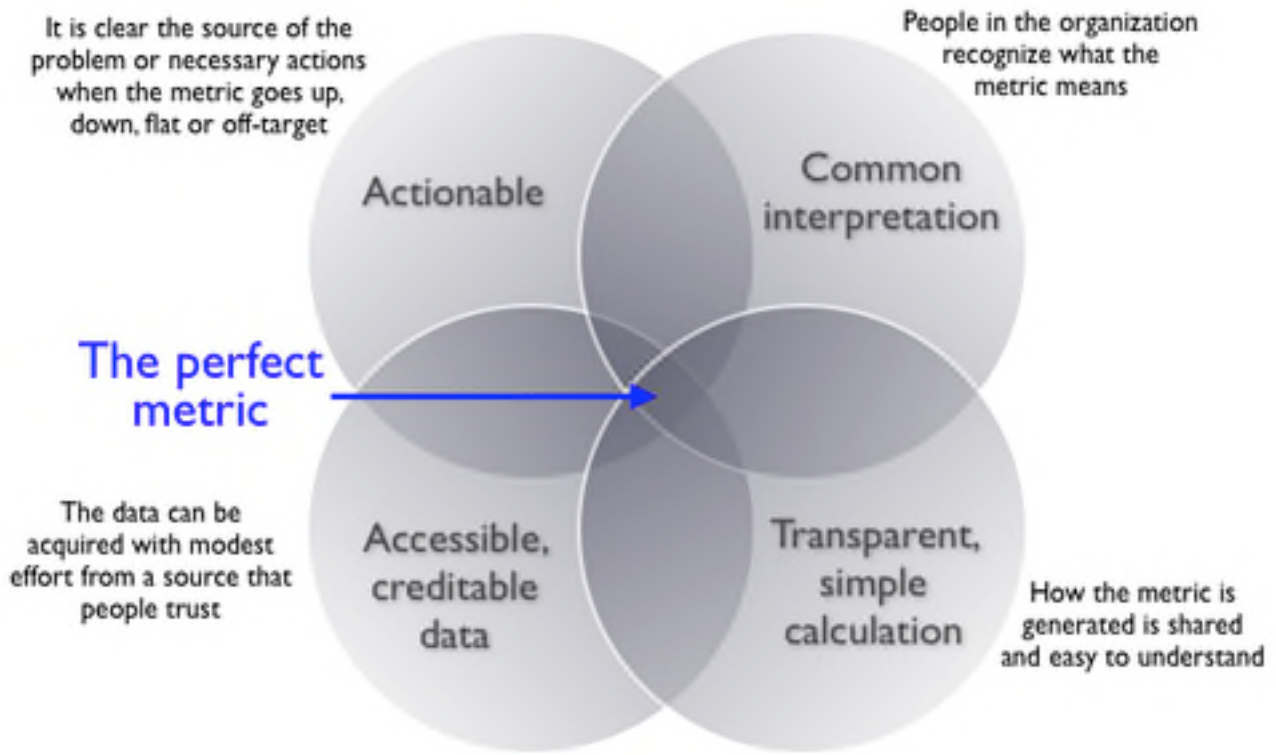


10. Metrics are a generally good tool, until they are used carelessly, or to manipulate people.

- 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



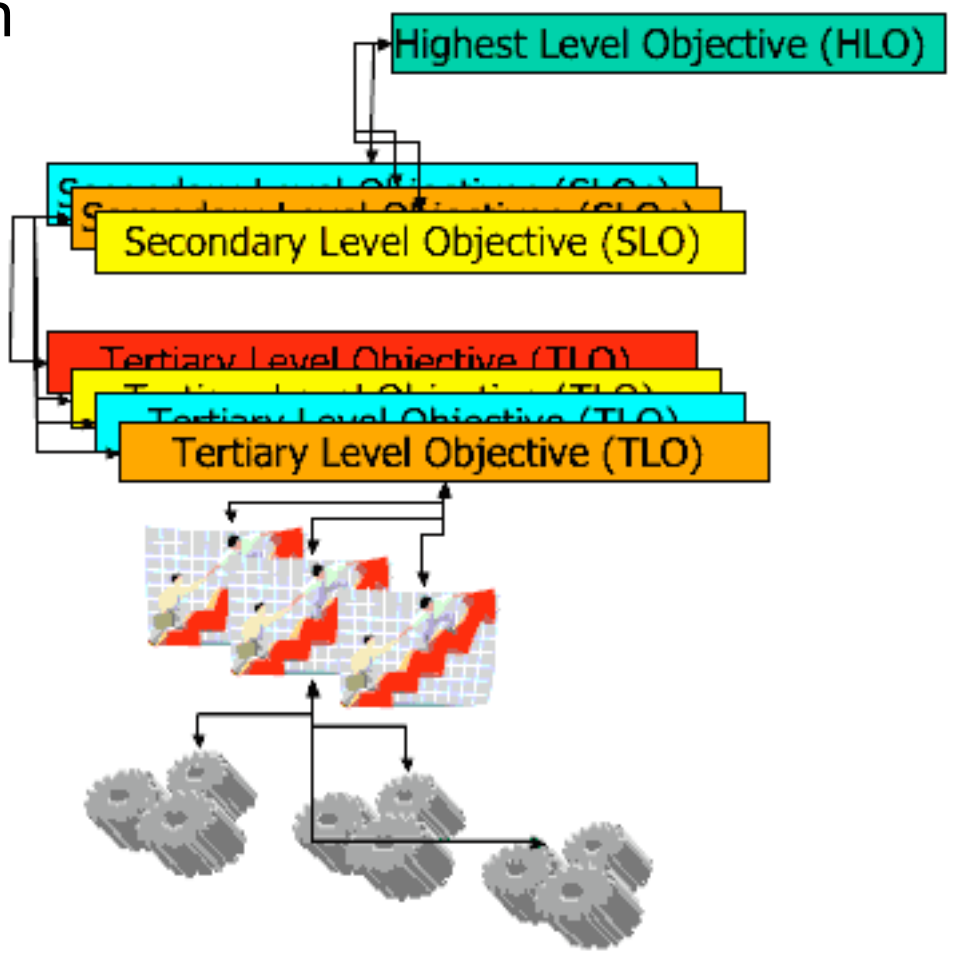
Part II:
