“Some Principles of Useful Knowledge”

Tom Gilb

19:00-19:30

Thursday 4th June 2015

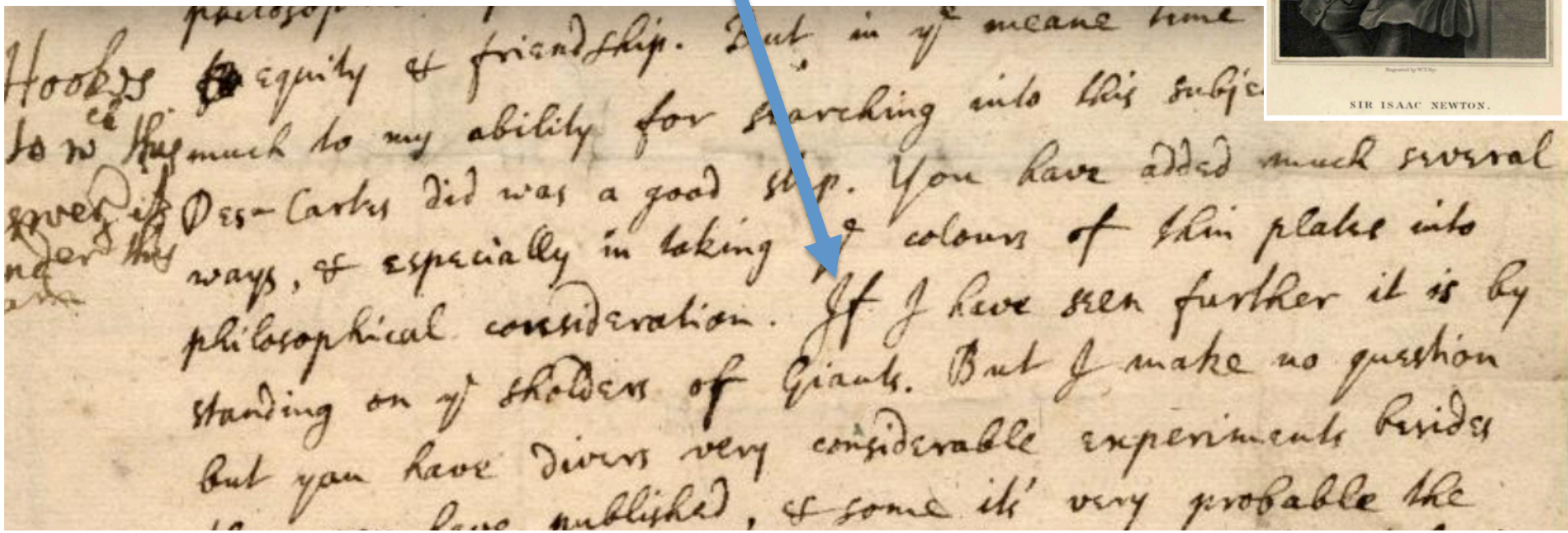
Kongsberg Systems Engineering Event (KSSE)

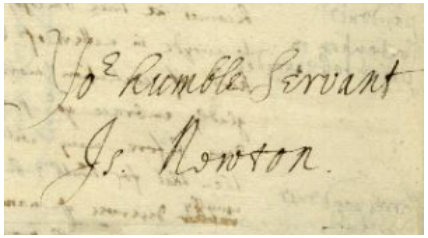
The theme in 2015 is

"**Managing knowledge: How to capture, store, find, use, and keep knowledge up-to-date?**".

"if I have seen further, it is by standing on ye 'sholders' of giants."

Newton





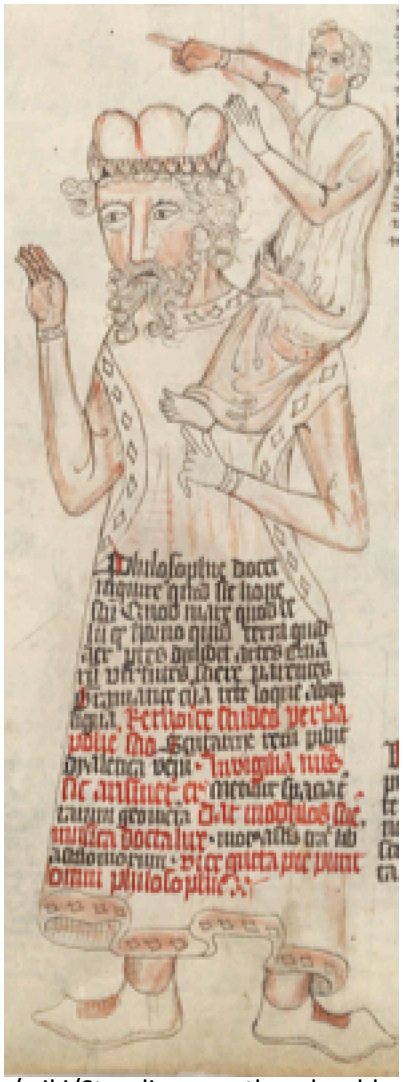
http://digitallibrary.hsp.org/index.php/Detail/Object/Show/object_id/9285

This picture is derived from Greek mythology,
where the blind giant Orion, carried his servant Cedalion on his shoulders.

"discovering truth by building on previous discoveries".

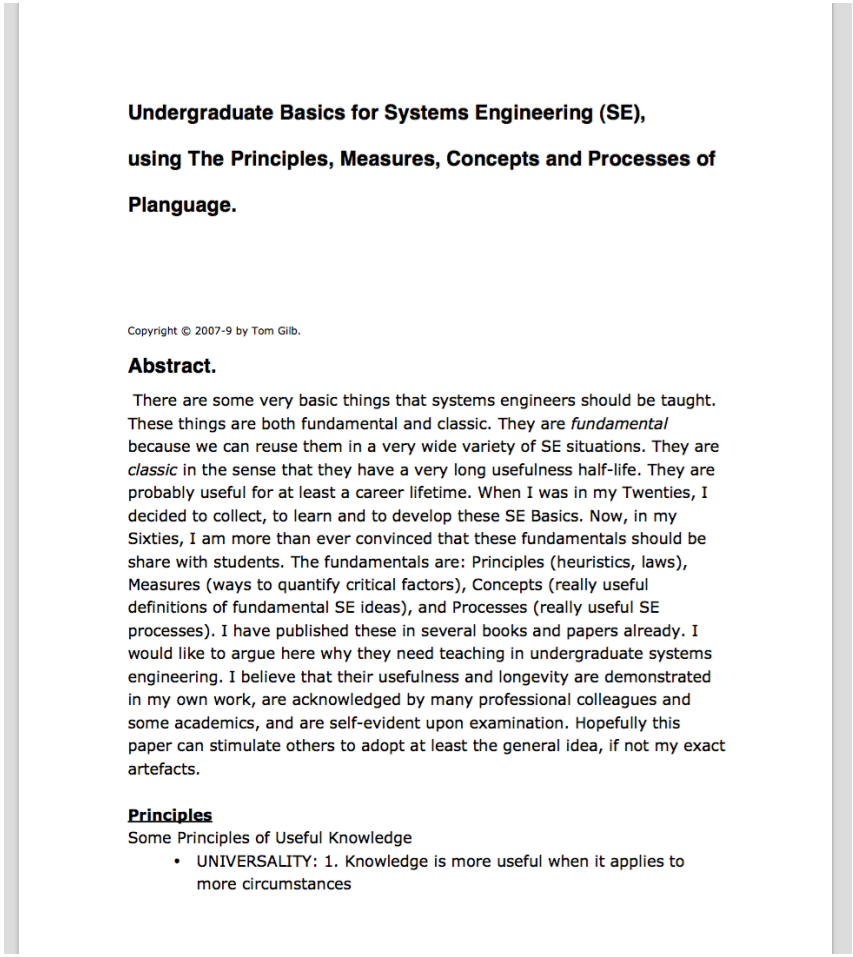
While it can be traced to at least the 12th century, attributed to [Bernard of Chartres]

its most familiar expression in English is found in a 1676 letter of Isaac Newton



http://en.wikipedia.org/wiki/Standing_on_the_shoulders_of_giants#cite_note-9

Undergraduate Basics for Systems Engineering (SE), using The Principles, Measures, Concepts and Processes of Planguage.



Undergraduate Basics for Systems Engineering (SE),

using The Principles, Measures, Concepts and Processes of

Planguage.

Abstract.

There are some very basic things that systems engineers should be taught. These things are both fundamental and classic. They are *fundamental* because we can reuse them in a very wide variety of SE situations. They are *classic* in the sense that they have a very long usefulness half-life. They are probably useful for at least a career lifetime. When I was in my Twenties, I decided to collect, to learn and to develop these SE Basics. Now, in my Sixties, I am more than ever convinced that these fundamentals should be share with students. The fundamentals are: Principles (heuristics, laws), Measures (ways to quantify critical factors), Concepts (really useful definitions of fundamental SE ideas), and Processes (really useful SE processes). I have published these in several books and papers already. I would like to argue here why they need teaching in undergraduate systems engineering. I believe that their usefulness and longevity are demonstrated in my own work, are acknowledged by many professional colleagues and some academics, and are self-evident upon examination. Hopefully this paper can stimulate others to adopt at least the general idea, if not my exact artefacts.

Principles

Some Principles of Useful Knowledge

- UNIVERSALITY: 1. Knowledge is more useful when it applies to more circumstances

- www.gilb.com/tiki-download_file.php?fileId=98

- Held INCOSE, San Diego, June 2007

- Written Originally for NTNU/Sintef Professors

Some Principles of Useful Knowledge
and also
Some measurable attributes of Knowledge

UNIVERSALITY: 1. Knowledge is more useful when it applies to more circumstances

ETERNALITY: 2. Knowledge is more worth learning if it can be applied for a long time after learning it

VALUE: 3. Knowledge is more useful if there is a high value from applying it

SHARING: 4. Knowledge is more useful if it can easily be shared with others

PROOF: 5. Knowledge is useful when early feedback can prove its usefulness in practice

SYNCHRONOUS: 6. Knowledge is more useful when it can be used together with a larger body of knowledge

MEASURABIILITY: 7. Knowledge is more useful when the results of its application can be measured

ACCEPTANCE: 8. Knowledge is more useful when it is widely accepted in your culture.

COST: 9. Knowledge is more useful when the cost of applying it is low.

