Value Management

For Live presentation (recorded)
Wednesday 13th May 2020, 18:00 to 20:00 UK +

Video URL= https://www.youtube.com/playlist?list=PLKBhokJ0qd3_wlvr0j85YhmNfNj8ZJ8M-
(BCS SPA Collection)

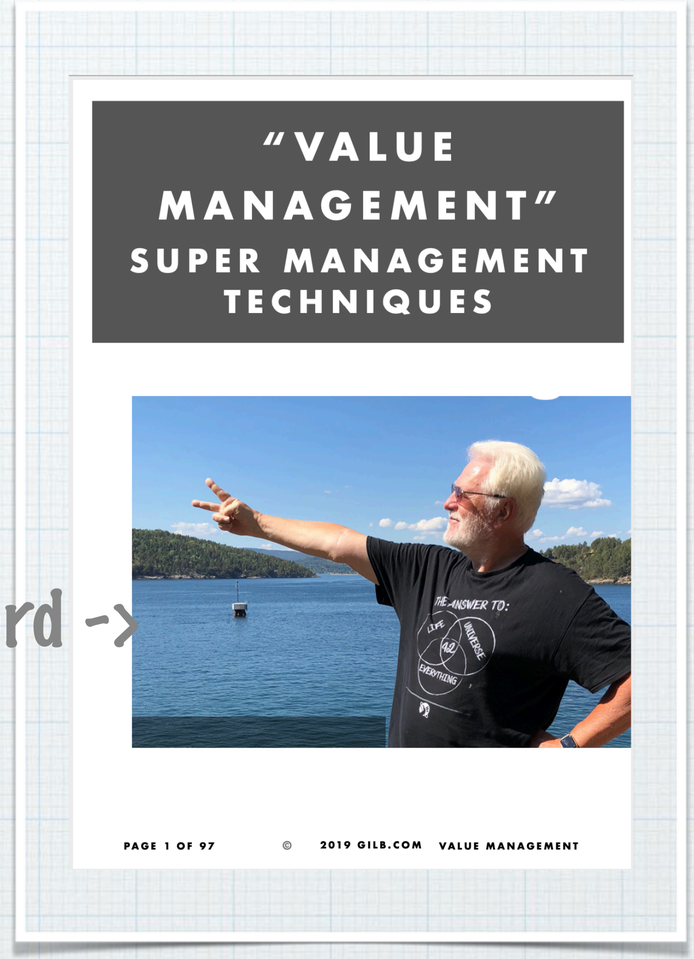
Slide Location: = https://tinyurl.com/ValueMgtBCS

By Tom Gilb, in Norway
(Kolbotn, near Oslo)
tom@Gilb.com
www.Gilb.com
@ImTomGilb (Twitter)
www.linkedin.com/in/tomgilb

Co-sponsored by BCS Specialist Group on Quality

‘Value Management’ Textbook
YOUR PERSONAL COPY

* ‘Value Management’ digital book
  * https://tinyurl.com/**ValueMgtBook**
  * Free, Shareware. (until published)
    * (share with a curious generous friend)
  * The VM book is the detailed basis for the sequence and content of these slides. Almost a mirror image



Oslofjord ->

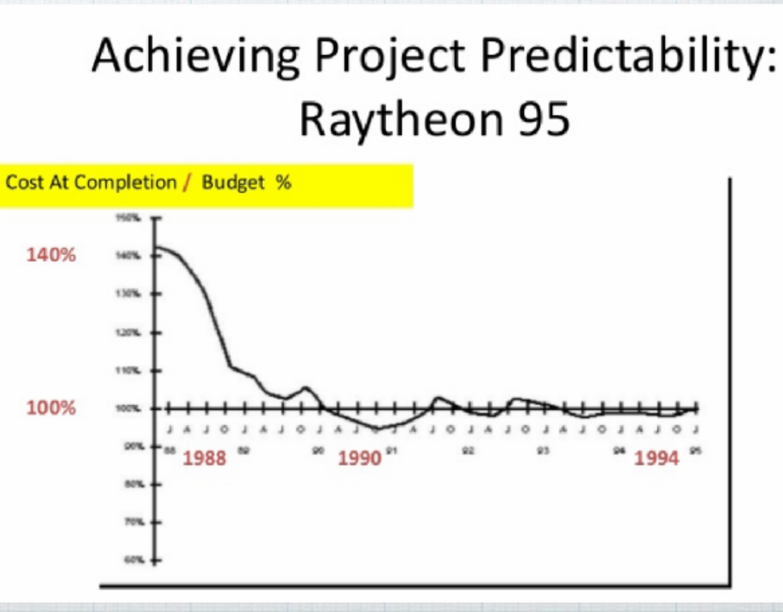
Free to generous people, share knowledge!
Others can pay £ 1,000 :)

In a sound bite

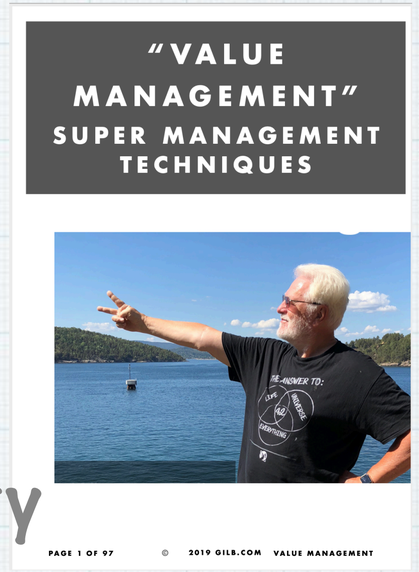
Value Delivery
Depends on

*Good**
Management

* Good = Defined in this course and book. IMHO :)



‘Good Management’
A 1-liner

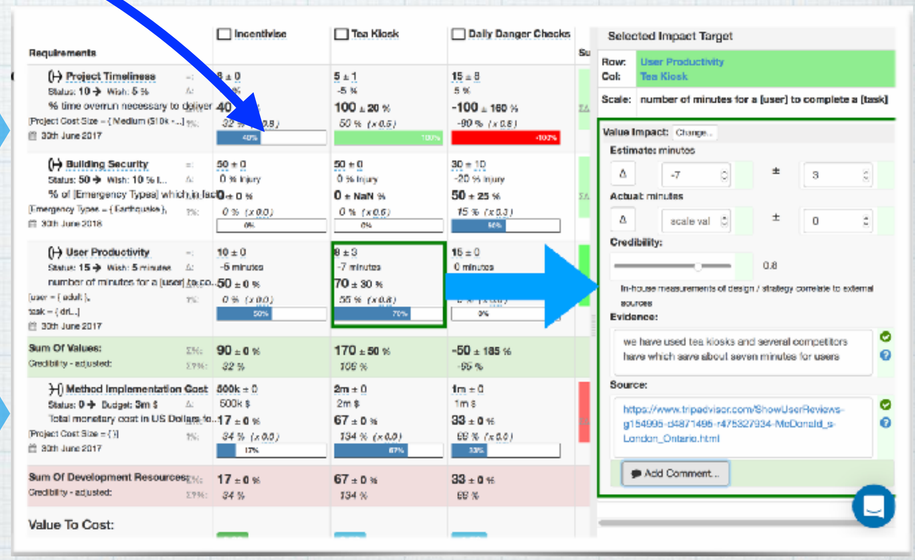


Management can improve value delivery by <u>quantifying all stakeholder values</u> as well as they quantify ‘money’ and ‘time’.

A Value delivery Estimate

Values

Costs



Good Management = partly this key idea. Quantify Values

5 Main Ideas

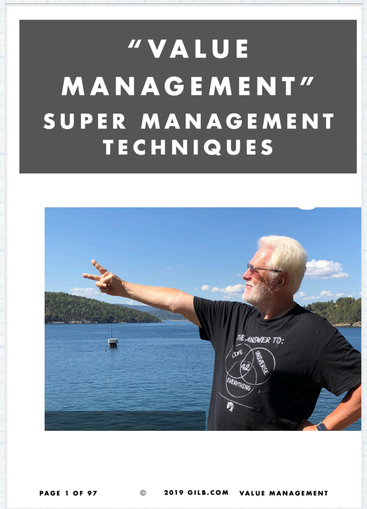
1. Stakeholders determine critical values

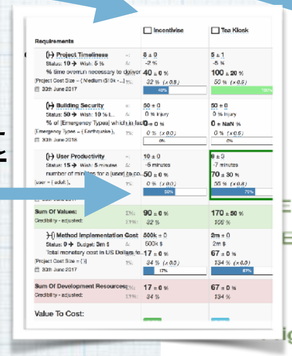
2. All critical values can be expressed as quantitatively as you do time or money

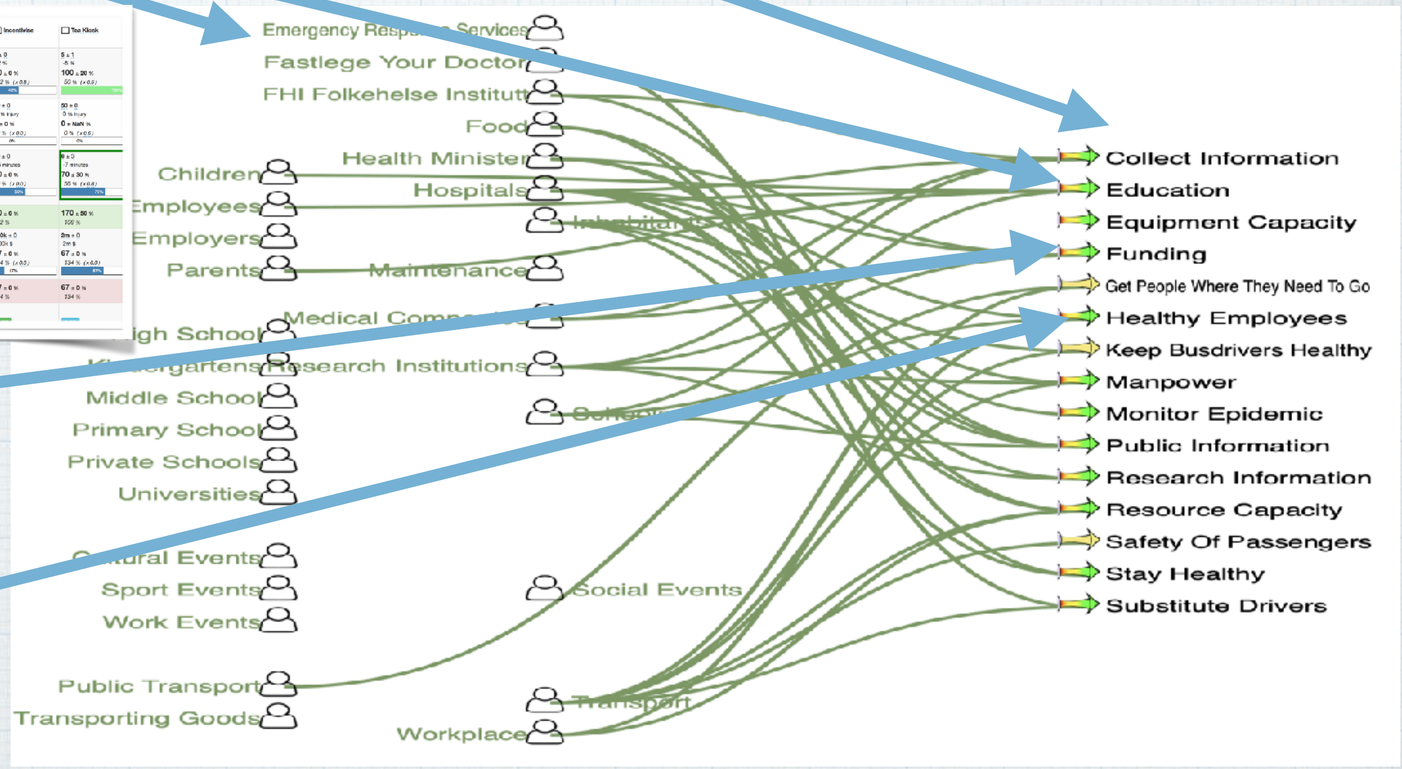
3. All strategies for delivering values can be estimated and measured for value and cost impacts.