Ten critical software metrics usage principles for success in the commercial environment



http://media.juiceanalytics.com/images/blog/metrics_framewor.pngk

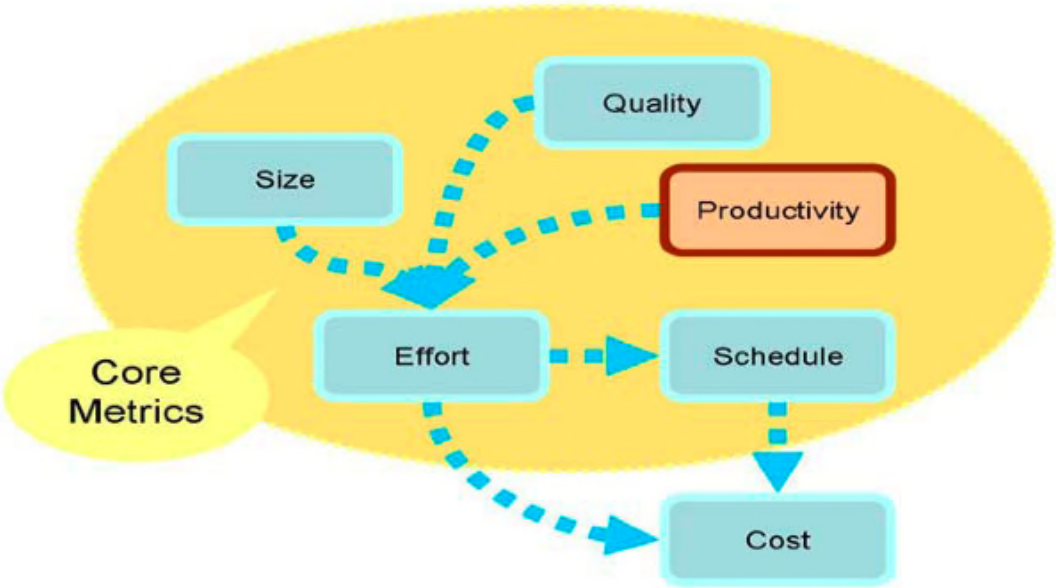
1. Develop requirements metrics top down from critical management objectives.