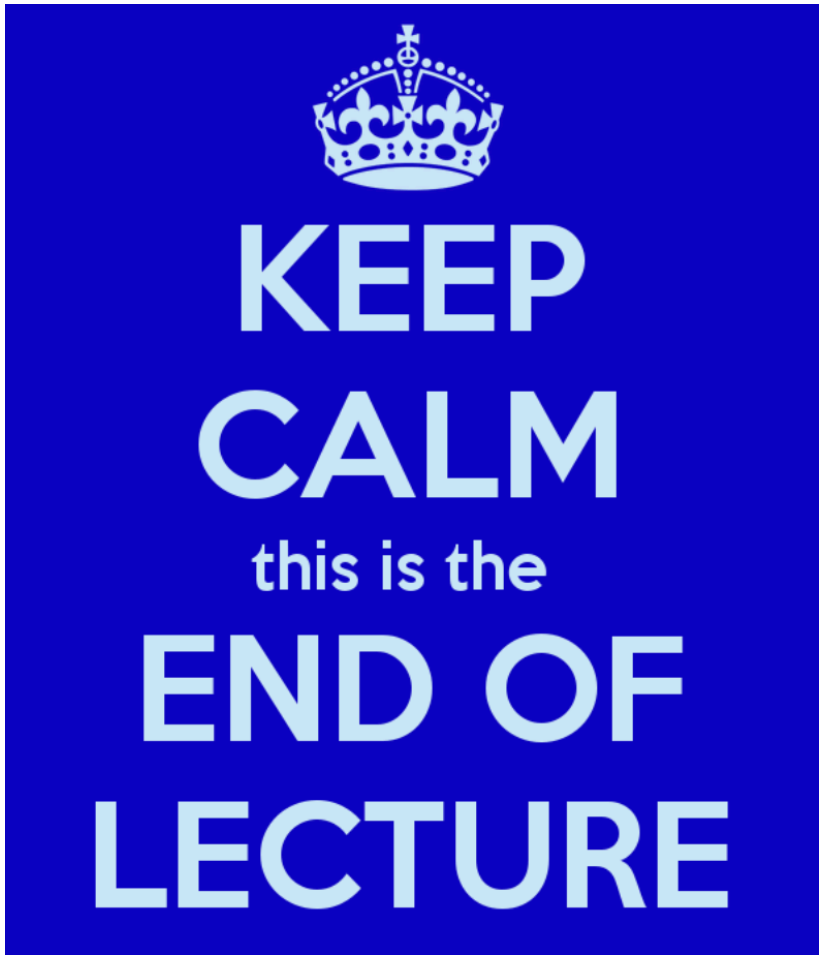
GENERATION: 10. Knowledge is more useful when is can be used to generate even more useful knowledge.

End of Lecture

- Well,
- If there is more time left, I have some more ideas to share

- If not you might like to study my extra slides at
- Gilb.com resources slide downloads

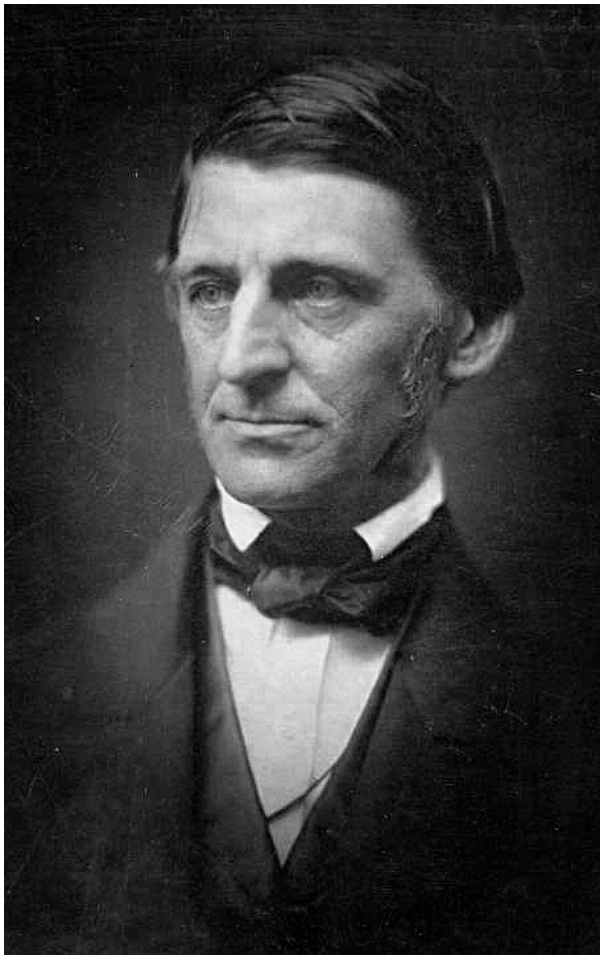
- Just 36 more slides, at 30 seconds each



The Principle that Principles beat methods

- "As to methods, there may be a million and then some, but principles are few.
- The man who grasps **principles** can successfully select his own *methods*".
- - Ralph Waldo **Emerson,**

  1803-1882, USA



The Notion of **Usefulness** of Principles:

- A ***principle*** is
  - a short statement that guides people
  - to take certain decisions or action.
- It is 'condensed **wisdom'**.
  - Wisdom is a class of knowledge.
- Principles are **useful** if
  - they remind or teach us to act in a better way than we otherwise would do

- For example, 1 principle:

- **"There is lots of uncertainty and risk of deviation from plans in any project. "**

- "You cannot eliminate risk. But, you can document it, plan and design for it, accept it, measure it and reduce it to acceptable levels. You may want to avoid risk, but it doesn't want to avoid you."
  - Source: Competitive Eng. book, page 23.

- **This principle tries to warn about the inevitability of risk**
- **It also is specific about what you can do about risk.**
- **It teaches that you cannot eliminate risk, but you can try to manage it in various ways.**

- **From the departure point of this principle, the teacher can then be more specific on how to identify, specify and mitigate risks.**
  - **"Risk Management: A practical toolkit for identifying, analyzing and coping with project risks." (Gilb)**
  - **http://www.gilb.com/tiki-download_file.php?fileId=208**

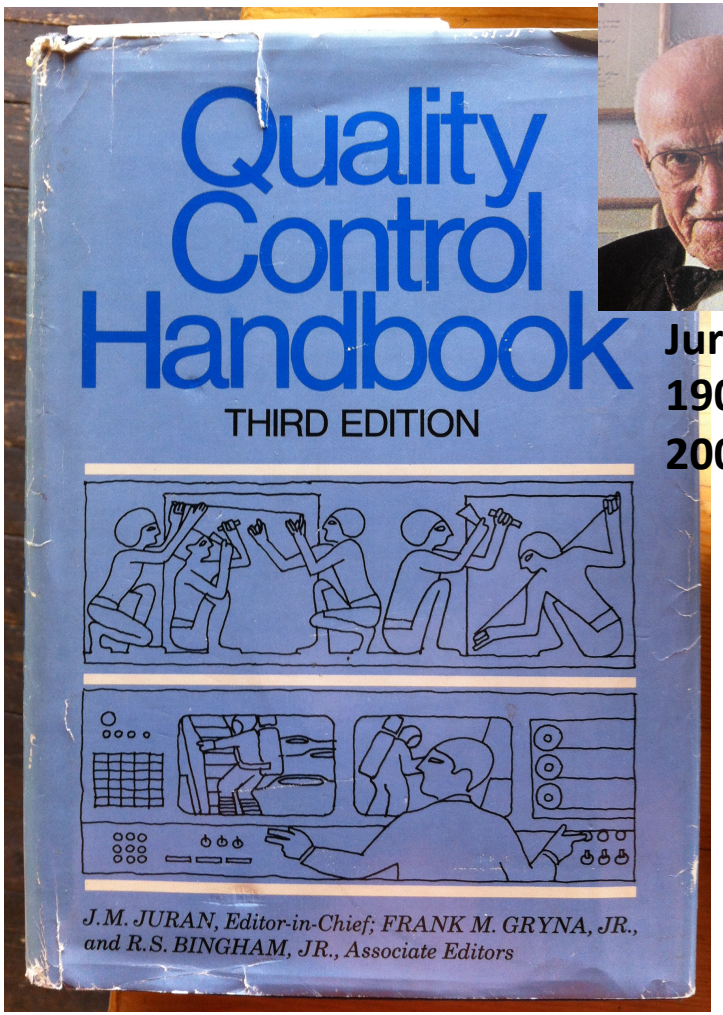
7 'da Vinci' Principles: Systems Engineering!

M. Gelb, *How to Think Like Leonardo Da Vinc*i , p.9

- Curiosità
  - –Insatiably **curious**, unrelenting quest for continuous learning
- Dimostrazione
  - –Commitment **to test knowledge through exper**ience, willingness to learn from mistakes. Learning for ones self, through practical experience
- Sensazione
  - –Continual **refinement** of senses. As means to enliven experience
- Sfumato
  - –Willingness to **embrace ambiguity**, paradox, uncertainty
- Arte/Scienza
  - –**Balance** science/art, logic & imagination, **whole brain thinking**
- Corporalità
  - –Cultivation of grace, ambidexterity, **fitness**, poise
- Connessione
  - –Recognition & appreciation for **interconnectedness** of all things and phenomena, **Systems thinking**

The Notion Of Half Life of Principles

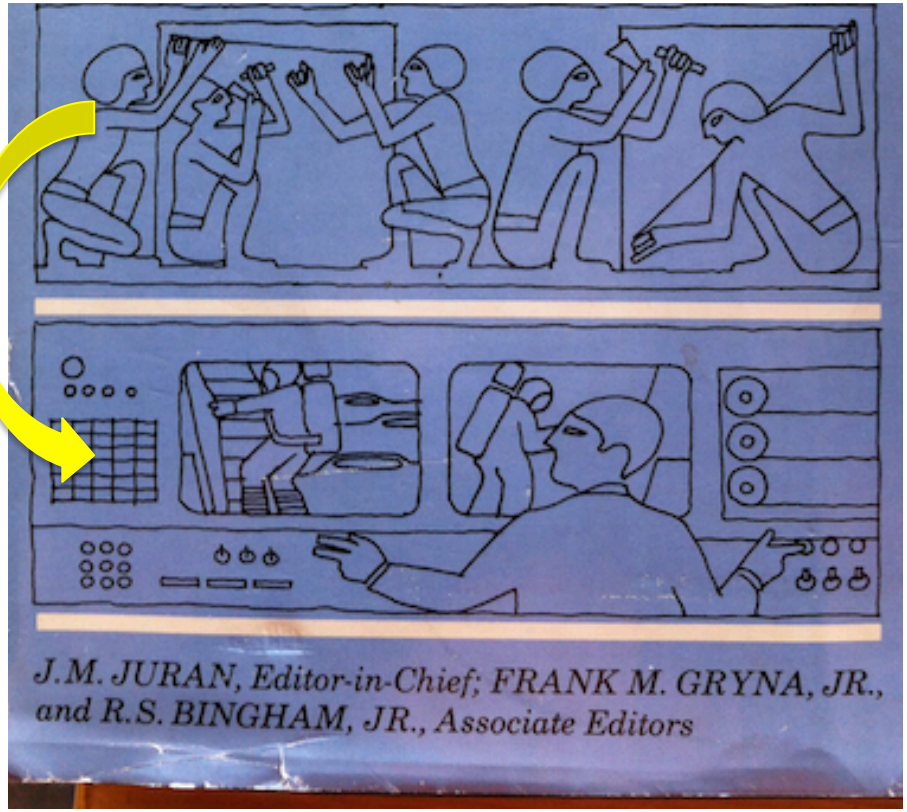
- If a principle became obsolete in a few years – perhaps because of new technology or new economics, **then it would be less valuable to learn**, and might even be dangerous to continue to practice beyond its true lifetime.
- So I prefer principles that we can imagine **'always were true'**, and we can so no clear reason why they 'will not be true for the foreseeable future".
- It takes decades from when a principle is stated, until it becomes **taught** in any substantial way.
- The student has **decades of their future in which to apply a principle.**
- So it makes good sense that the **principle is** something we **can rely on in the long term.**



Juran 1904-2008

The Principle of Quality Control Inspection in relation to a standard

Juran's QC Handbook
34 Centuries, same principle



books, and handbooks.