5. Contracting can be based on real incremental delivery of useful value improvements

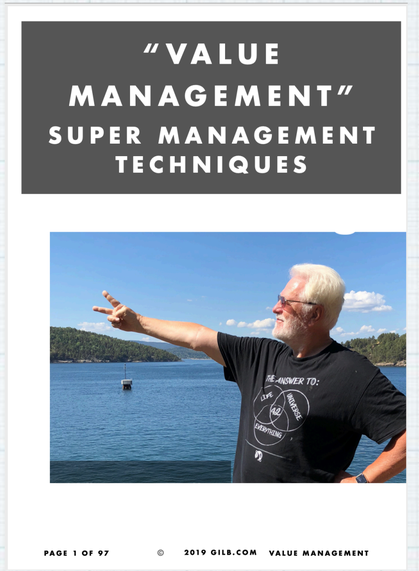
7. Motivation and responsibility can be value driven







Value Management Principles



* CRITICAL VALUES:
  * If you analyze your *stakeholders* well enough,
  * you will discover values *critical to your success* or failure

* QUANTIFY VALUES:
  * You can pin down all critical values *clearly*,
  * by *quantifying* them.

* ESTIMATE AND MEASURE STRATEGY* VALUES:
  * You can analyze, understand and make good decisions on strategies,
  * if you *estimate and measure* their values and costs.

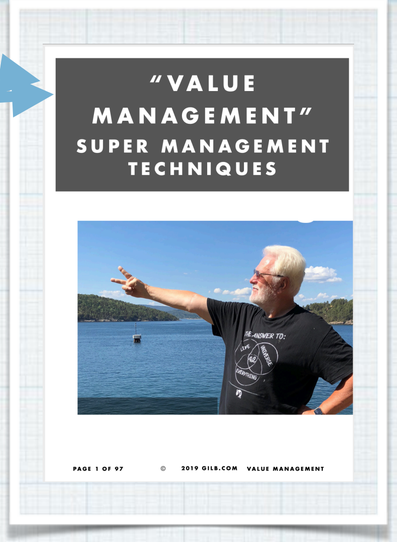
* EVOLVE EARLY FREQUENTLY:
  * You can incrementally deliver big successful improvements in values, in very early *small increments*,
  * so that failure is impossible, and impressive results are inevitable.

* FOCUS ON VALUES:
  * Extreme *focus* on values
  * will result in values:
    * do not get distracted.

* 6. VALUE-DRIVEN ORGANIZATIONS WIN:
  * If your organization does not seriously care about delivering real values,
  * then consider switching to one that will survive.

* Strategy = tactic, solution, design, architecture, means

2 Hour Course Content



Learn

Stakeholders

Values

Solutions

Decompose

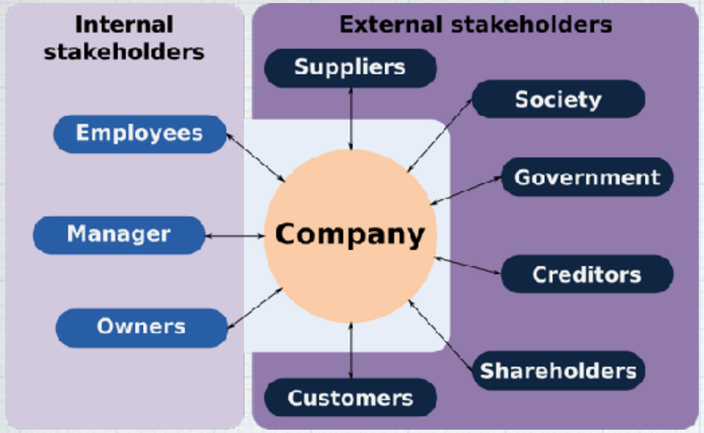
Develop

Deliver

Measure

Evolutionary
Value
Optimisation

Gilb Value Cycle



Learn

Stakeholders

Values

Solutions

Decompose

Develop

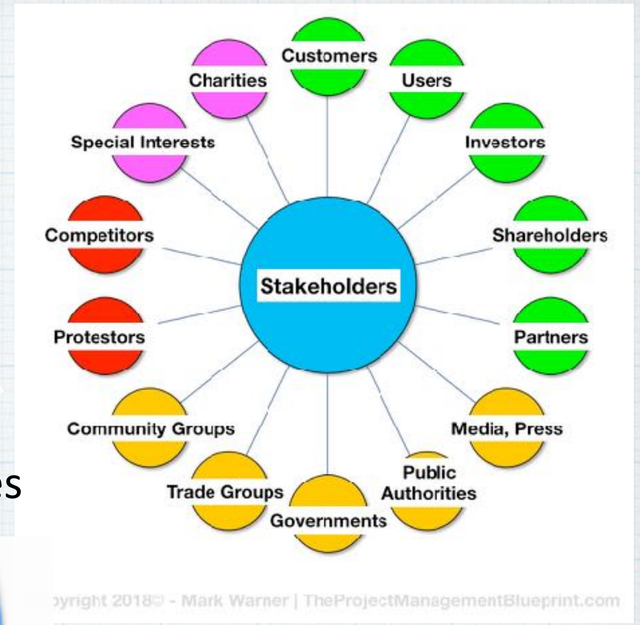
Deliver

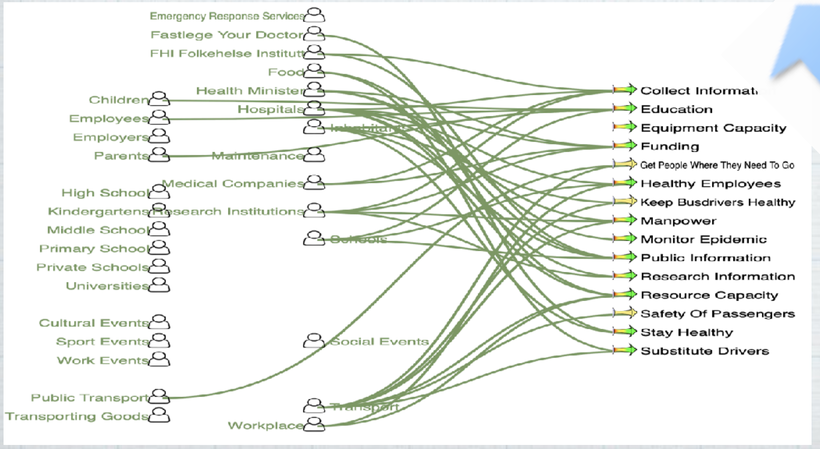
Measure

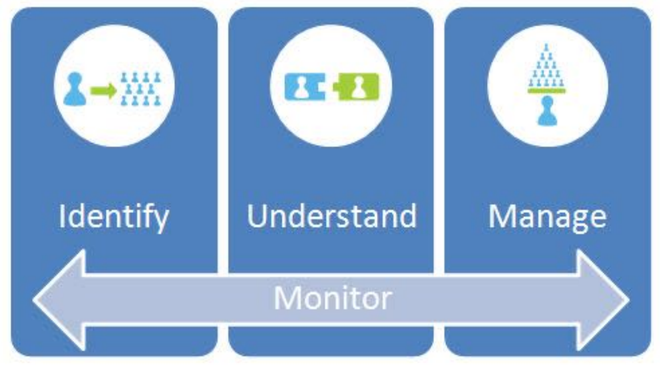
Identify Critical Stakeholders

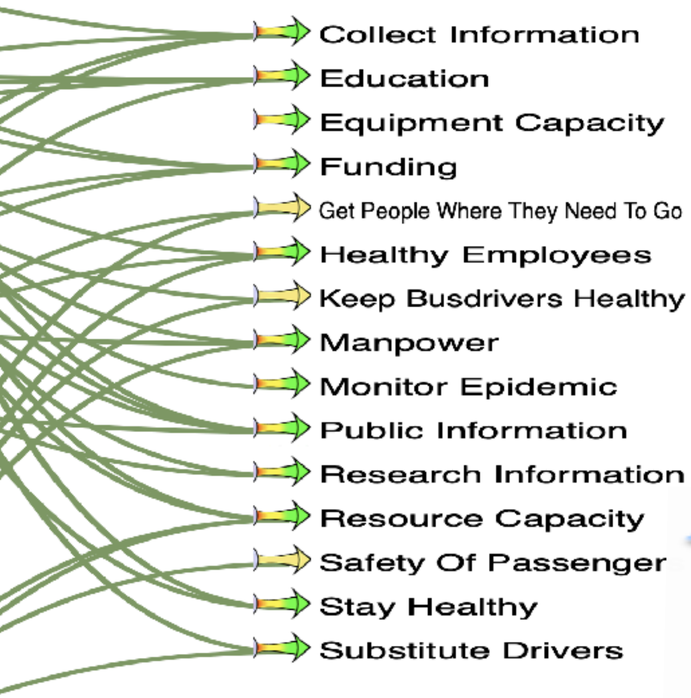
Who and what cares about the outcome of our project?

NOT just users and customers



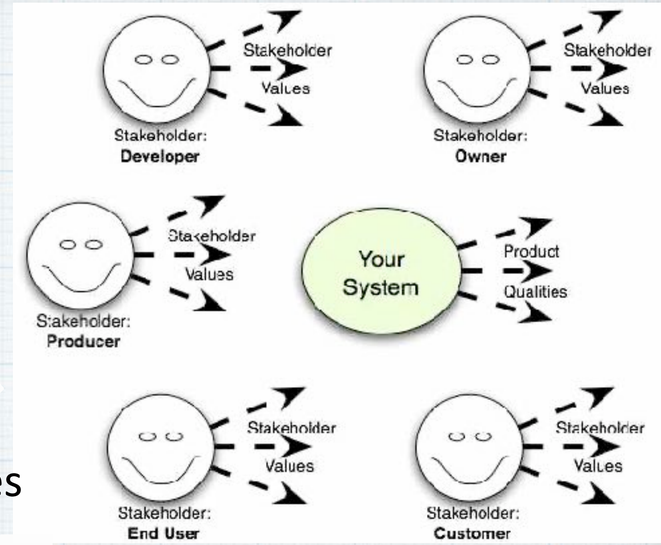






Learn

Stakeholders



Values

Measure

Specify Critical Value Requirements

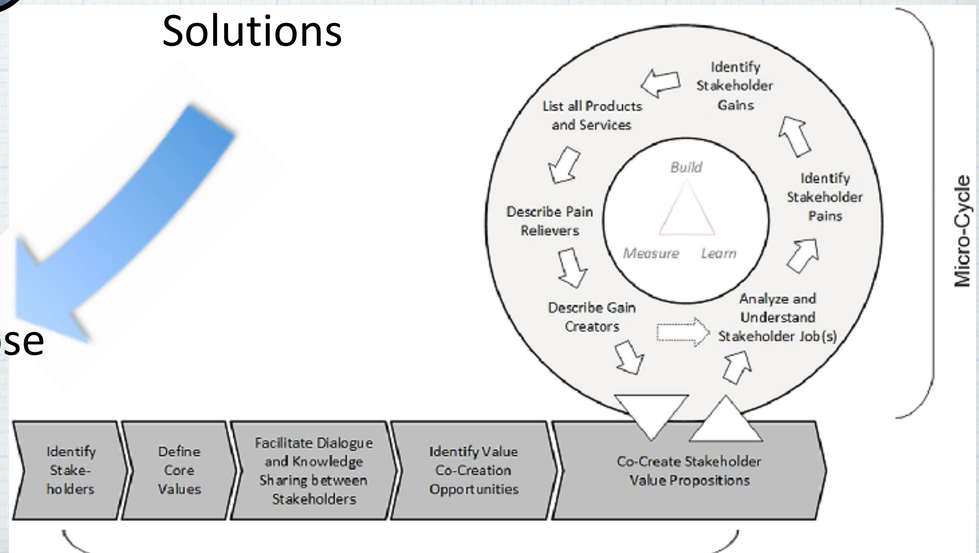
Find & specify quantitatively Stakeholder Values, Product Qualities & Resource improvements.