- 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



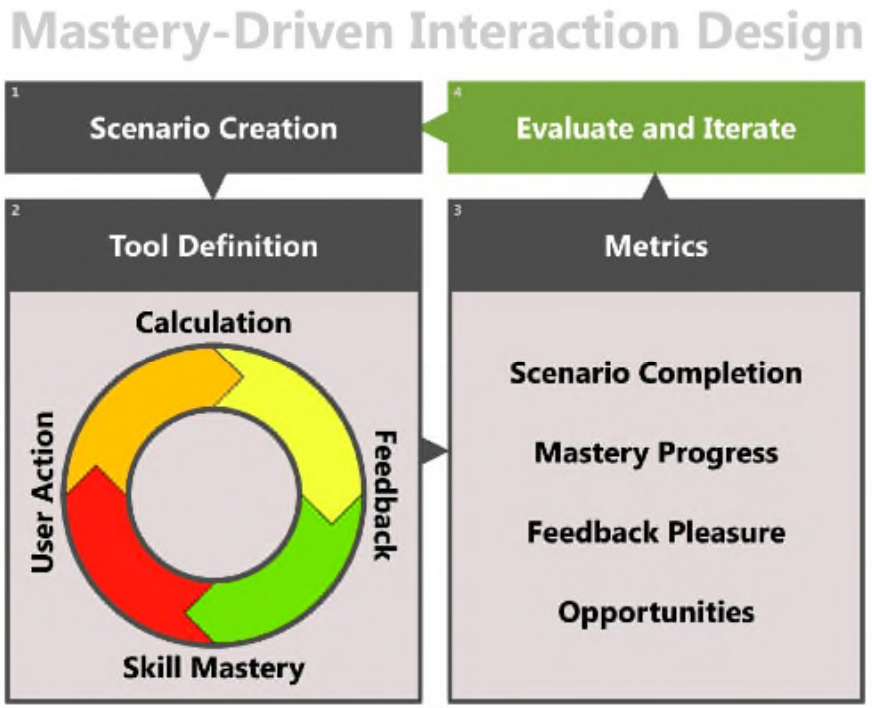
2. Connect metrics with metrics.

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



3. Develop metrics with early rapid numeric and non-numeric feedback.

- 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!



http://www.lostgarden.com/evolutionary_game_design.htm

4. Use metrics to describe metrics, credibility, uncertainty

- 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.