JACKET ILLUSTRATIONS

Ancient stone construction and modern expeditions to the moon are separated by 34 centuries of time, but are united by the timeless principles of quality control.

The Egyptian workmen dressing stone are getting data feedback from inspectors who are (left) checking the blocks with boning rods to measure the deviation from an exact plane, and (right) using a string to check a block for flatness. The pictures are from a tomb in Thebes, c. 1450 B.C. (Metropolitan Museum of Art. Reconstructed drawing from Singer et al, A History of Technology, Vol. I, p. 481, fig. 313, Oxford University Press)

The late 20th century figures depict an inspection station of the National Aeronautics and Space Administration using incredibly complex and precise instruments to monitor numerous activities of men and machines conducting a mission far from planet Earth. The drawing is by Dan Nevins.

Jacket design by Dan Nevins.

My 'Principle' Concerns

- In 'Competitive Engineering' I have offered 100 such principles.
  - http://www.gilb.com/dl352 (← a collection of principles)
- I have 'brain-stormed' many more, in other books and papers.
- I am sure my many systems engineering, and other disciplines, colleagues, have, and will continue to develop, **principles** that deserve to be taught formally.

- My **concern** is that *we place far too little emphasis on selecting and teaching these principles*.
- My **concern** is that *students do not even get a dozen good principles* to base their professional work on.

- I think we need a *course,*
  - called something like **"The Most Important Systems Engineering Principles".**

The Notion of **Fundamentality** of Principles "Principles that apply to *everything*"

- Principles should be *fundamental*.
  - They should **be basic tools for everyday use** in planning, engineering, discussing, decision-making, and reasoning.
  - We should be able to use them as the basis for all our more-detailed actions and thinking processes.
- For example:
- "The Principle of
- '**Quality Early**: Quality In, From the Beginning':
- **Quality needs to be *designed into* processes and products.**
  Cleaning up bad work is a loser, but cleaning early is better than late.
- *A stitch in time still saves nine,*
  But an ounce of prevention is still worth a pound of cure. "
  - Source Competitive Engineering (2005), page 24.

- This *'Quality Early'* **principle applies to all engineering and management planning work.**
  - We humans seems to have a strong natural tendency to clean up our faulty work when faults are discovered, rather than to consciously discover how we can prevent the faults from getting into our work in the first place.
- This principle is at the heart of CMMI Level 5 (Defect Prevention). And Deming PDSA/SPC, and 'Lean'
- This principle is **fundamental**.
  - It is at the basis of all improvement efforts in a systems engineering process.
  - It is the basis for a paradigm shift for many professionals I deal with; the shift
    - from 'fix problems', to 'prevent problems'.
- Students should be taught such profound principles before they waste years discovering them, if ever.

Measures

- *One single experience overshadows all others in my technological wanderings*.
- ***Engineers do not seem to have been taught how to quantify most of the critical quality aspects*** of their systems.
  - Like: productivity, usability, security
- Most real engineers have been taught how to deal with qualities
  - like availability and reliability.
- But these are **just two** of *hundreds* of quality aspects,
  - we meet when engineering systems.

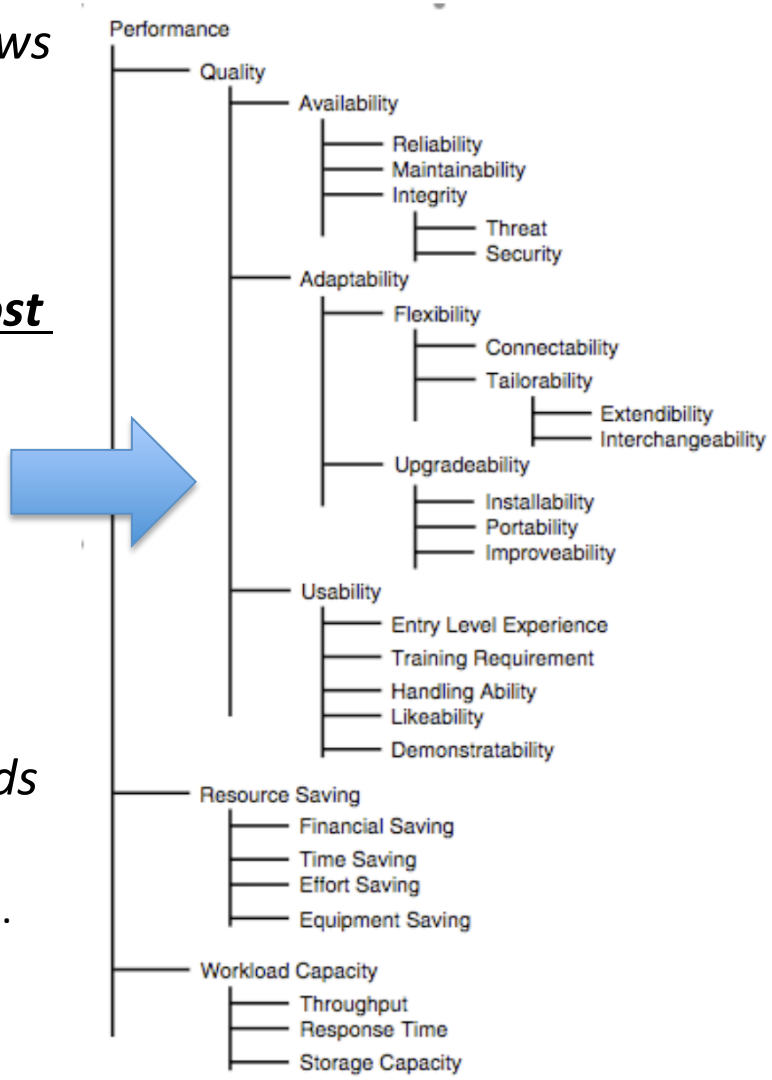


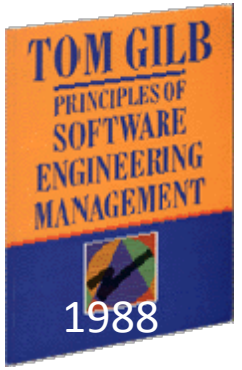
Illustration: Some quality concepts and their possible decompositions. Source CE, page 154.

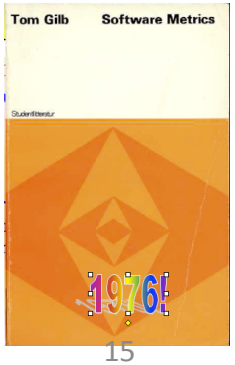
The Changing face of Systems Engineering (= more quality metrics)

- Dr. Hastings of MIT, in describing
  - *the changing face of systems engineering,*
- spoke of conventional SE (2005) with
- *"Focus on reliability, maintainability, and availability"*
- *and referred to 'Expanded SE' as having an*
- *"Emphasis on **expanded set of 'ilities'** and*
- ***designing in** robustness, flexibility, adaptability in concept phase".*
- "The incorporation of system properties, such as **sustainability, safety and flexibility** in the design process. (These **are lifecycle properties** rather than first use properties.)" (2004)

https://esd.mit.edu/symposium/pdfs/monograph/future.pdf
2004 version

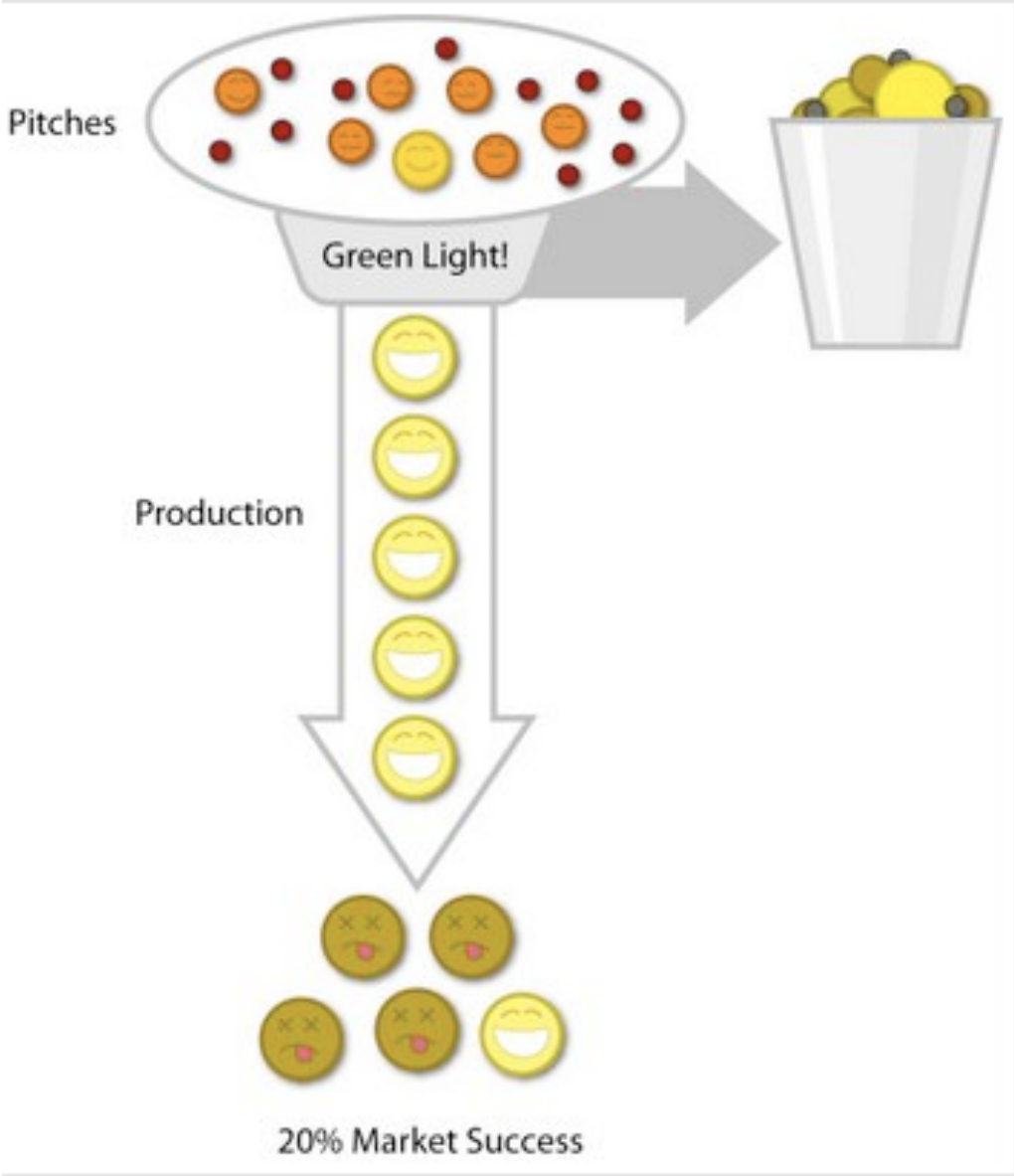
- I agree.
- But we **are not being trained** to do so.
- The textbook **literature** is extremely **sparse** on the subject.
- **Most** all professional engineers I meet have **never seen this done** in an engineering manner, by defining the system requirements quantitatively.
- It is not sufficient to state **slogans** ('we need more robustness") and then throw in all the robustness technology we can think of at the moment.
  - But that is an good description (management BS, no action) of what I see done in practice.
- The problem is that we **do not even teach** basic patterns of defining these ilities measurably.