Solutions

Deliver

Develop

Decompose



Learn

Stakeholders

Values

Solutions

Develop

Decompose

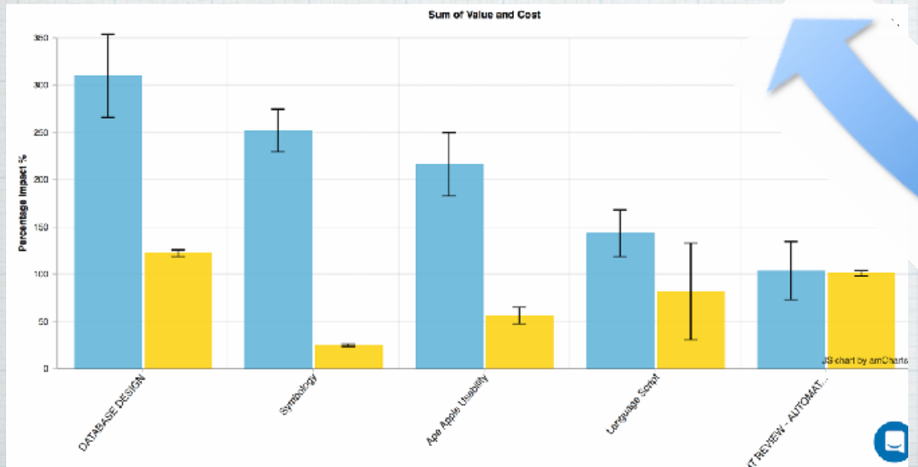
Deliver

Measure

Designs for Value

Find, Evaluate & Prioritize Solutions to satisfy Requirements.

Solutions prioritized By Values/Costs ratio Left to right



Learn

Stakeholders

Values

Solutions

Decompose

Develop

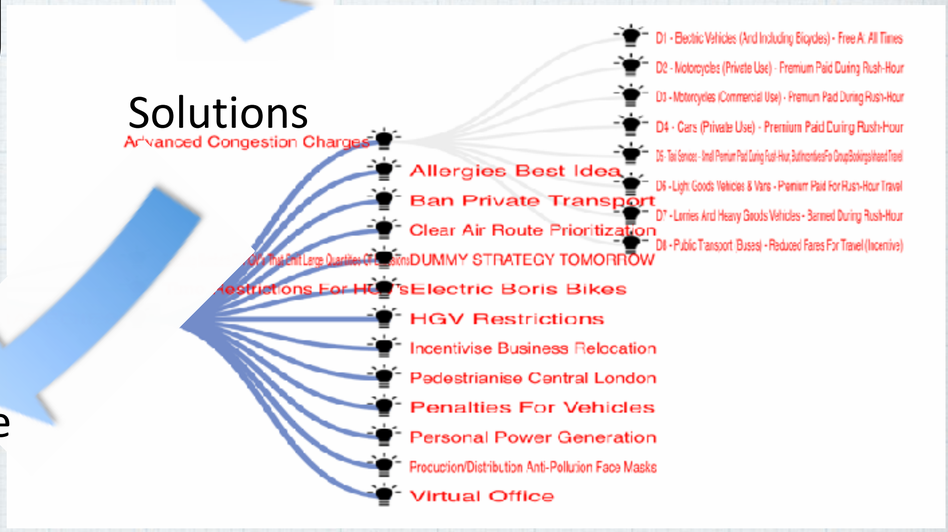
Deliver

Measure

Decompose into small Evo Steps

Decompose the winning Solutions down into smaller entities, then package them so they deliver maximum Value.

Sub-Value Decomposition



Value Decomposition

Learn

Stakeholders

Values

Solutions

Decompose

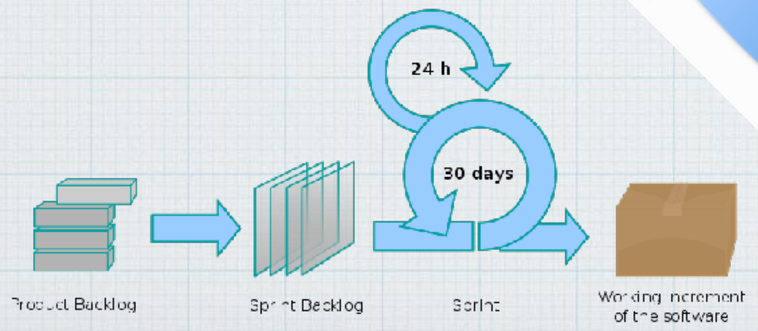
Develop

Deliver

Measure

Develop

Develop the packages that deliver the Value.



Learn

Stakeholders

Values

Solutions

Decompose

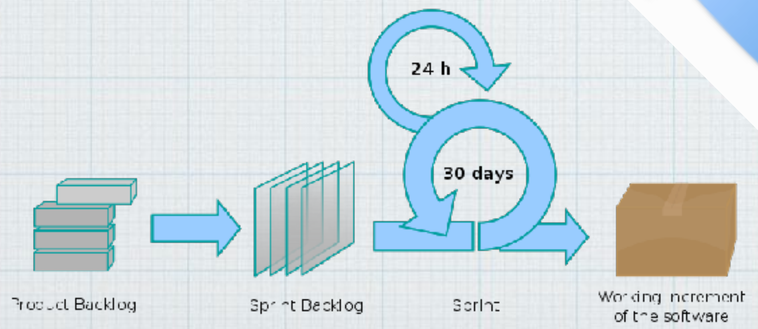
Develop

Deliver

Measure

Deliver

Deliver to Stakeholders
improved Value.
(not always a thing or code)



Learn

Stakeholders

Values

Solutions

Decompose

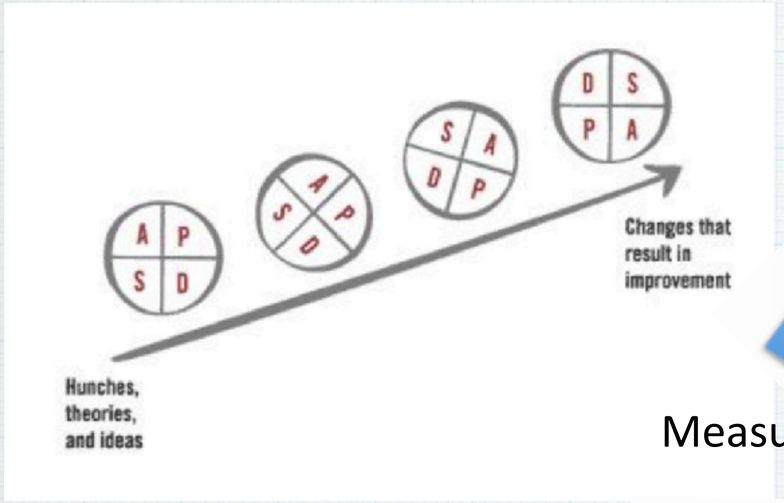
Develop

Deliver

Measure

Measure Value Improvement

Measure how much the Values changed.