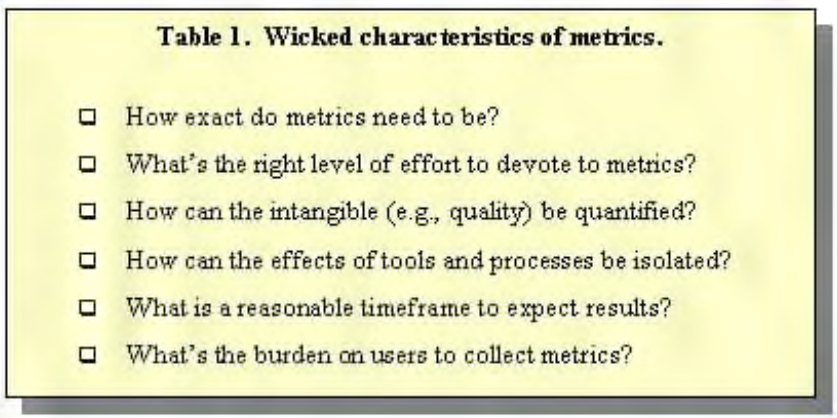
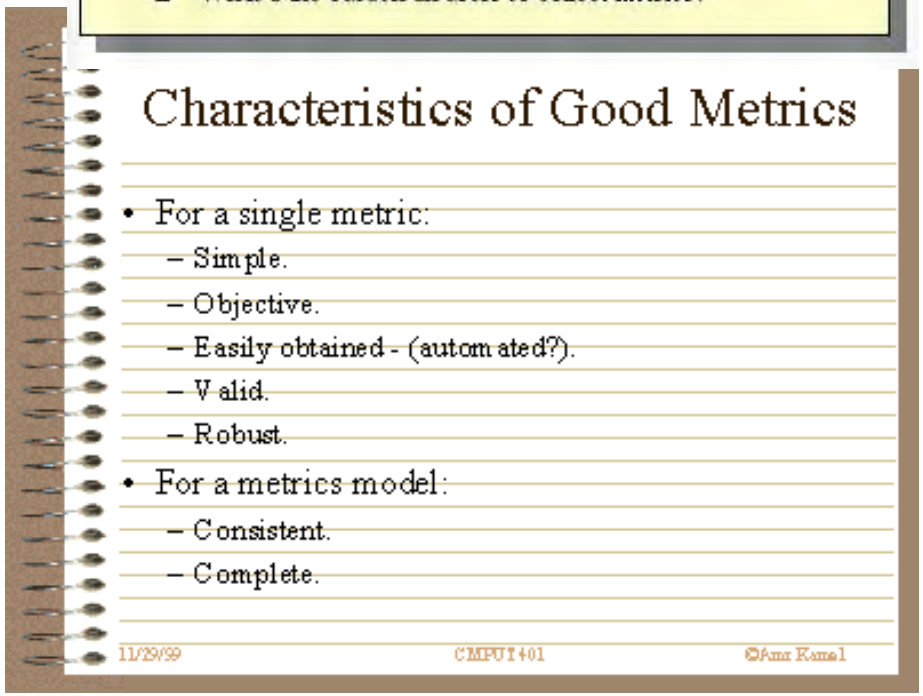


Table 1. Wicked characteristics of metrics.

- How exact do metrics need to be?
- What's the right level of effort to devote to metrics?
- How can the intangible (e.g., quality) be quantified?
- How can the effects of tools and processes be isolated?
- What is a reasonable timeframe to expect results?
- What's the burden on users to collect metrics?