‘Horror’ Project.

Requirements Case



Based On Real Case of mine 2006-8

Summary of Top '8' Project Objectives

Real Example of ***Lack*** of Quantification in large Engineering Company Project

*1. Central to The Corporations business strategy is to be the world's **premier** integrated_<domain> service **provider**.*

*2. Will provide a much more efficient **user** experience*

*3. Dramatically scale back the **time** frequently needed after the last data is acquired to time align, depth correct, splice, merge, recompute and/or do whatever else is needed to **generate** the desired **products***

*4. Make the system much **easier** to **understand** and **use** than has been the case for previous system.*

*5. A primary goal is to provide a much more **productive** system **development** environment than was previously the case.*

*6. Will provide a richer set of functionality for **supporting** next-generation logging **tools** and applications.*

*7. **Robustness** is an essential system requirement*

*8. Major improvements in **data quality** over current practices*

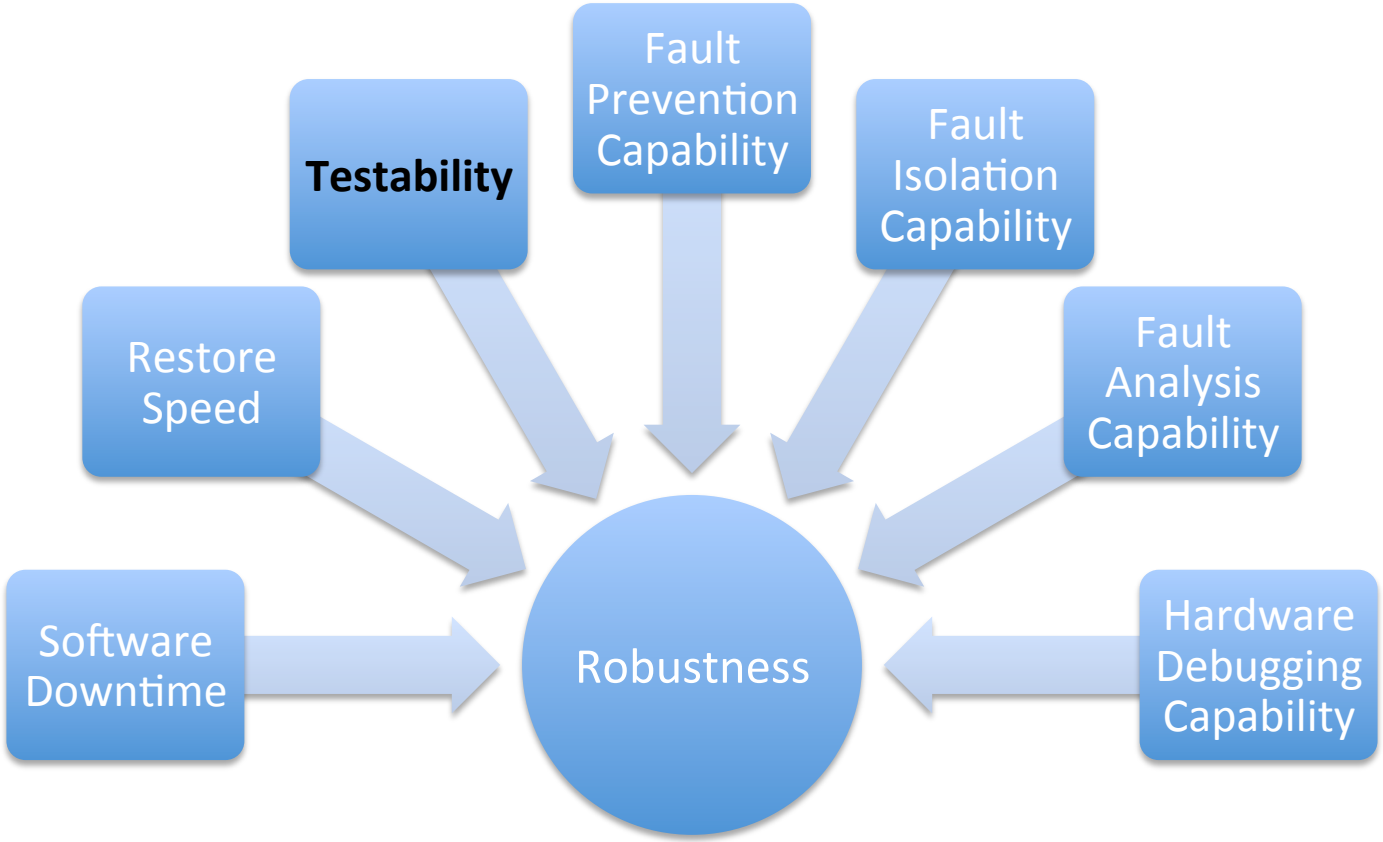
This lack of clarity cost them over $100,000, 000.
and 8 years delay

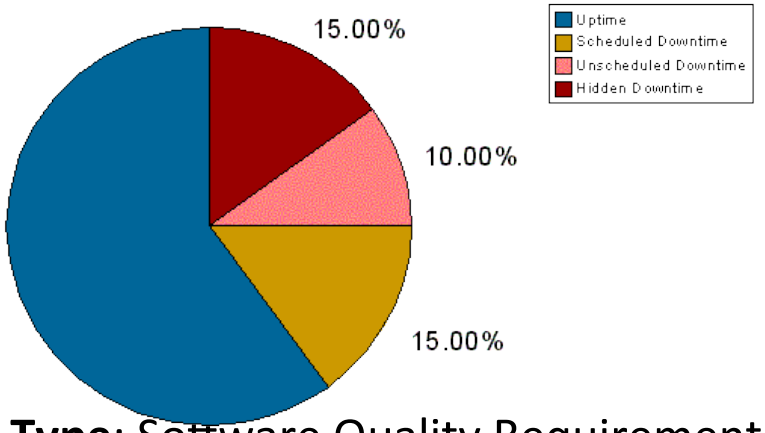
Rock Solid Robustness: *many splendored*

- **Type**: *Complex* Product Quality Requirement.
- **Includes**:
  - {*Software Downtime*,
  - *Restore Speed*,
  - *Testability*,
  - *Fault Prevention Capability*,
  - *Fault Isolation Capability*,
  - *Fault Analysis Capability*,
  - *Hardware Debugging Capability*}.

-

A Complex Requirement "Robustness"





Software Downtime:

Type: Software Quality Requirement. **Version**: 25 October 2007.

Part of: Rock Solid Robustness.

Ambition: to have minimal downtime due to software failures <- HFA 6.1

Issue: does this not imply that there is a system wide downtime requirement?

Scale: <mean time between forced restarts for defined [Activity], for a defined [Intensity].>

Fail [Any Release or Evo Step, Activity = Recompute, Intensity = Peak Level] 14 days <- HFA 6.1.1

Goal [By 2008?, Activity = Data Acquisition, Intensity = Lowest level] : 300 days ??

Stretch: 600 days.

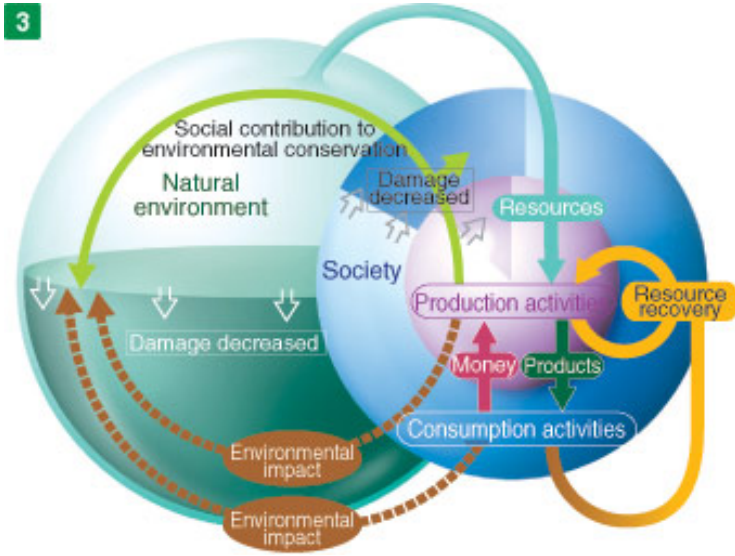
Restore Speed:

Type: Software Quality Requirement. **Version**: 25 October 2007.

Part of: Rock Solid Robustness

Ambition: Should an error occur (or the user otherwise desire to do so), the system shall be able to restore the system to a previously saved state in less than 10 minutes. <-6.1.2 HFA.

3



Scale: Duration from Initiation of Restore to Complete and verified state of a defined [Previous: Default = Immediately Previous]] saved state.

Initiation: defined as {Operator Initiation, System Initiation, ?}. Default = Any.

Goal [Initial and all subsequent released and Evo steps] 1 minute?

Fail [Initial and all subsequent released and Evo steps] 10 minutes. <- 6.1.2 HFA

Catastrophe: 100 minutes.