Learn

Stakeholders

Values

Solutions

Decompose

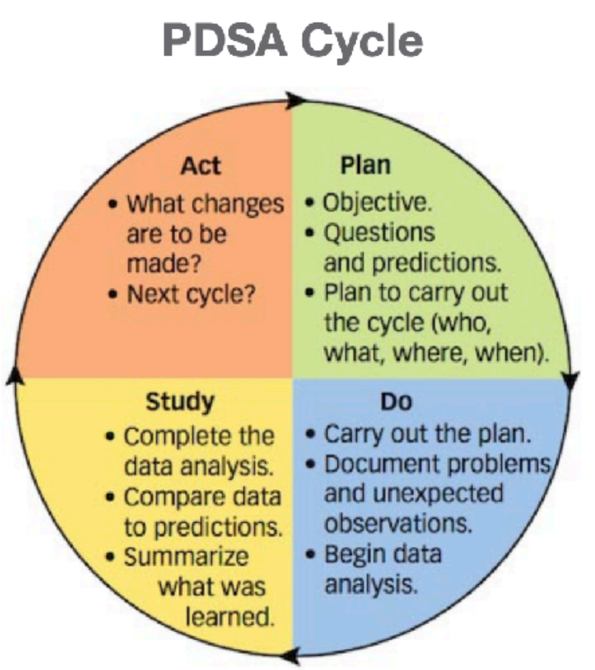
Develop

Deliver

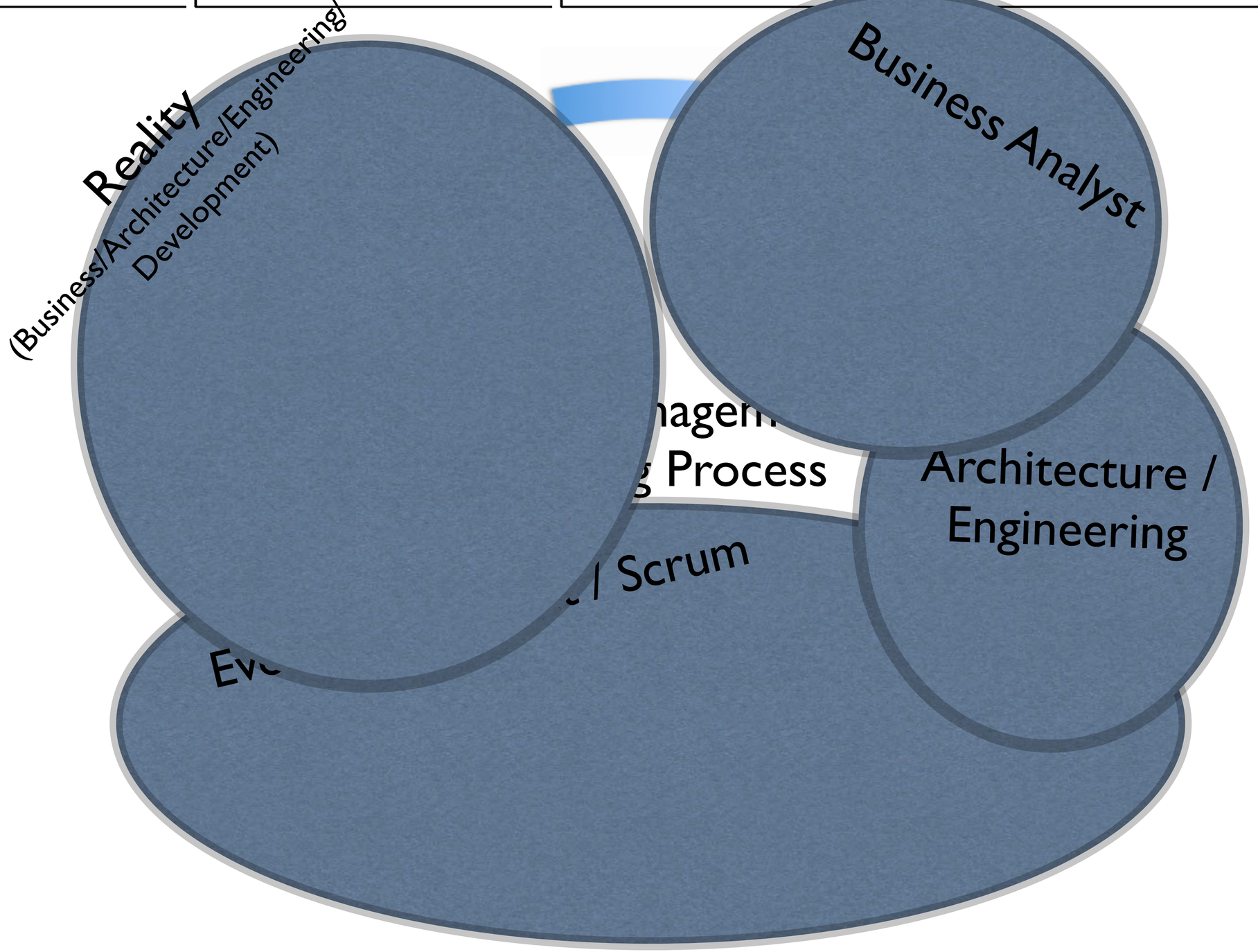
Measure

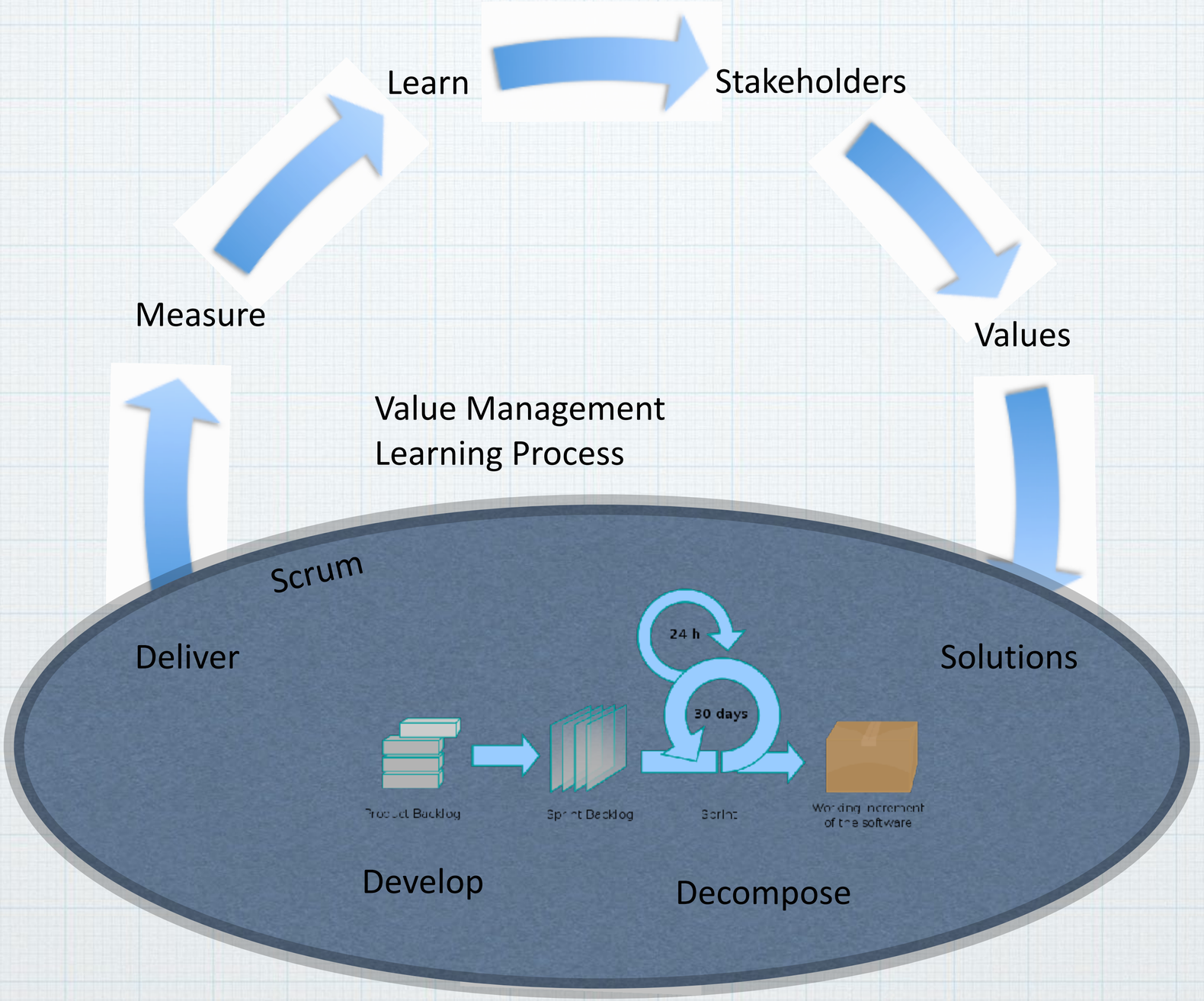
Learn & Improve Designs

Learning is defined as a change in behavior.

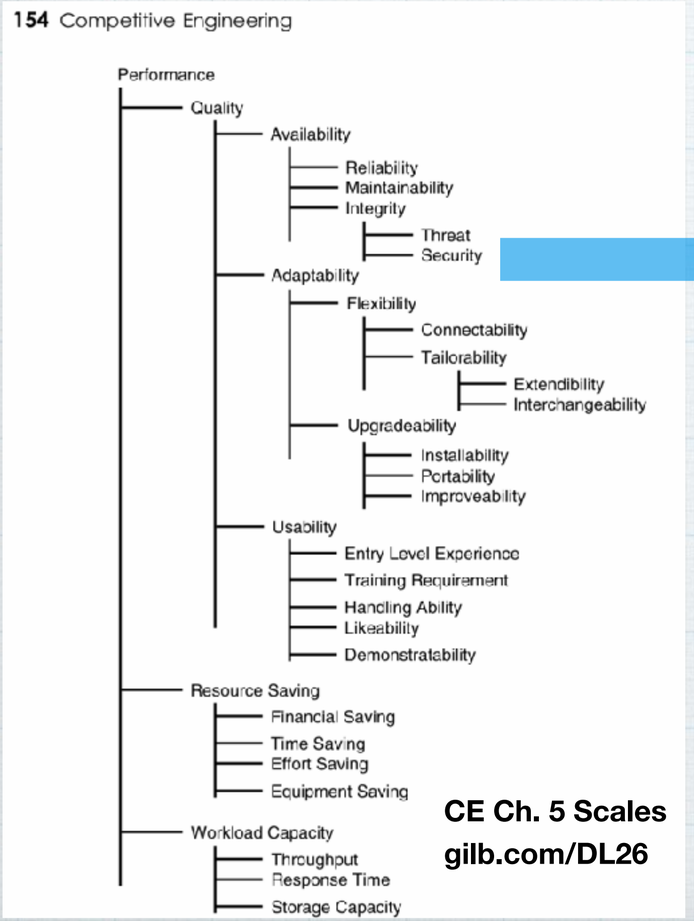


Deming PDSA


Reality
(Business/Architecture/Engineering/
Development)
Business Analyst
Architecture /
Engineering


Learn
Stakeholders
Measure
Values
Value Management
Learning Process
Scrum
Deliver
Solutions
24 h
30 days
Develop
Decompose

1. Managing Value Requirements



Security:

Scale: % probability of detecting a hacker within 5 seconds.

Status: 10% last year.
(Benchmark level)

Tolerable: 80% by End this year.
(Constraint Level)

Wish: 98% by End Next Year.
(Target Level)

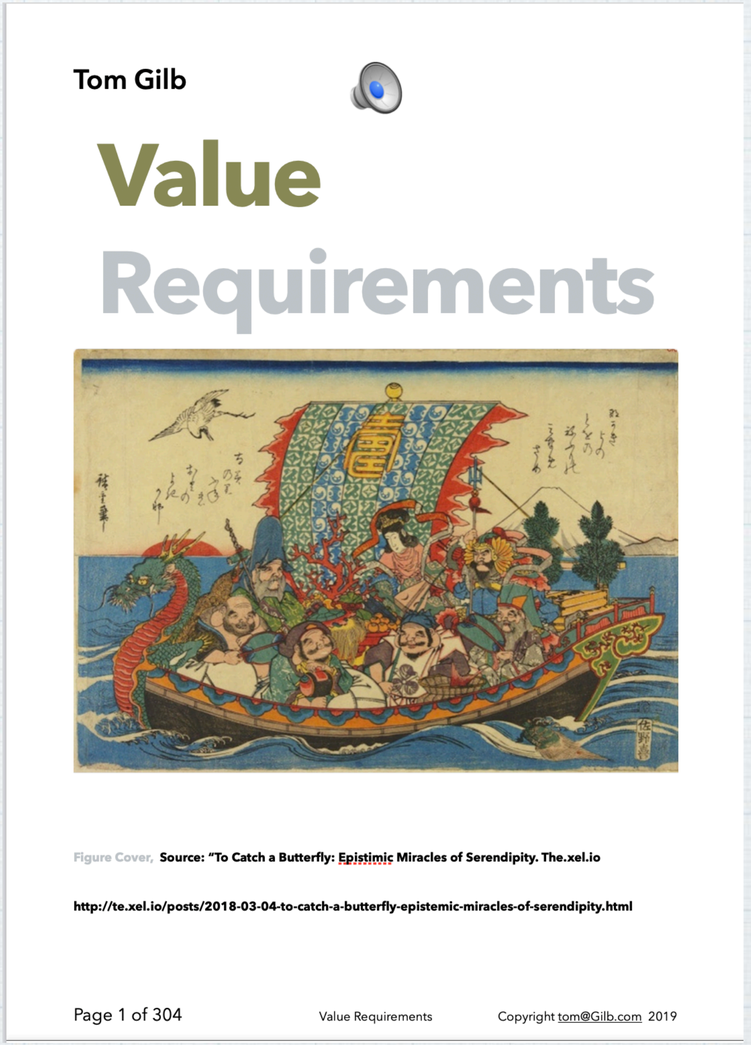
Value Requirements: the foundation

(previous value Requirements course),
not a prerequisite,
but quite useful) of the design problem articulation.

Wednesday 22nd April 2020

Video URL=
https://lnkd.in/dNEDuc6

Slide Location: =
http://concepts.gilb.com/dl970



https://www.dropbox.com/s/hxg1rx9rzesw2id/Value%20RequirementsPDF%20BEST%20%2070MBQ%20011019%202245%202.pdf?dl=0

Free download During Corona
Normal Price £ 1,000 from 1st April 2020
Value: > £1 million/project. :) avg.

Value Requirements and Resource Constraints

Requirements

Project Timeline
Status: **10** → Wish: **5**
% time overrun necessary to deliver
[Project Cost Size = { Medi... ($10k -...]
30th June 2017

Building Security
Status: **50** → Wish: **10** % l...
% of [Emergency Types] which ... fact...
[Emergency Types = { Earthquake },
30th June 2018

User Productivity
Status: **15** → Wish: **5** minutes
number of minutes for a [use... to co...
[user = { adult },
task = { dri...]
30th June 2017

Sum Of Values:
Credibility - adjusted:

Method Implementation Cost
Status: **0** → Budget: **3m $**
Total monetary cost in US Dollars fo...
[Project Cost Size = { }]
30th June 2017

Sum Of Development Resources:
Credibility - adjusted:

Value To Cost:

Which numeric improvements do stakeholders need, *critically*?

We can, and must always, express *their* values with *well-defined* numbers

Define both failure and success numerically

and

keep learning what those critical numbers are continuously

21

In our planning language, Planguage, these 3 levels might be expressed like this.

Security is the reference tag for the entire specification.

Scale is a parameter in Planguage for defining a value variable, such as Security, so that the various levels of Security can be expressed numerically.

Status gives us the moving current change of status in the level. Step by step feedback.

Tolerable gives us the bare minimum level which is acceptable. Worst acceptable case.

Wish is the stakeholder-desired, or stakeholder-needed, level of Security, on that Scale.

Wish = The 'Success level.

Security:

Scale: % probability of detecting a hacker within 5 seconds.