Characteristics of Good Metrics

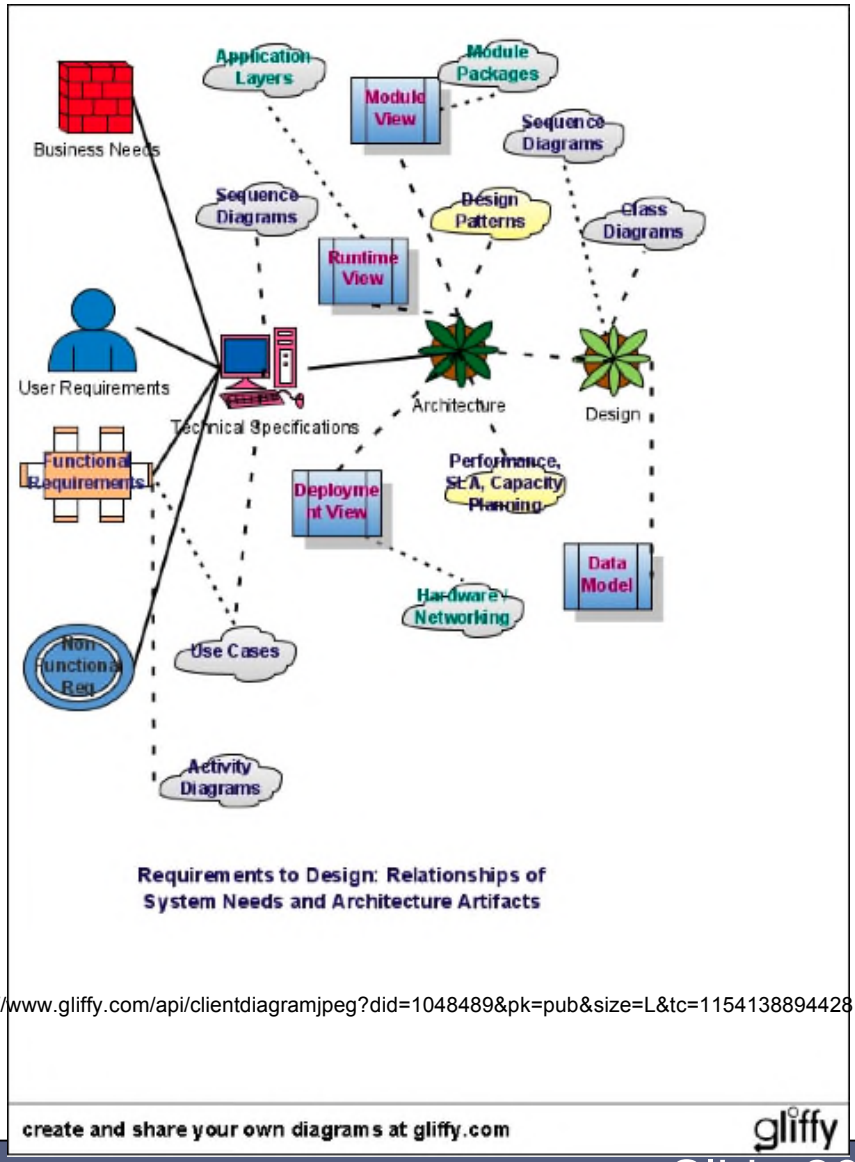
- For a single metric:
  - Simple.
  - Objective.
  - Easily obtained - (automated?).
  - Valid.
  - Robust.
- For a metrics model:
  - Consistent.
  - Complete.

11/29/99 CMPUT401 ©Amr Kamel

http://www.cs.ualberta.ca/~sorenson/cmput401/lectures/SoftwareMetrics/sld005.htm
http://collaboration.mitre.org/practguide/PGTable1.jpg

5. Use metrics to describe solutions, designs, and architecture

- 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)



http://www.gliffy.com/api/clientdiagramjpeg?did=1048489&pk=pub&size=L&tc=1154138894428