Testability (part of "Robustness")

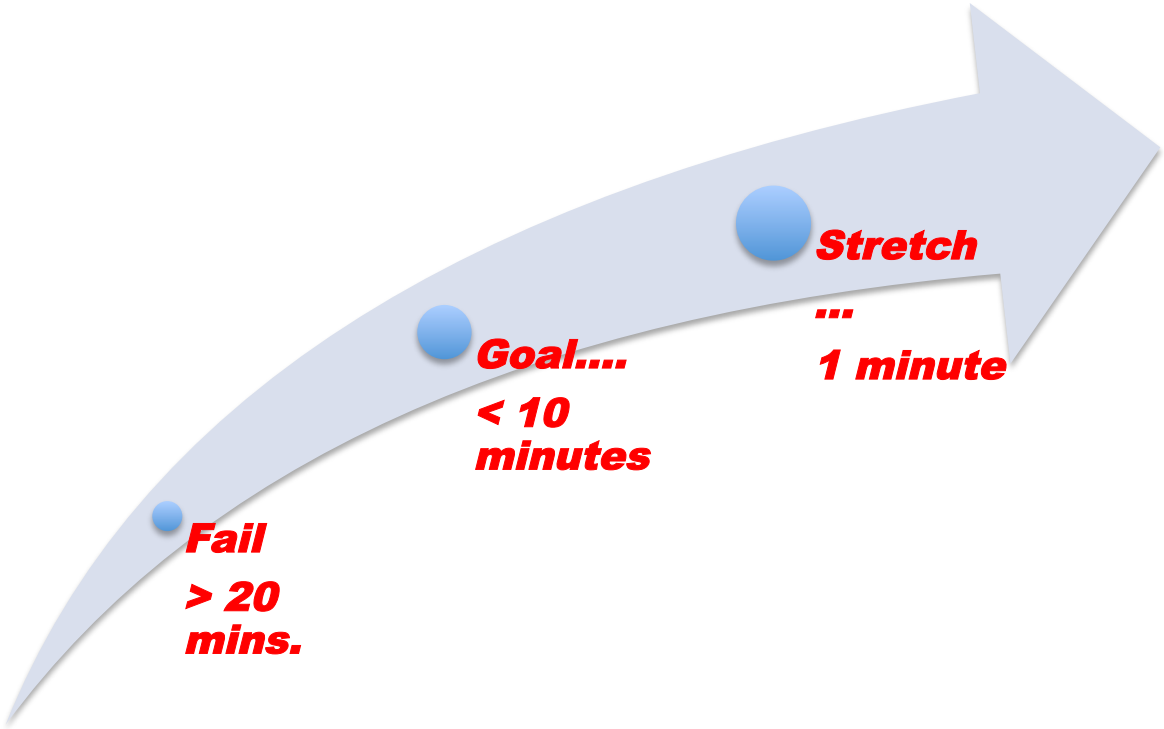
Scale: the *duration* of

a defined [Volume] of testing,

of a defined [Type],

by a defined [Skill Level] of system operator,

under defined [Operating Conditions]



Testability:

Type: Software Quality Requirement.

Part of: Rock Solid Robustness

Initial Version: 20 Oct 2006

Version: 25 October 2007.

Status: Demo draft,

Stakeholder: {Operator, Tester}.

Ambition: Rapid-duration automatic testing of <critical complex tests>, with extreme operator setup and initiation.

Scale: the duration of a defined [Volume] of testing, or a defined [Type], by a defined [Skill Level] of system operator, under defined [Operating Conditions].

Goal [All Customer Use, Volume = 1,000,000 data items, Type = WireXXXX Vs DXX, Skill = First Time Novice, Operating Conditions = Field, {Sea Or Desert}. <10 mins.

Design Hypothesis: *Tool Simulators, Reverse Cracking Tool, Generation of simulated telemetry frames entirely in software, Application specific sophistication, for drilling – recorded mode simulation by playing back the dump file, Application test harness console <-6.2.1 HFA*

'**Impact Estimation Table**' (simple, real UK case): an objective* Knowledge Store and Reflector

Man-Chie Tse1,2 & Ravinder Singh Kahlon 1,2
{Man-Chie, Ravi}@dkode.co

HEALTHCARE SYSTEM IMPACT ESTIMATION

	Automate Rules	Web Self Service	Decision Support	Total Impacts
Increase Transmission of Requests *(30 minutes → 10 minutes)*	10 minutes 100%	3 minutes 100%	-	200%
Decrease Number of Errors Occurring *(353 per week → 30 per week)*	100 errors 80%	< 50 90%	-	170%
Decrease Time for Processing of Requests *(70 minutes → 10 minutes)*	35 minutes 70%	-	< 10 minutes 90%	160%
Decrease Time to Learn process *(1 day → 1 hour)*	-	1 hour 100%	10 minutes 103%	203%
TOTAL DESIGN REQUIREMENT IMPACT	250%	290%	193%	

* Requirements are objectively measurable, costs are objectively measurable.
Impacts are of defined objectivity based on documented evidence, documented sources,
And ± uncertainty ranges. See Gilb.com and Competitive Engineering, or tinyurl.com/valueplanning
for more detail on Impact Estimation method. See Brodie PhD 2015, Middlesex University.

Quality Function Deployment QFD for Comparison; A BAD Knowledge Store
due to lack of metrics in requirements and in design impacts, and lack of clear concepts
Much less well defined and less objective quantification than Impact Estimation
See Paper written by me for Kongsberg Students **"How problems with Quality Function Deployment's (QFD's) House of Quality (HoQ) can be addressed by applying some concepts of Impact Estimation (IE)** " http://www.gilb.com/tiki-download_file.php?fileId=119

Stakeholder Need

How 'Reliable'?

How 'Strong' is Strong?

What is this "importance' rating?

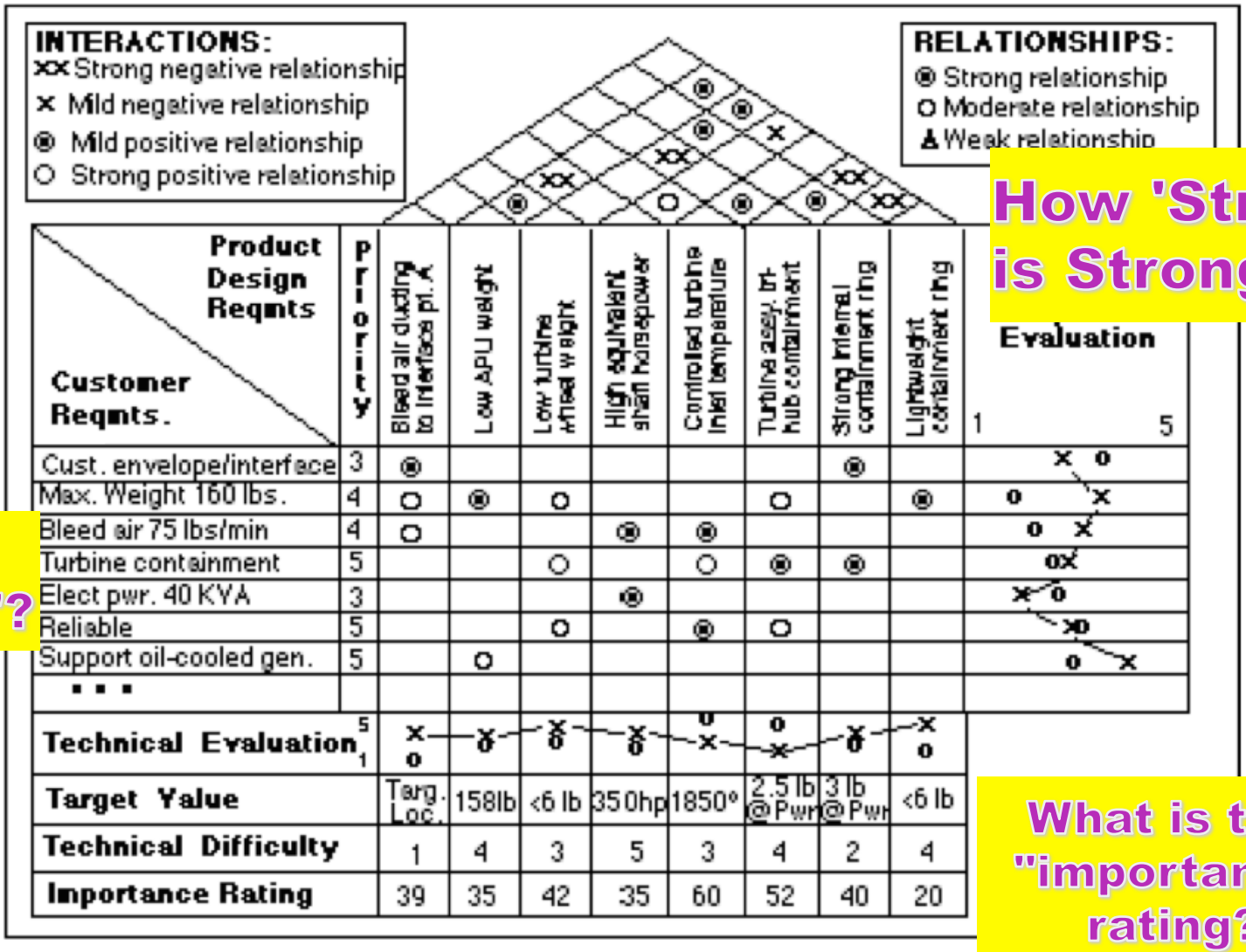
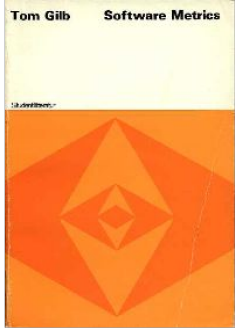


Figure 4 QFD House of Quality



Why are metrics Important in Systems Engineering?

• **Simplify requirements** (if the top few requirements are quantified, there is less need for copious documentation as the developers are focused on a clearer, simpler 'message'); http://www.gilb.com/dl554

• **Communicate quality goals much bet**ter to all parties (that is, users, customers, project management, developers, testers, and lawyers);

• **Contract for results**. Pay for results only (not effort expended). Reward teams for results achieved. This is possible as success is now measurable;

• **Motivate** technical people to focus on real business results;

• **Evaluate** solutions/designs/architectures against the quantified quality requirements;

• **Measure evolutionary project progress** towards quality goals and get early & continuous improved estimates for time to completion;

• Collect numeric historical data about designs, processes, organizational structures for future use.

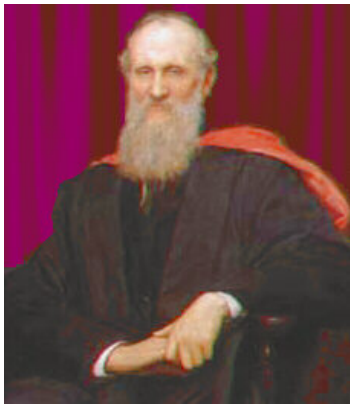
Use the data to obtain an **understanding of your process efficiency**, to bid for funding for improvements and to benchmark against similar organizations!