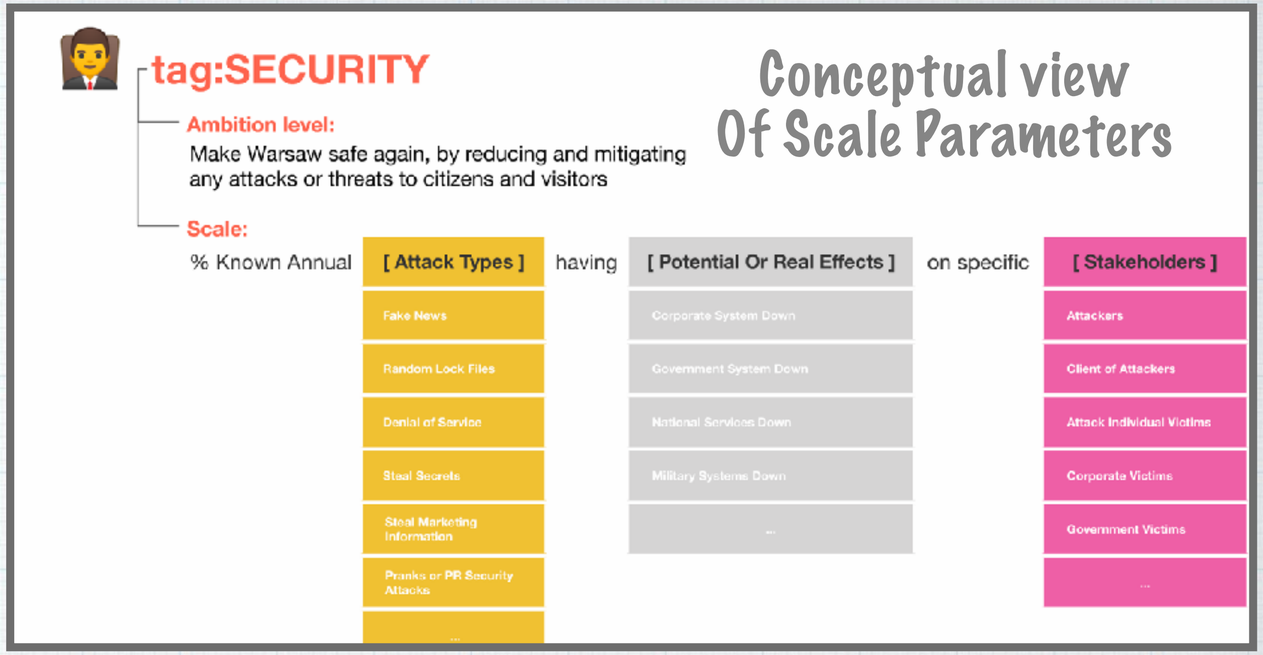
Status: 10% last year.
(Benchmark level)

Tolerable: 80% by End this year.
(Constraint Level)

Wish: 98% by End Next Year.
(Target Level)

Scale [Parameters]: A specification technique for decomposing complex requirements

- **[Scale Parameters] might seem 'complicated' at first sight.**
- **But it is in fact a way of *simplifying* very complex problems,**
  - **by allowing us to carefully extract, something simple**
  - **that we can work on, and**
  - **deliver some value improvements <u>early</u>, for critical subsets. (Hint: 'Agile as it should be')**
- **Early partial value delivery is *also* about**
  - **'learning about complex realities',**
  - **…. before we commit to 'scaling up'.**



Conceptual view Of Scale Parameters

Scale: Star Rating number for [**Person Type]** and [**Car Specs**] for [**Safety Equipment**] with [**Alternative Model Validity**] for a [**Publication Date**] by a [**Rating Agency**].

Wish: 5 Stars, by Next Year,

Person Type = All,

Car Specs= {Tesla 3, RWD, 4 Door, 2019},

Safety Equipment= {Front Airbag, Belt Pretensioner, Belt Load Limiter, Knee Airbag, Side Head Airbag, …},

Alternative Model Validity= Dual Motor AWD Model 3,

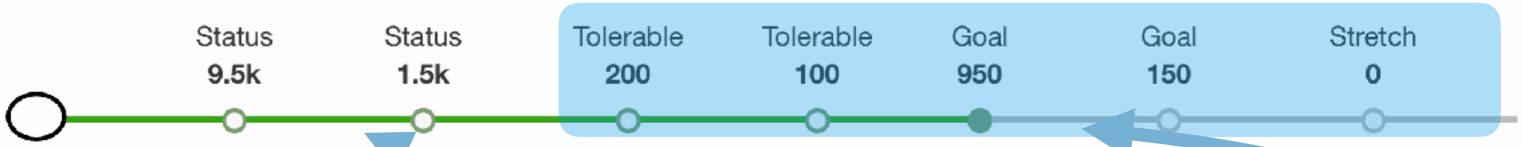
Publication Date =2019,

Rating Agency= All.

Air Quality

Level: Stakeholder, **Type:** Value, **Labels:** critical Edit

Is Part Of: Top Values



Goal [Persons = **<All>**, Pollution = **<All>**, Area = **<All>**] @ 2028 : **950** People <- *Sadiq Khan, Mayor of London Website, Press Release Toxic Air*

AQ.Ambition Level: Drastically improve air quality in London to acceptable legal levels as stated in the Paris Accord (Paris Agreement)

Scale: Number of [Persons] who reside in London Boroughs dying from exposure over [Time] to [Pollution] per year in [Area]

Meter: Recent Hospital records from London hospitals for deaths by Pollution Exposure related illness

Status: 9.5k People [Persons = **<All>**, Pollution = **<All>**, Area = **<All>**] When 2019

Status: 1.5k People [Persons = **Senior**, Pollution = **NO2**, Area = **<All>**] When 2019

Tolerable: 200 People [Persons = **Senior**, Time = **5 years**, Pollution = **NO2**, Area = **Greater London**] When 2022

Tolerable: 100 People [Persons = **Child**, Time = **1 Year**, Pollution = **{NO2, Carcinogens}**, Area = **Greater London**] When 2020

Goal: 950 People [Persons = **<All>**, Pollution = **<All>**, Area = **<All>**] When 2028

Goal: 150 People [Persons = **Senior**, Pollution = **NO2**, Area = **Greater London**] When 2029

Stretch: 0 People [Persons = **Senior**, Time = **5 years**, Pollution = **NO2**, Area = **Greater London**] When 2030

Stakeholders: Mayor Of London. Mayor of London

Assumption: IssueActionStatistic on number of deaths from pollution per year from Mayor of London press

Benchmarks

Requirements levels 3 types

We're Online!
How may I help you today?

Detailed 'Value Requirement' specification

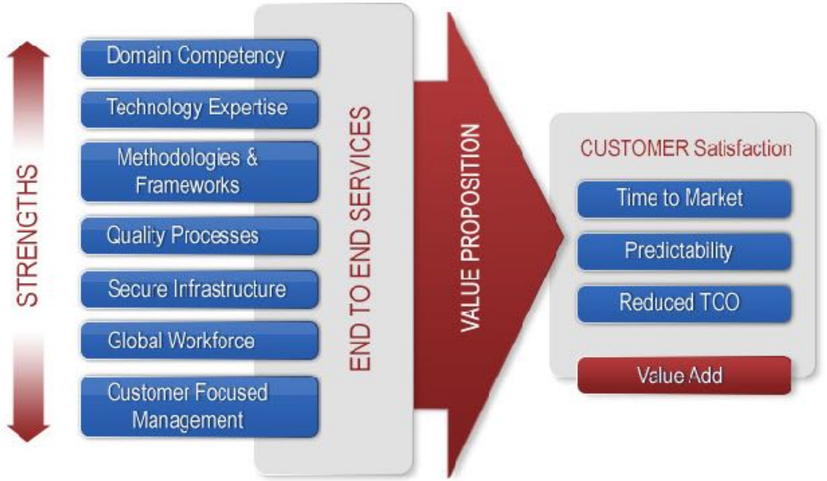
(1-line/statement view)

"Air Quality'

Source BCS Exercise Sept 2017, 'London Congestion for Air Quality'.

‘Predictability of Time To Market’: (Example from real case)

- **TTMP: Predictability of Time To Market:**
  - *Ambition: From Ideas created to customers can use it. Our ability to meet agreed specified customer and self-determined targets.*
  - **Scale: % overrun of actual Project Time compared to planned Project Time**
    - **Project Time: Defined:** time from the date of Toll-Gate 0 passed, or other Defined Start Event,
      to, the Planned- or Actually- delivered Date of All [Specified Requirements], and any set of agreed requirements.
    - **Specified Requirements: Defined:** written approved Quality requirements for products with respect to Planned levels and qualifiers [when, where, conditions].
      And, other requirements such as function, constraints and costs.
  - **Meter: Productivity Project or Process Owner will collect data from all projects, or make estimates and put them in the Productivity Database for reporting this number.**
  - **Past [1994, A-package] < 50% to 100%> <- Palli K. guess.**
    [1994, B-package] 80% ?? <- Urban Fagerstedt and Palli K. guess
  - **Record [IBM Federal Systems Division, 1976-80] 0%**
    <- RDM 9.0 quoting Harlan Mills in IBM SJ 4-80
  - ***“all projects on time and under budget”***
  - **[Raytheon Defense Electronics, 1992-5] 0% <- RDE SEI Report 1995**
    Predictability.
  - **Fail [All future projects, from 1999] 5% or less <- discussion level TG**
  - **Goal [All future projects, from 1999] 0% or less <- discussion level TG**



This detail is referenced in slide 54 ‘Ericsson Case’

From Ericsson case study on engineering productivity in the Value Requirement slides

The Evo Process., the One Page Summary

A Systems Engineering Agile Process

Stakeholder Value Requirements

Clarify your environment: critical-stakeholders' territory

1. Identify your critical **stakeholders**: the ones that can make or break your project
2. Identify their critical **values**
3. **Quantify** and **clarify** your critical values: what degree of values do you expect the design to deliver to stakeholders
4. Identify design **constraints**: legality, political, cultural, policy, other plans
5. Identify design **resource-limitations**: time, money, operational costs for example.

Top-Level Design: Architecture Level

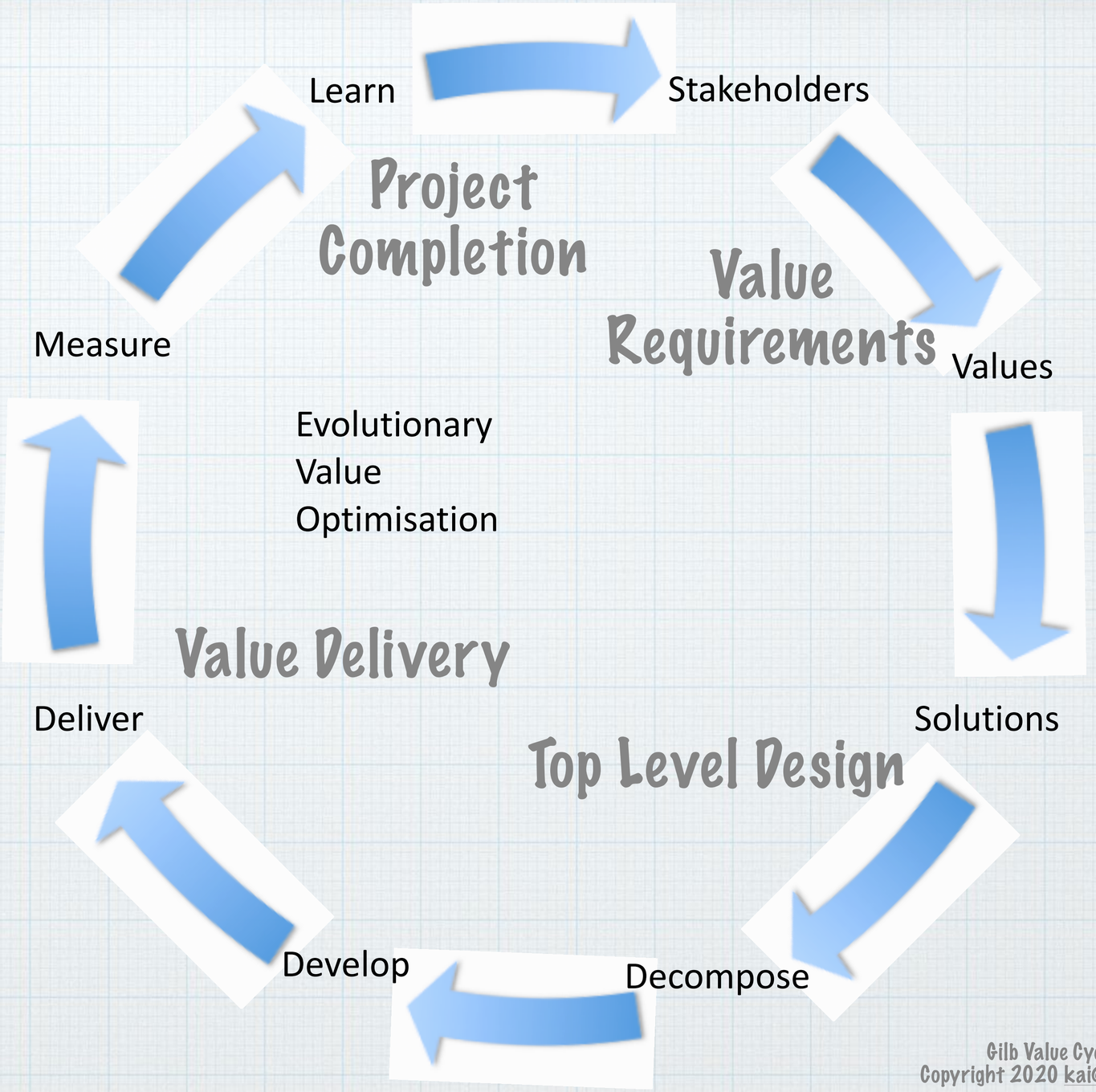
6. The Project Startup Week: an **architecture overview**
7. Identification and **prioritization** of top-level architecture
8. **Decomposition** of top-level architecture into design components.