6. Use multiple metrics to compare alternatives

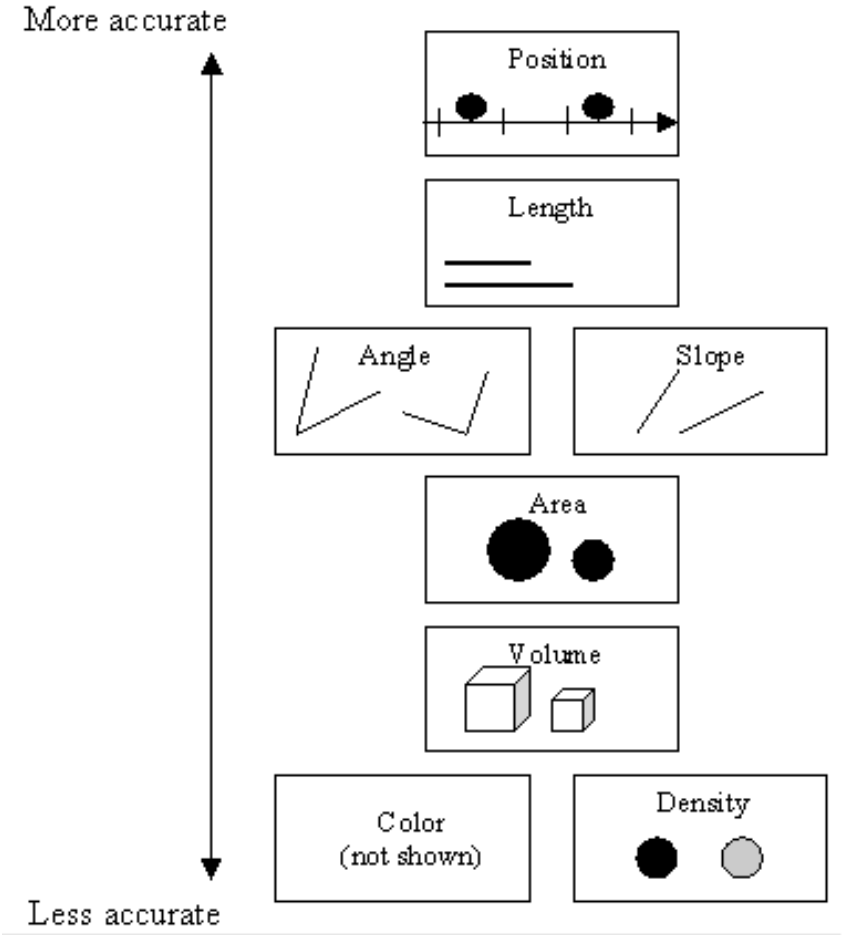
Technical Specifications	Palm Treo 680	Apple iPhone
Screen resolution	320 x 320	320 x 480
Input method	Touchscreen Full QWERTY Keyboard	Multi-touchscreen Virtual On-Screen Keyboard
Operating system	PalmOS 5.4.9	Mac OS X
Storage (Internal)	128MB	4GB or 8GB
Storage (External)	SD Card: Up to 8GB	Not Available
GSM	Quad band (850/900/1800/1900 MHz)	Quad-band (850/900/1800/1900 MHz)
Wireless data	GSM/GPRS/EDGE Not Available Bluetooth 1.2	GSM/GPRS/EDGE Wi-Fi (802.11b/g) Bluetooth 2.0
Camera	0.3 megapixels	2.0 megapixels
Removable Battery	Yes	Not Available
Battery	4 hours Talk Data Not Available	5 hours Talk/Video/Browsing 16 hours Audio playback
Dimensions	113 x 59 x 21 mm 4.4 x 2.3 x 0.8	115 x 61 x 11.6 mm 4.5 x 2.4 x 0.46 inches
Weight	5.5 ounces / 157 grams	4.8 ounces / 135 grams
Price (Unlocked)	$399	N.A.
Price (Carrier) 4GB	$80 to $279	$499
Price (Carrier) 8GB	$140 to $339	$599
Price (Carrier)	FREE to $199	N.A.

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

Typical Values	346 cals goji	355 cals orange
Total Fat	5.7g	1g
Saturated Fat	1.1g	
Protein	10.6g	7g
Total Carbohydrate	21g	89g
Sugars	17.3g	
Sodium	24mg	0
Energy (kcal)	346	355
Calcium	112.50mg	302.40mg (30%)
Iron	8.42mg	0.76mg
Crude Fiber	7.78g	18g
Vitamin C	306mg	402mg (670%)
Carotene	7.38mg	No data
Amino Acid	8.48mg	
Thiamin (B1)	0.15mg	0.66mg (55%)
Polysaccharides	46.5mg	No data
Folate	No data	229mcg (57%)

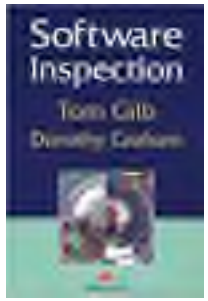
7. Measure critical variables, but with sufficient qualities and lowest costs

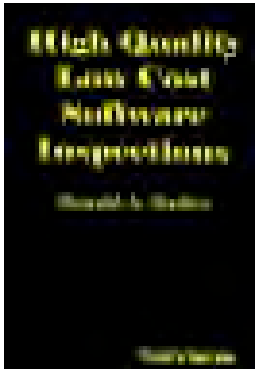
- Quantification *seems* exact: 5.0, 3.14
  - even though it is an approximation.
- Measurement is
  - determining where we really are
  - along a scale of measure,
  - in relation to benchmark level, constraint levels, and target levels.
- Measurement cannot be perfect.
  - Perfect measurement has infinite 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 [23]