- Simplify requirements
- Communicate Quality
- Result Contracts
- Motivation
- Evaluation
- Tracking
- Process Management

The Principle Of 'Quality Quantification'
The Words of a 'Lord'

"All qualities can be expressed quantitatively, *'qualitative'* does *not* mean unmeasurable". (Gilb)

http://tinyurl.com/**GilbTedx**



"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, *Lecture to the Institution of Civil Engineers, 3 May 1883* **From http://zapatopi.net/kelvin/quotes.html**

Born: 26 June 1824; Belfast, Ireland
Died 1907..

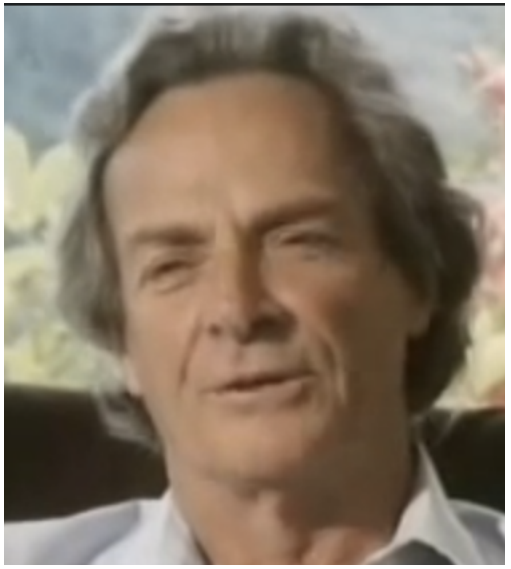
Conclusion
‘Metrics are Basic Knowledge Tools’

- I think that learning to *quantify*, and *measure*
  - ‘E v e r y t h i n g’ (variable) that is
  - and ‘CRITICAL’
    - TO YOUR PROJECT OR SYSTEM
    - All values, qualities, costs
  - Is *fundamental* to systems **engineering** studies

‘Concepts’ as knowledge tools

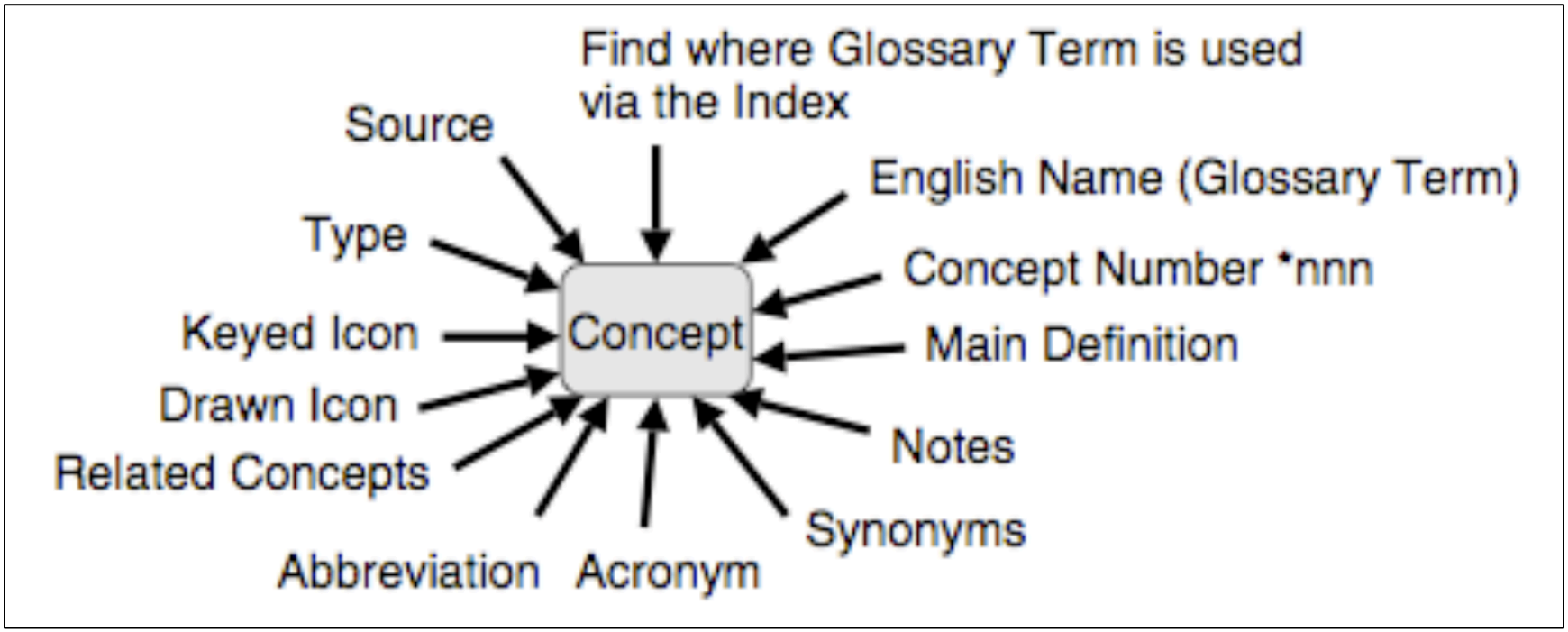
‘What’s The Name of The Game’ (ABBA)
Names are not ‘knowledge’

- “You can know the name of that bird in all the languages of the world,
  - but when you’re finished,
  - you’ll **know absolutely nothing whatever** about the bird.
- You’ll only know about humans in different places, and what they call the bird.
- So let’s look at the bird and *see what it’s doing*—that’s what counts.”
- I learned very early
  - the difference between knowing the name of something
  - and *knowing* something.”
  - *Richard Feynman*



http://www.haveabit.com/feynman/2

Concepts



Concepts:
My primary concerns are that

1. *we do **not** have a **rich enough** set of concepts: we need to distinguish between many types of requirements, many types of designs, many types of constraints – and much more.*

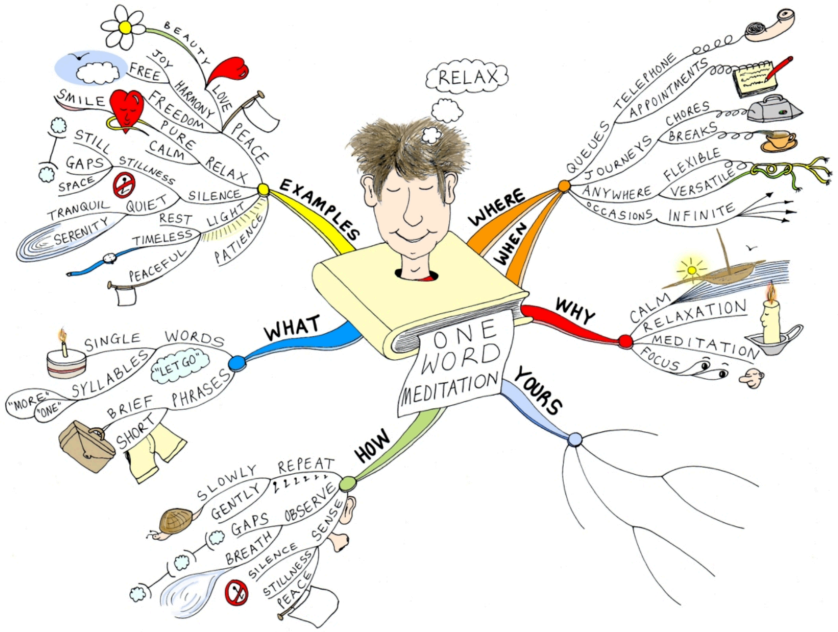
*2. We use words with **no agreed meaning**, as though others would know what we mean*

3. Our concepts are not specifically aligned with their particular specialized use; their 'unique concept',

like the concept of 'requirement' →

We use generic words (like 'requirement') for a variety of different concepts, like

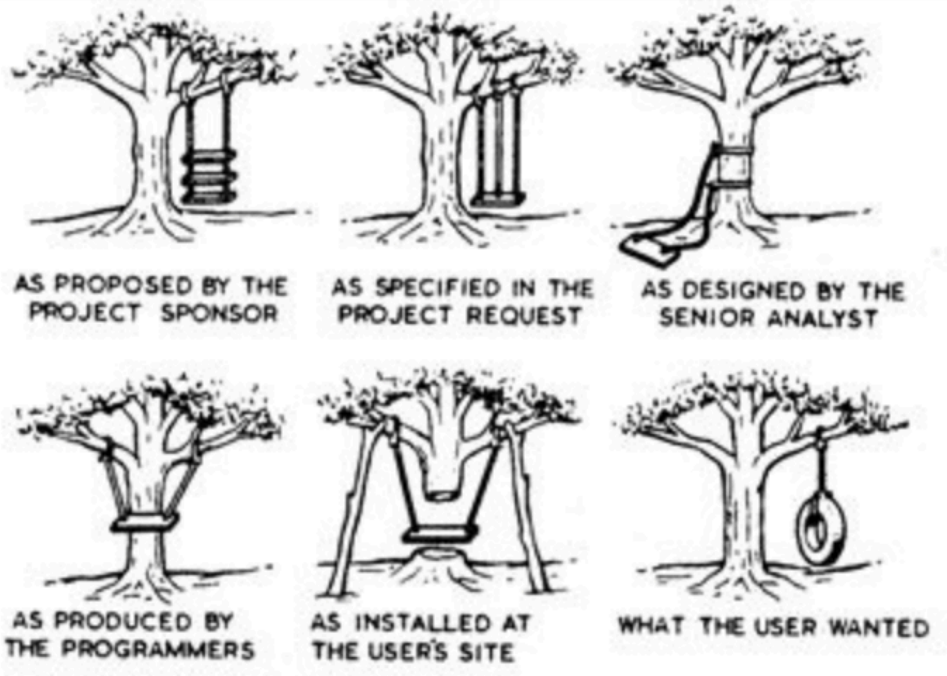
- *Technical design*
- *Architecture*
- *Need*
- *Value*
- *Constraint*
- *Management Long Range Objective*



Question

- Define, 30 seconds)
- *'REQUIREMENT'*
- **Tell person next to you**
- **A requirement is (IMHO) :**

  _____



"**Requirement**" is (IMHO)