Value Delivery Cycle (Designs deliver the value !)

9. The **Evo Value delivery steps** (about a week, or 2% of total project budget)
   1. Select your highest **priority value**, and the most-critical scale-parameter attributes.
   2. Find a design component which will deliver the most **value-for-resources** to your priority requirement.
   3. Ready the design component for delivery: **integration to the existing system.**
   4. **Deliver** the design component to the **real system**
   5. **Measure** the results (values and costs) of the design increment
   6. If results are **negative**, attempt design **improvement, and redeliver.**
   7. If results **ok** then **repeat** this value delivery cycle, scale up, until 'done'.

Project Completion: All Value is delivered

10. When **all value requirements are reached**, or when **critical resources are used up**. Stop.



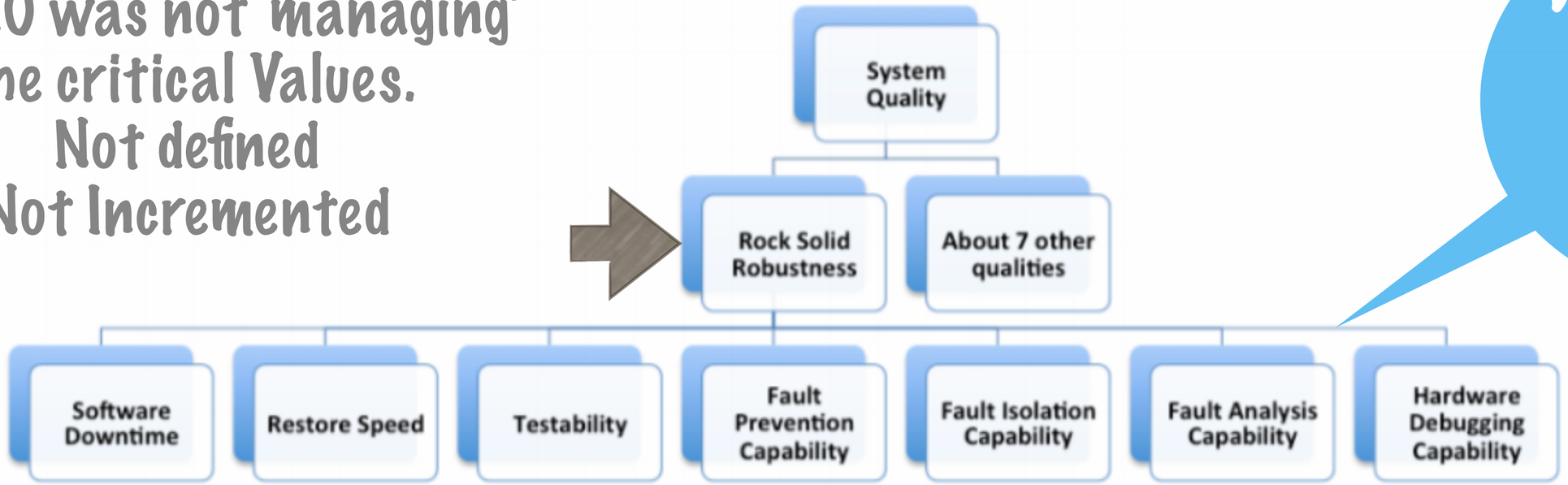
End of requirements re-cap, from specialist Value Requirements course (3 Hours BCS Video)

https://www.gilb.com/valuefirst-requirements-online-course
Requirement Online Kai Gilb 2020

The following slides
Introduce management ideas from the Value Management book
Related to Requirements and Value Objectives

The 8 year, 100 person, failed project, CEO Requirements-driven 'Rock Solid Robustness' the CEO dictated

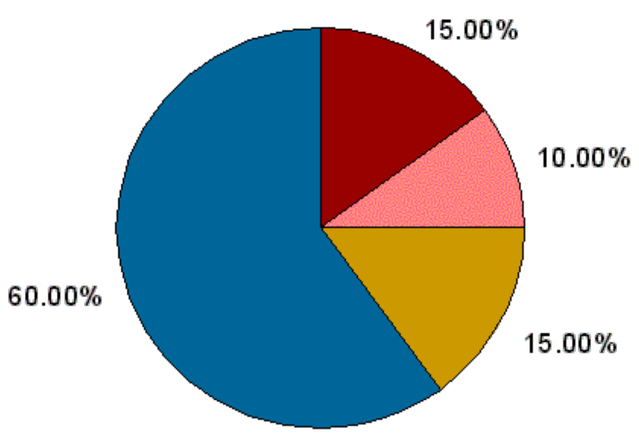
The CEO was not 'managing'
The critical Values.
Not defined
Not Incremented



Decomposition Of Rock-Solid Robustness

FIGURE 1 (Robustness) In the case above, the key to decoding what the CEO wanted for Robustness, was to decompose it into 7, fairly-conventional engineering understandings of it. Then to quantify each one of them. See next 3 slides for detail of some.

Hot Tip
(Cartesian Analysis)
Decomposition is often a good
First step towards clarification
And then quantification



Software Downtime:

What the CEO and CIO failed to do for 8 years

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].>

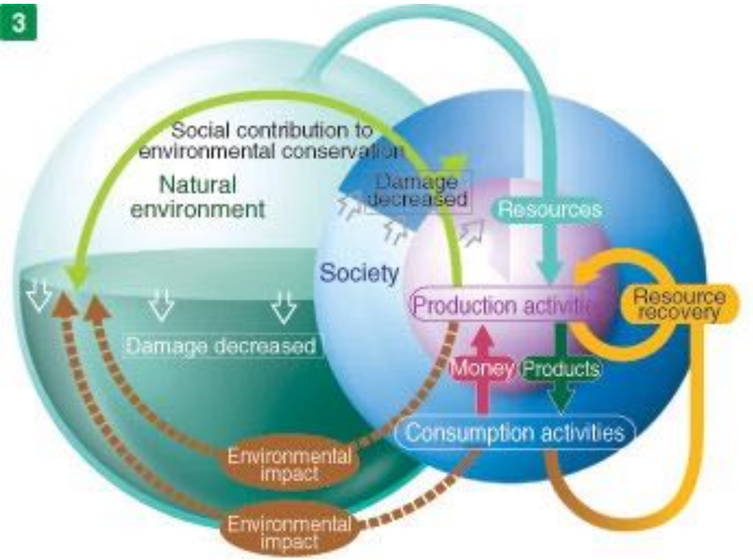
Fail [Any Release or Evo Step, Activity = Recompute, Intensity = Peak Level] 14 days <- HFA 6.1.1

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.

What the CEO and CIO failed to do for 8 years

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*

https://en.wikipedia.org/wiki/Software_testability

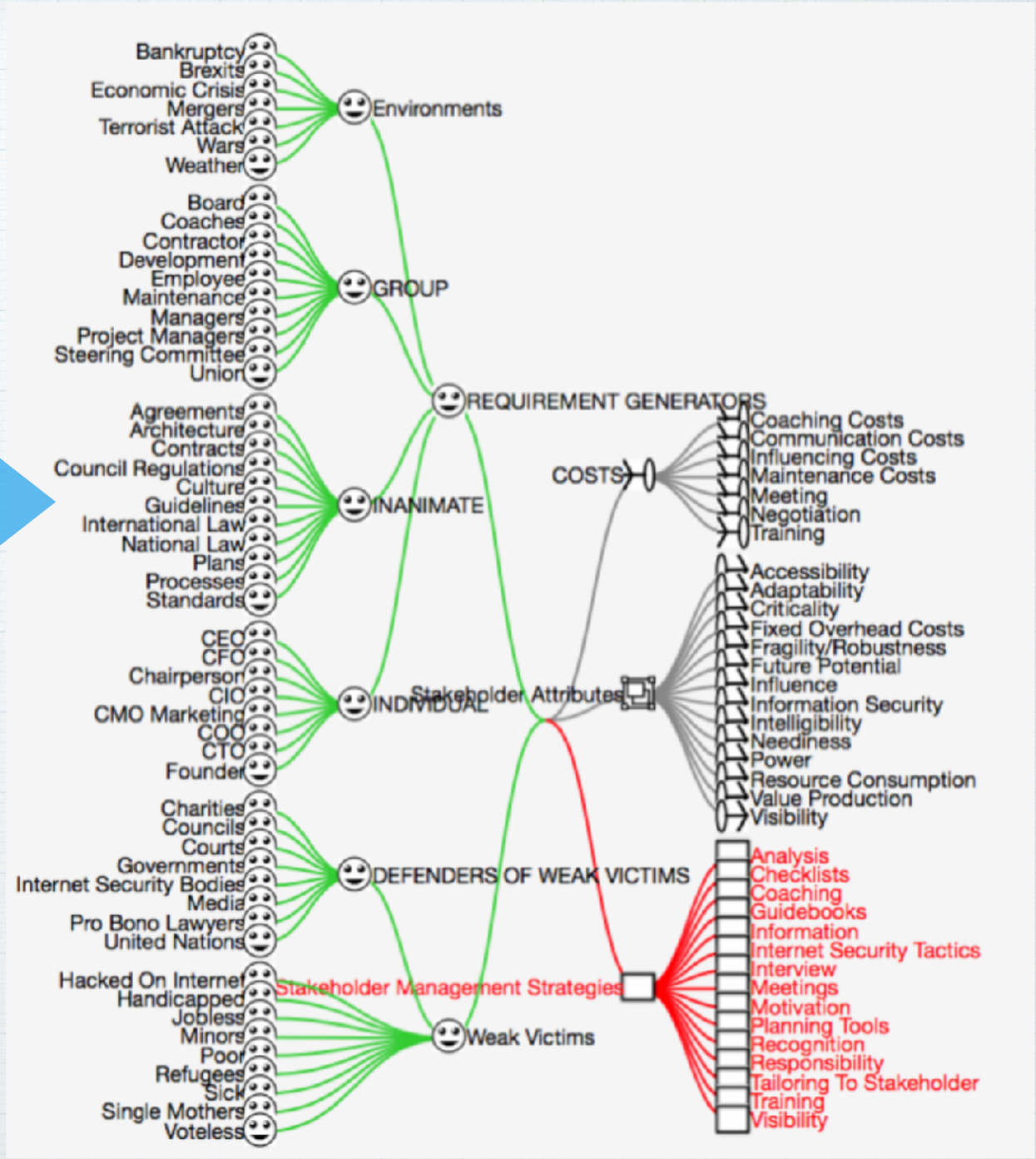
What the CEO and CIO failed to do for 8 years

More ,
effective ways, of managing value requirements

Stakeholder

Management

(Not ‘User,’ and/or ‘Customer’ alone)



Security

Level: Product, **Type:** Value, **Labels:** - Edit

Wish [Security Results = **Attack Attempt Detected**, Attacks = **{Take Control of System, Steal Money}**, Attackers = **Evil People**, Targets = **Organization**, Attack Results = **All**] @ 06 Jul 2022 : **42**

Ambition Level: I want the best security to fight hackers and protect my customers and company.

Scale: % of [Security Results] for [Attacks] carried out by [Attackers] on [Targets] with [Attack Results].

Status: 5 [Security Results = **Attack Attempt Detected**, Attacks = **{Take Control of System, Steal Money}**, Attackers = **Evil People**, Targets = **Organization**, Attack R...

Wish: 42 [Security Results = **Attack Attempt Detected**, Attacks = **{Take Control of System, Steal Money}**, Attackers = **Evil People**, Targets = **Organization**, Attack R...

Goal: 0 When ?