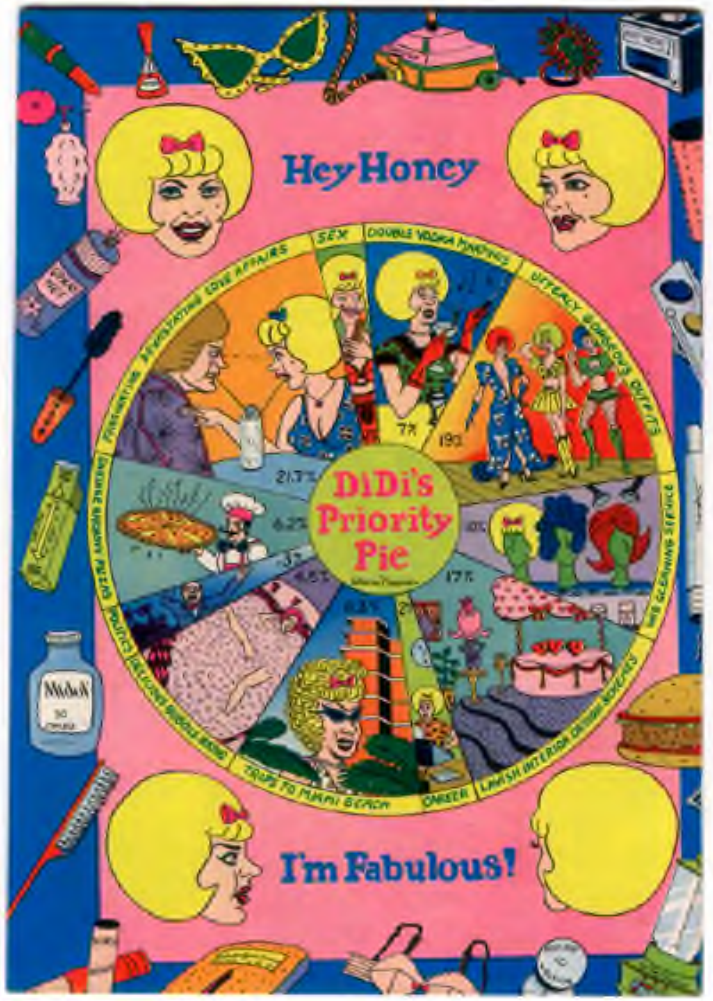
- basic metric: major defects per 300 words
  - –Major: can threaten to hurt the system
  - –Defects: deviations from our standards for how to write the specs
    - Examples (see CE book for many Rules)
      - – The spec must be unambiguous to the intended readership
      - – All qualities must be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