"Stakeholder-Valued

System State,

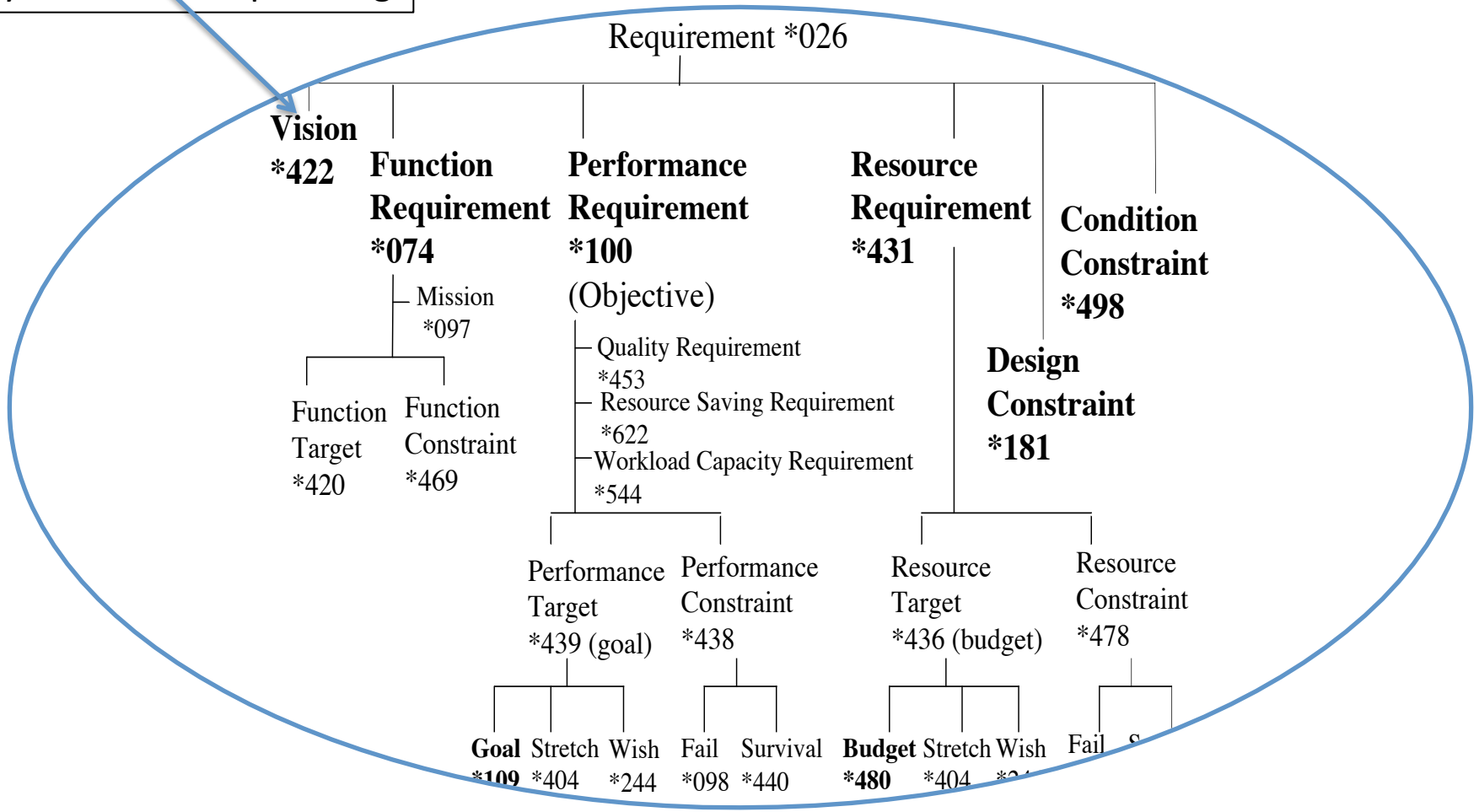
under stated conditions"

Source: Gilb, Planguage Concept Glossary September 4 2012 version

http://www.gilb.com/tiki-download_file.php?fileId=386

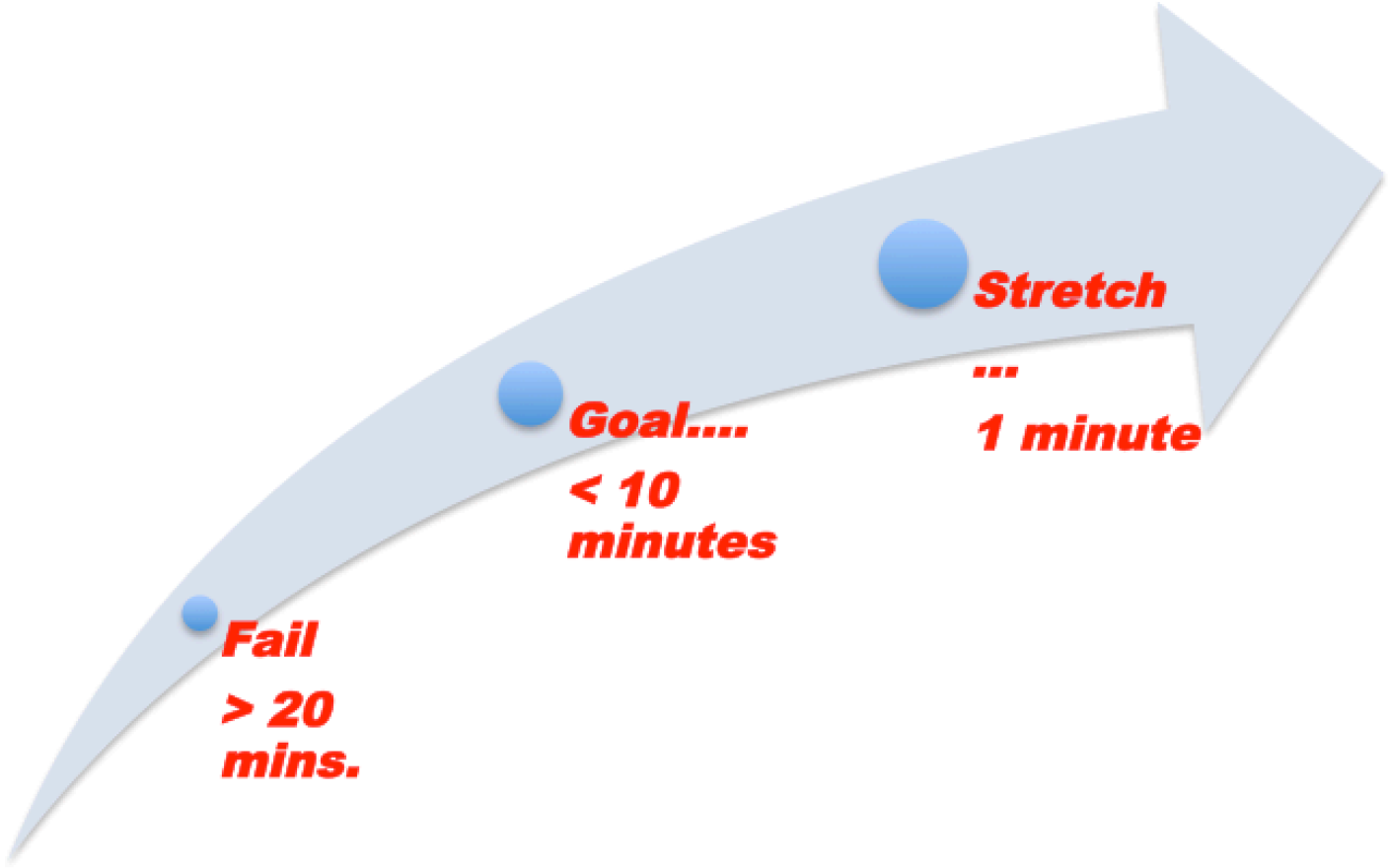
There are Many Basic Requirement Types (*as defined in Planguage*)

See 'Vision Engineering' in Tinyurl.com/valueplanning



What is a ‘Goal ?

(1 of these 3 types of requirements’



Goal Concept *109
IMHO, Planguage Concept

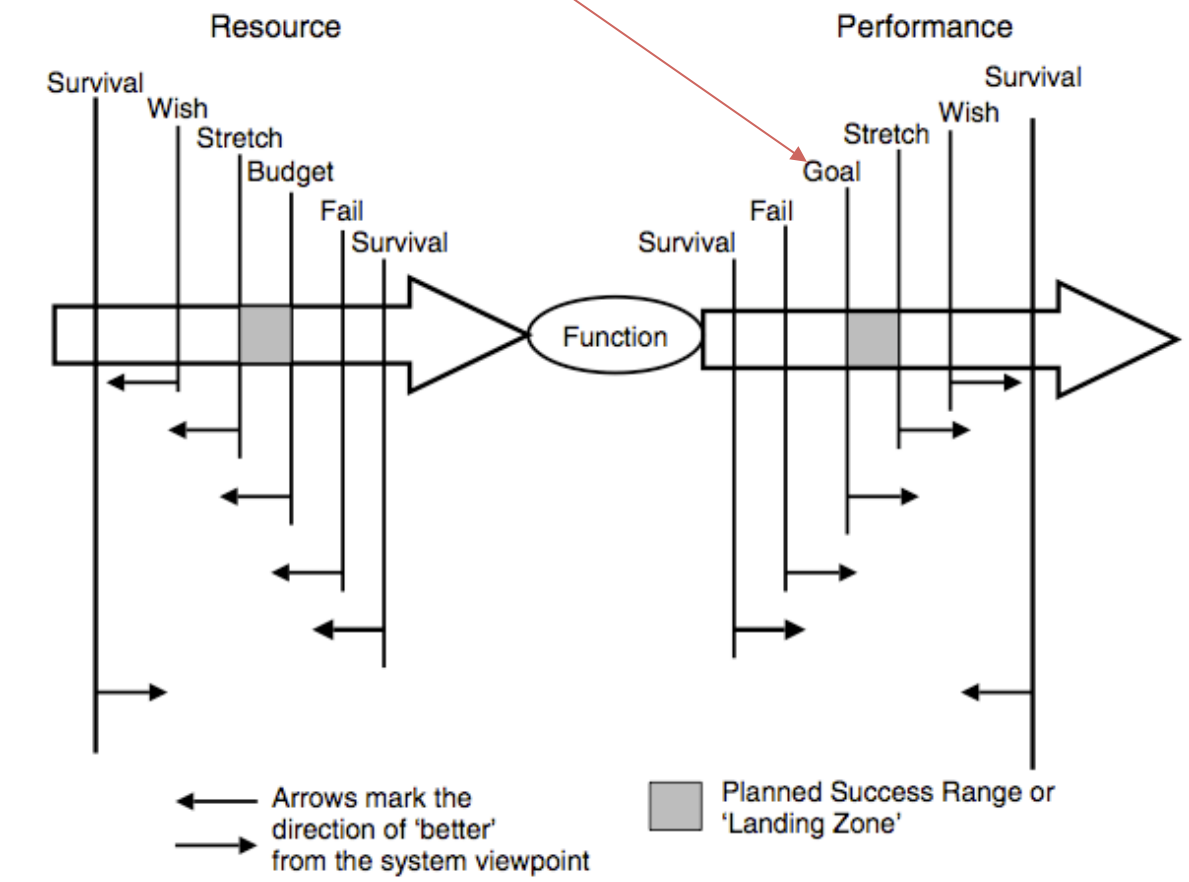
- **Goal** Concept *109
- A goal is a primary numeric **target level** of *performance*.
- An implication of a Goal specification is that there is, or will be, a **commitment** to deliver the Goal level (something not true of a Stretch or Wish target specification).
- Any commitment is **based on a trade-off pro**cess, against other targets, and considering any constraints.
- The specified Goal level may need to go through a **series of changes**, as circumstances alter and are taken into consideration.
- A specified Goal level will **reasonably satisfy** *stakeholders*.
- Going *beyond* the goal, at the cost of additional resources, is not considered necessary or profitable – even though it may have some value to do so for some stakeholders.