Stakeholders:

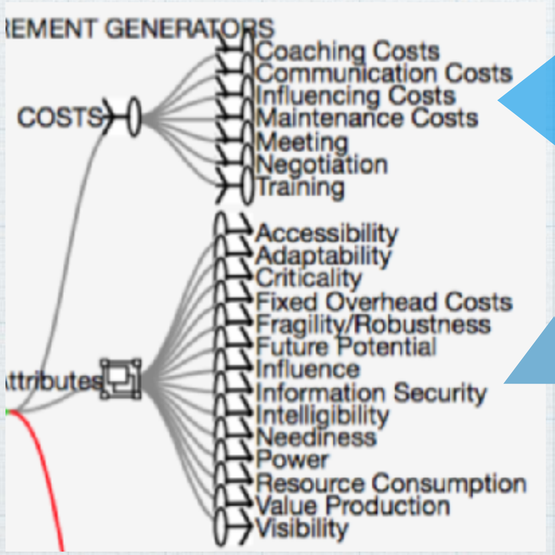
Stakeholder	Roles	Notes
From: Board Of Directors		
From: Chief Security Officer		
From: Our Security Consultant F-Secure		

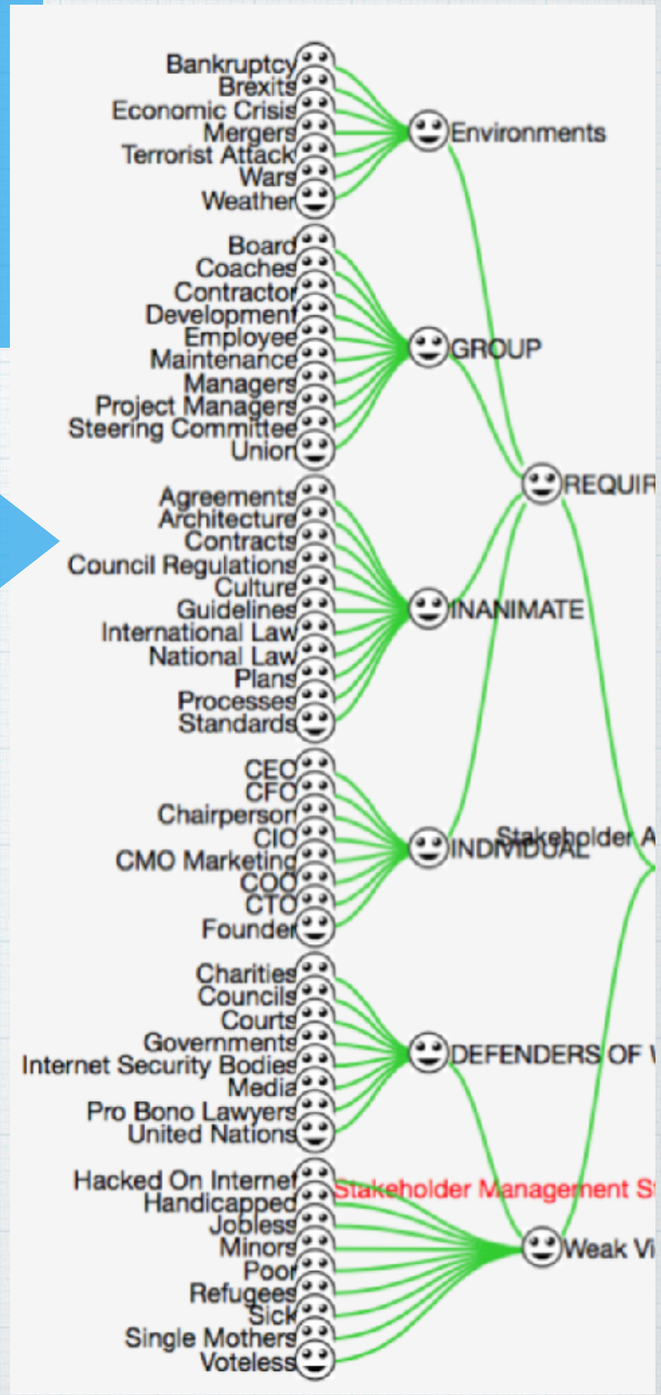
A Security Value Objective Spec example

Value Requirements Principles (warning signals)

1. If you focus on ‘users and technology’, instead of stakeholders, values, and what is critical: you will probably fail

2. If you quantify values, and make all related conditions clear, you have the basics for value-delivery success; otherwise failure is pretty well guaranteed.





Value Requirements
Policies
(leadership declarations)

Action options For managers Tomorrow

1. We will discover and track all critical <u>stakeholders</u>,

in order to discover the critical <u>values</u> and constraints of our project.

2. We will <u>quantify</u> and <u>clarify</u> all critical values,

so they cannot be misunderstood, and will be delivered as specified.

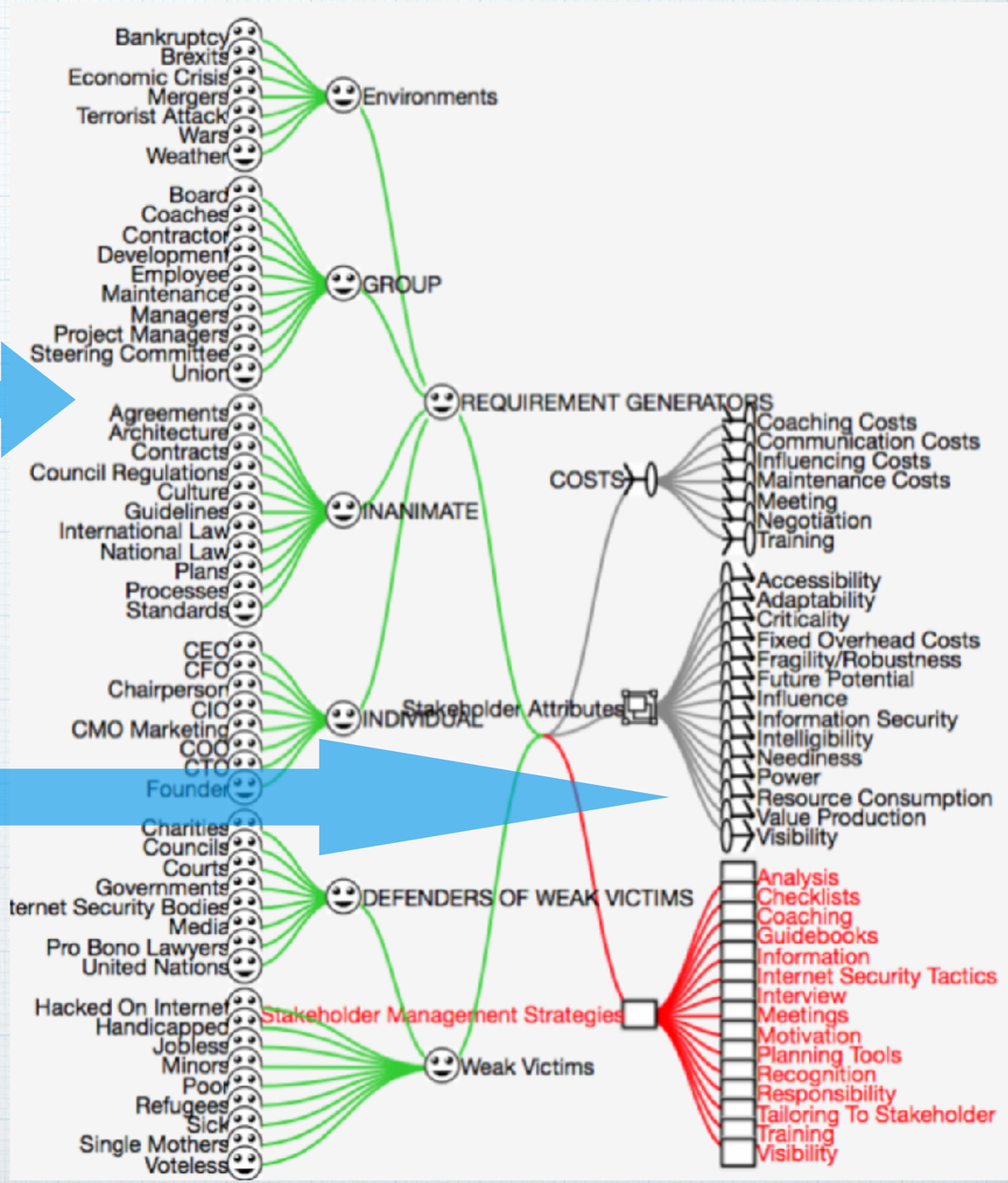
<u>Security</u>:

Scale: % probability of detecting a hacker within 5 seconds.

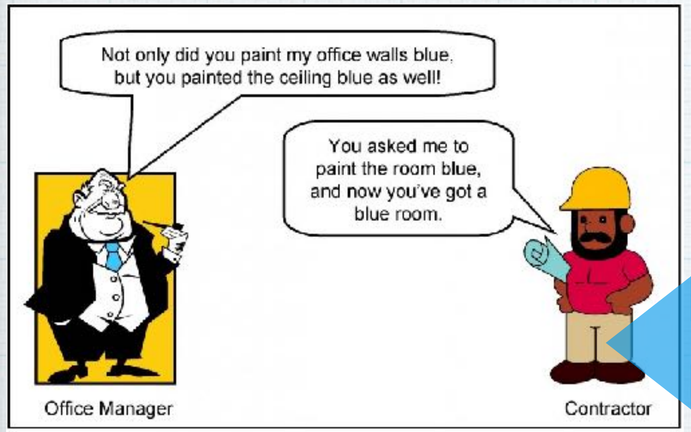
Status: 10% last year.
(Benchmark level)

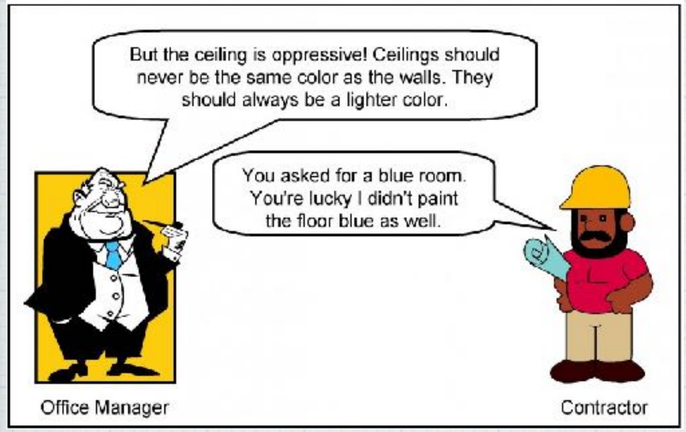
Tolerable: 80% by End this year.
(Constraint Level)

Wish: 98% by End Next Year.
(Target Level)



Value Requirements Checklists (analytical warnings)



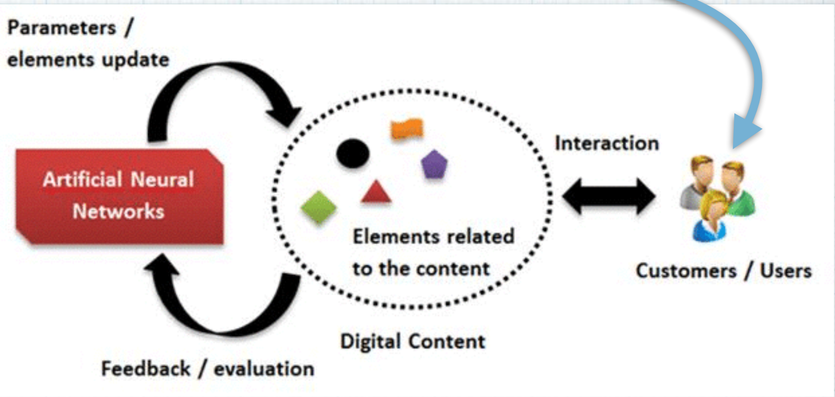


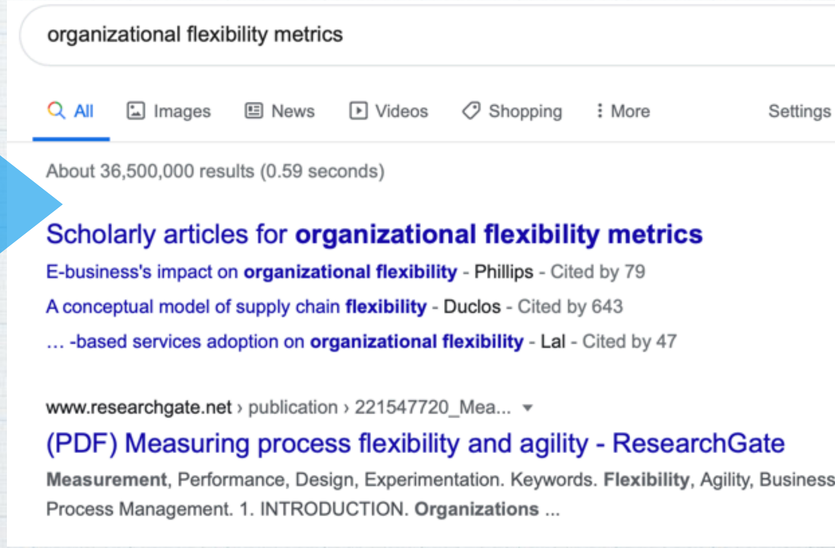
1. Are your people only talking 'users' and 'customers', or do they have an 'all critical stakeholders' culture?