9. Use metrics to prioritize, and determine priorities

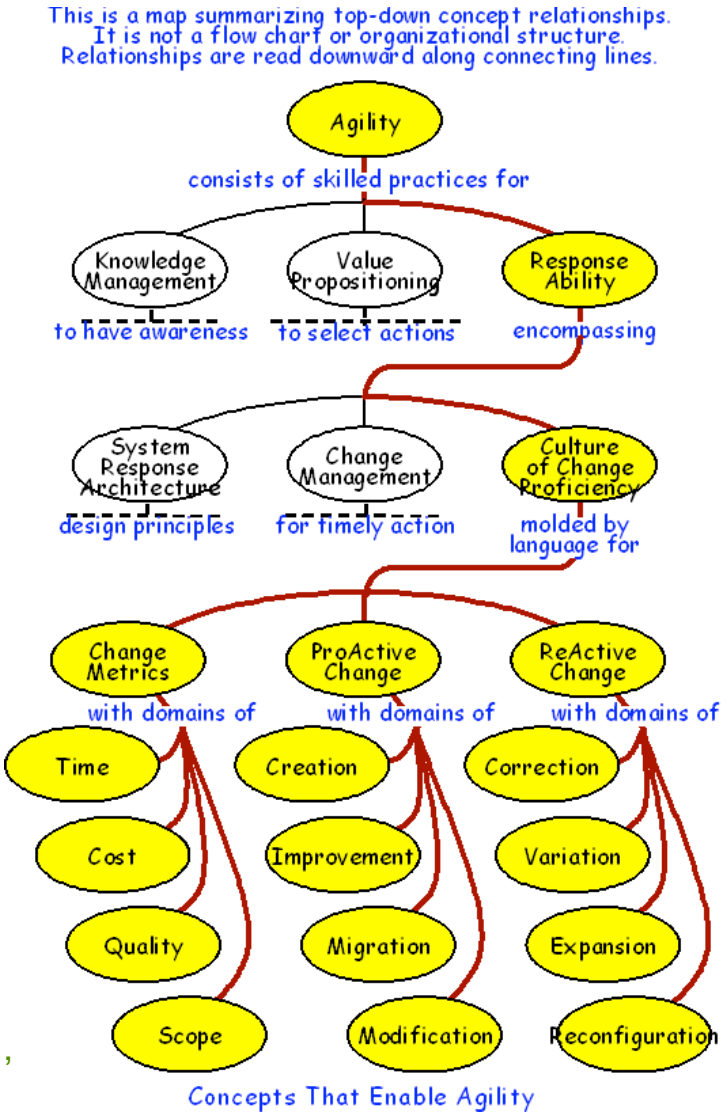
- I argue that traditional weighting metrics are a very bad way of communicating priorities for requirements
  - what are your weights for eating, breathing, drinking?
- I would argue that the natural and logical way to understand priorities is in terms of
  - quantified requirements, and
  - repeated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

10. Use metrics to create commonly understood, and really agreed requirement or objectives.

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



www.parshift.com/ Essays/images/essay067.gif , Rick Dove ----->

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.

These slides with lots of additional detail (96 slides) are at :
http://homepage.mac.com/tomgilb/filechute/Making%20Metrics%20Practical%20UKSMA%20Master.ppt
http://www.gilb.com/community/tiki-download_file.php?fileId=132 (pdf

Extra Slides if Time

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

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

- Duration: 6 minutes

See note for Sutra

Love Attributes:
Brainstormed By Dutch Engineers

- Kissed-ness
- Care
- Sharing
- Respect
- Comfort
- Friendship
- Sex
- Understanding
- **Trust**

Support
Attention
Passion
Satisfaction
...
...
...

Trust [Caroline]

Love.Trust.Truthfulness

Ambition: No lies.

Scale:

Average **Black** lies/month from [defined sources].

Meter:

independent confidential log from sample of the defined sources.

Past Lie Level:

Past [My Old Mate, 2004] 42 <- Bart

Goal

[My Current Mate, Year = 2005] Past Lie Level/2

Black: Defined: Non White Lies

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

Camaraderie (Real Case UK)

Ambition: *to maintain an exceptionally high sense of good personal feelings and co-operation amongst all staff: family atmosphere, corporate patriotism. In spite of business change and pressures.*

Scale: **probability that individuals enjoy the working atmosphere so much that they would not move to another company for less than 50% pay rise**.

Meter: Apparently real offer via CD-S

Past [September 2001] 60+ % <- R & CD

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

Rationale:

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

Love: Biblical Dimensions <- Lawrence Day, 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

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

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*

Quality Requirement: Recoverability

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}.				

- 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[redacted]		**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	[redacted] Crew2, 2004] ~30 min. ?? <-BN				
Links	req ??				

Business Objective
TTM
Same Format

2.1. *Time to market*

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

Template for Quality Requirements

Template for Quality Requirements:

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

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.]

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.

http://homepage.mac.com/tomgilb/filechute/Making%20Metrics%20Practical%20UKSMA%20Master.ppt
http://www.gilb.com/community/tiki-download_file.php?fileId=132 (pdf