Conditions for a **'Goal'** level

When is a **G**oal level really a valid **G**oal? <-CE 366, *109

1. Technically **possible** - within state of art
2. **Economically** Possible - resources exist
3. **Costs** consistent with other Requirements
4. **Effective**, and effect necessary to satisfy stakeholder needs
5. **Profitable**: value over cost
6. **Prioritized**: by any rules of priority
    1. Effectiveness
    2. Profitability
    3. Politics
7. All [**Conditions**] in the Goal statement are 'true'

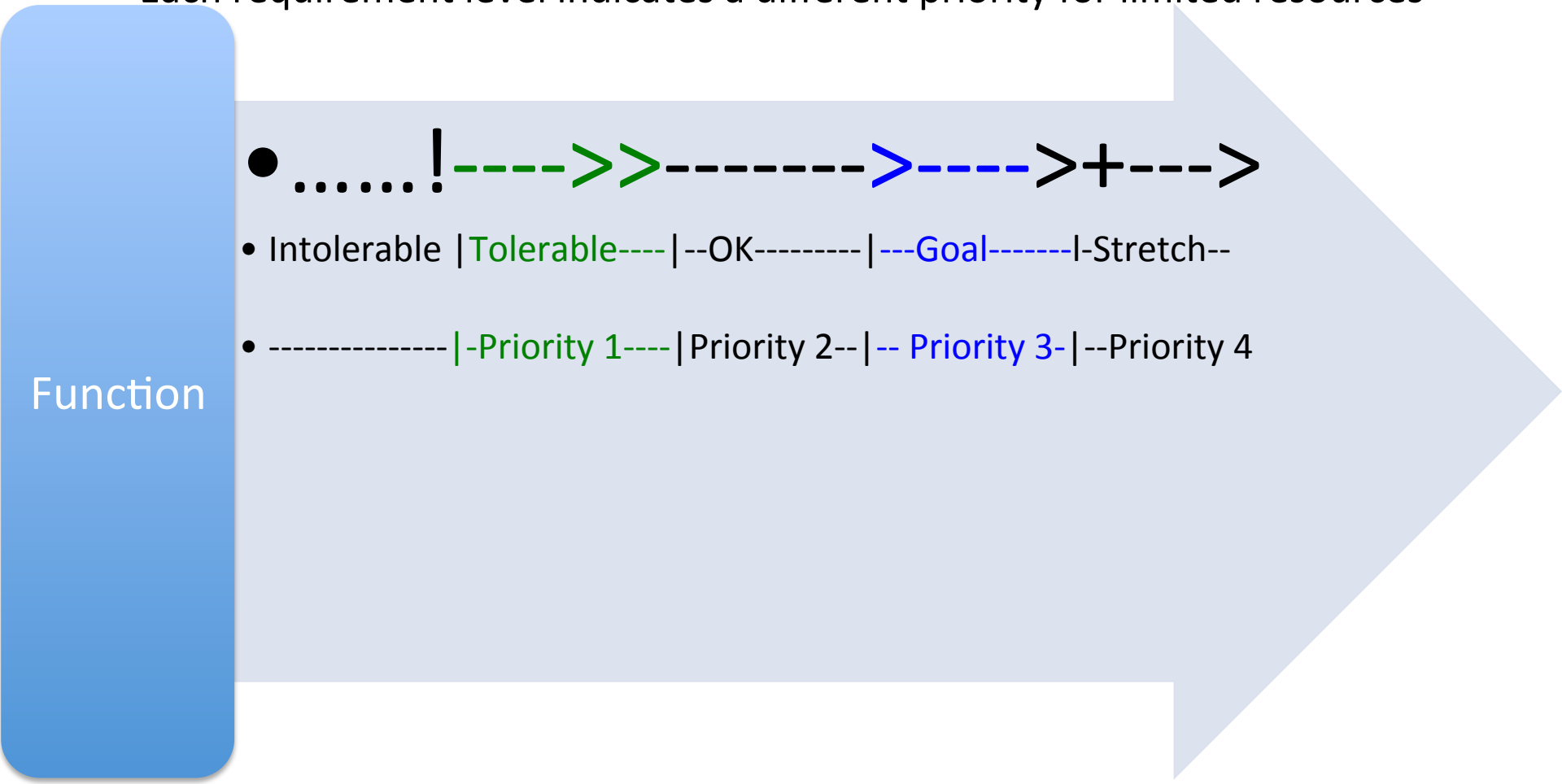
All points above must be satisfied!
For a Goal statement to be 'activated'



Source: 'Competitive Engineering' 2005

PRIORITY RANGES

Each requirement level indicates a different priority for limited resources



……!---->>-------->---->+--->
these symbols are Planguage-defined keyed icons
For levels of performance, in CE book

COMFORT RANGES

A more-popular view of the priority levels

O……Catastrophe|Tol.----------Fail|OK--------|Goal----------|Stretch---→

Dead |Alive | Good | Success | Incredible

— Confucius, *Sayings of Confucius*

"True wisdom is knowing what you don't know"

— Confucius, *Sayings of Confucius*

Da Vinci on *Experience* as source of useful knowledge

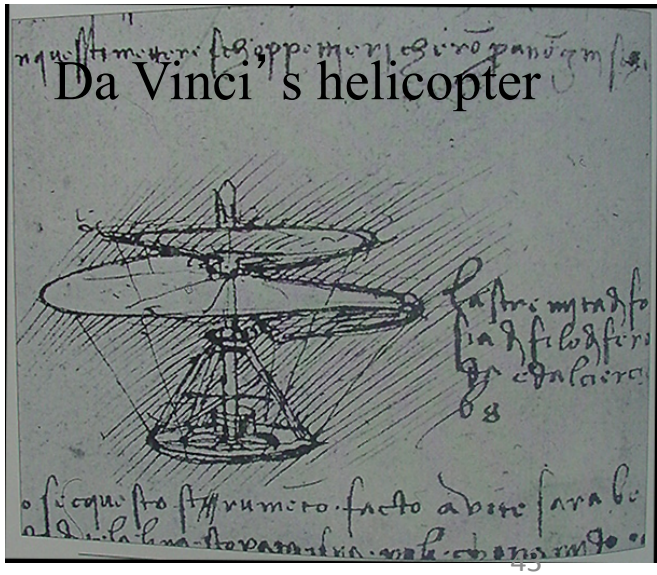
- Leonardo, proudly described himself as:
  - **Uomo senza lettre** (man without letters)
  - **Discepolo delle esperienza** (disciple of experience)
- "To me it seems that those sciences are in vain and full of error
  - which are not born of experience,
    - mother of all certainty,
  - *first hand* experience
    - which in its origins, or means, or end has passed through one of the *five senses*."
  - Source: Gelb page 78

Leonardo's persistence

- "Although generally recognized as the greatest genius of all time, Leonardo made many colossal mistakes and staggering blunders." <-Gelb

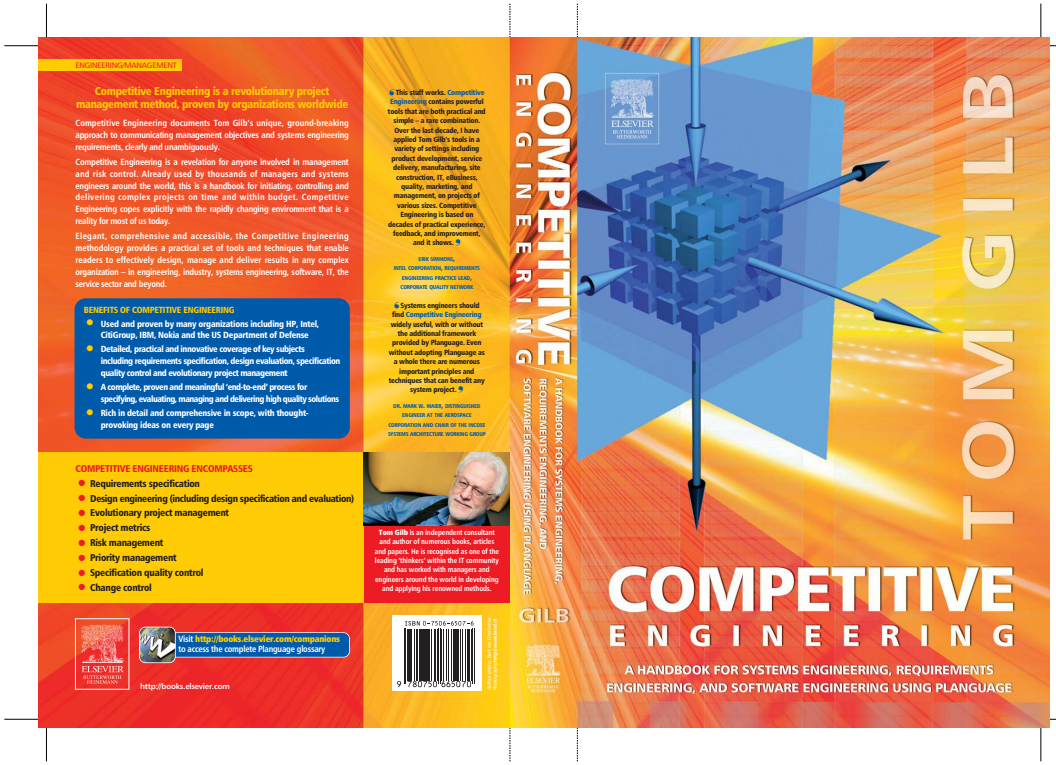
- "Despite mistakes, disasters, failures, and disappointments, Leonardo never stopped learning, exploring, and experimenting. He demonstrated Herculean persistence in his quest for **knowledge**." <- Gelb

- Leonardo wrote: <-Gelb p.79
  - *"I do not depart from my furrow.*
  - *"Obstacles do not bend me"*
  - *"Every obstacle is destroyed through rigor"*



‘Competitive Engineering’ (2005): a handbook of knowledge

- Your CE Book free pdf:
  - http://www.gilb.com/dl541
- **Over 100 Principles**
  - http://www.gilb.com/dl352.php?fileId=352
- **Over 100 Metrics**
  - **Chapter 5: Scales of Measure:**
  - http://www.gilb.com/tiki-download_file.php?fileId=26
- **Over 700 Defined Concepts**
  - http://www.gilb.com/tiki-download_file.php?fileId=387
    - **Book Glossary**
  - http://www.gilb.com/tiki-download_file.php?fileId=46
    - **Full Glossary**



That's All Folks !

- Gilb.com

- Tom@Gilb.com
- @ImTomGilb
- +47 920 66 705 Cell