2. Are the requirements, objectives, and 'expected benefits' of the project specified in fuzzy words? Can they be rewritten clearly and quantitatively instead?

3. When you seem to have difficulty quantifying a value (like 'Security'), try searching the internet with a keyword like 'Security metrics'.

   It is amazing how many people have solved this quantification problem for you already. For free for you.





End of Part 1: Requirements/Objectives? Goals/Targets

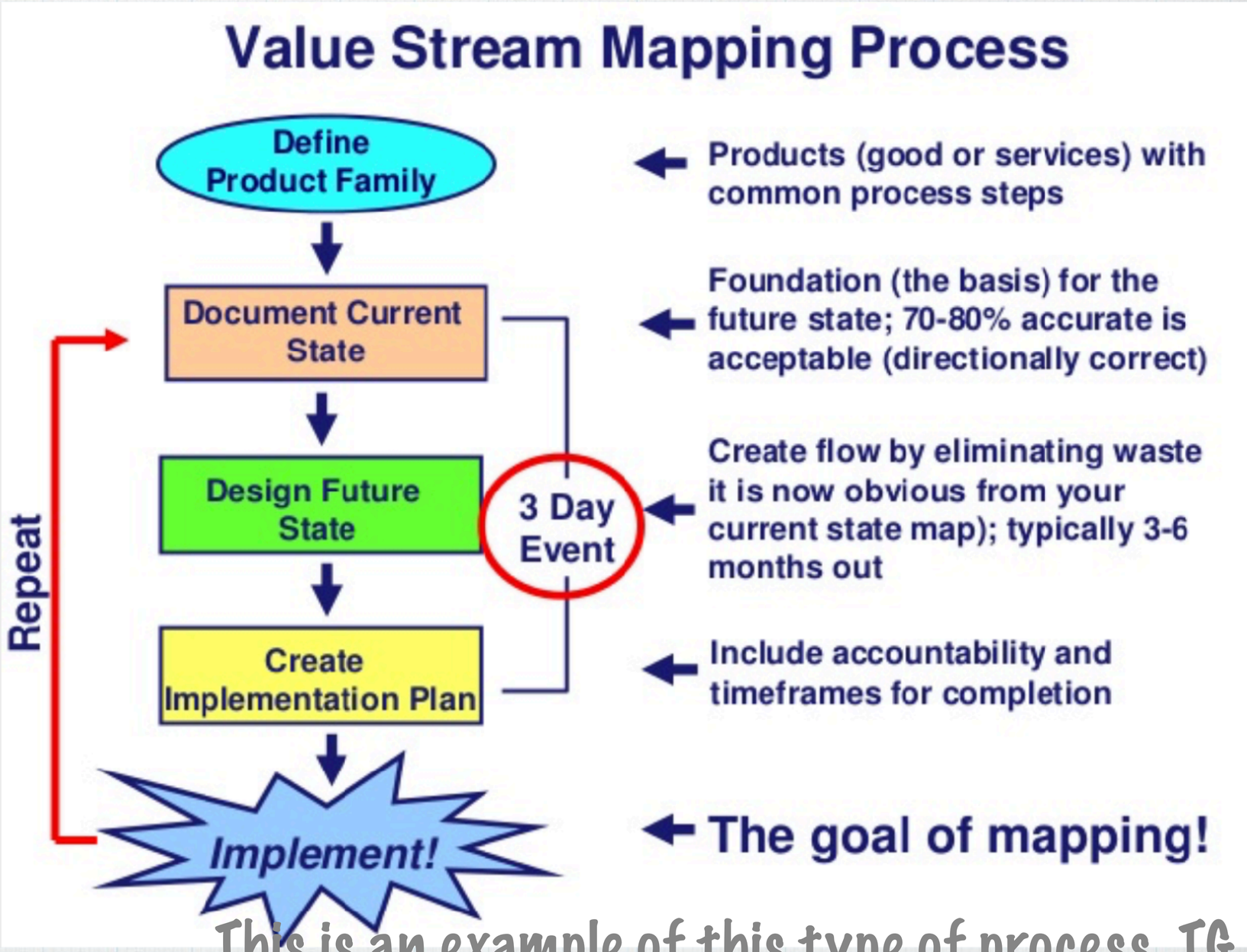
Part 2.
Managing Value Design

Figure out exactly which means/strategies/ architecture/solutions

will deliver

the multiple values you want

within your resources and constraints



This is an example of this type of process. TG

https://www.slideshare.net/KarenMartinGroup/value-stream-mapping-in-office-service-setttings/31-Value_Stream_Mapping_ProcessProducts_good

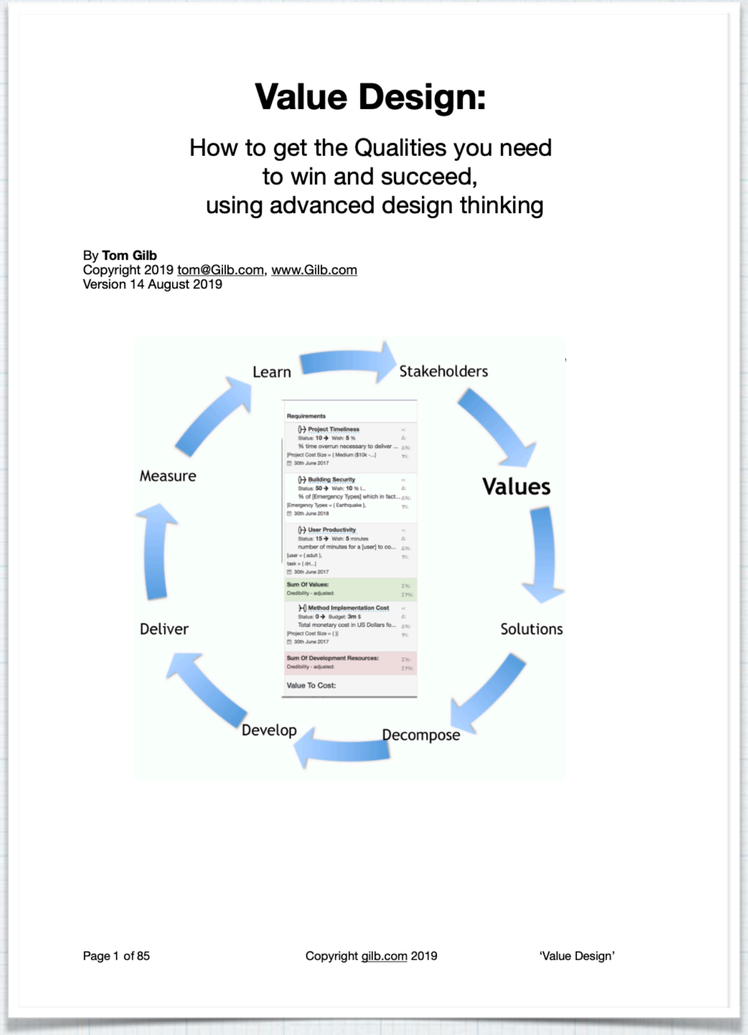
3 Hour 'Value Design' Course Content
Previous course,
not Value Management (This one)

Main subjects:
Value Requirements: the foundation (previous value objectives course, not a prerequisite, but quite useful) of the design problem articulation.

What is Design?

1. **Basics for Valid Design**
2. **How not to evaluate a design**
3. **Simple ways to evaluate and compare designs**
4. **More powerful ways to evaluate design options**
5. **Starting a larger project, first design week.**
6. **Decomposing designs, some more ideas**
7. **Risk management for value design**
8. **Design Prioritisation.**
9. **Dynamic Design to Requirements**
10. **Organising the value design process**

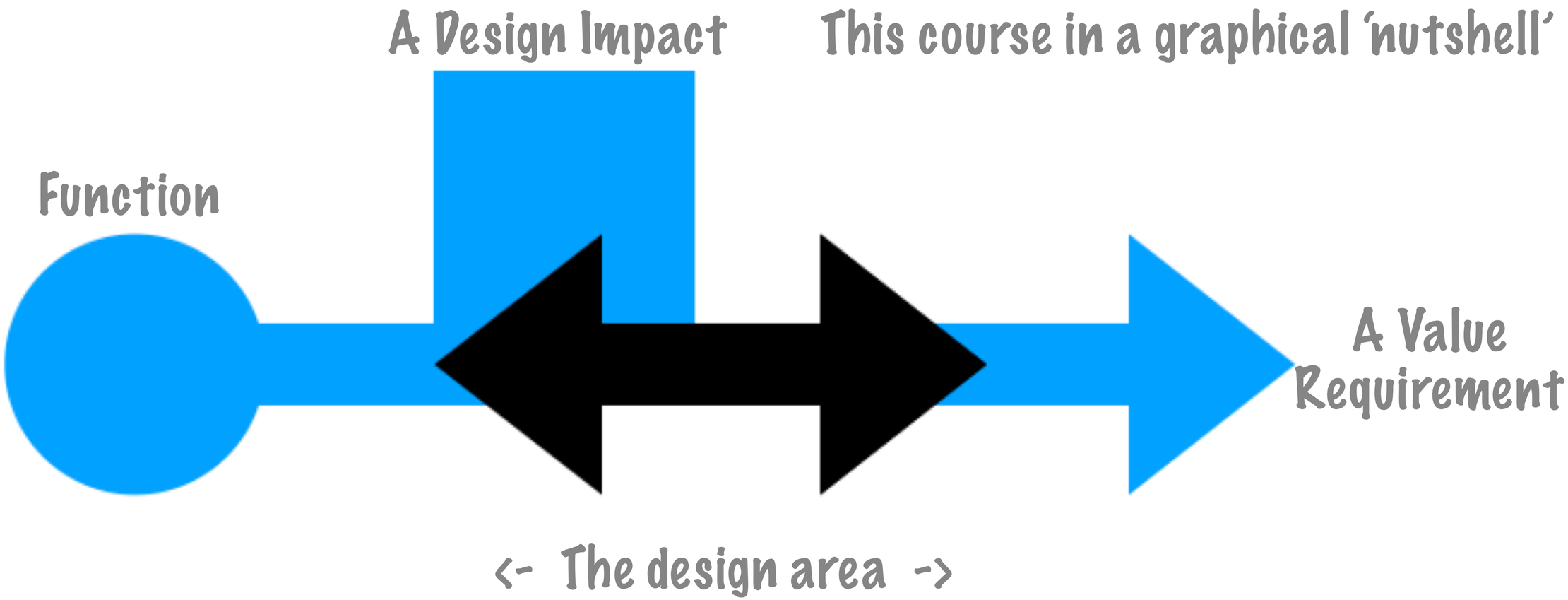
The star of this show is the Value Decision table aka Impact Estimation Tables) which allows us to see any set or level designs, with their estimated or real effects on our critical objectives and costs. A unique general tool for all designers. Complex systems at a glance, on a page or screen.



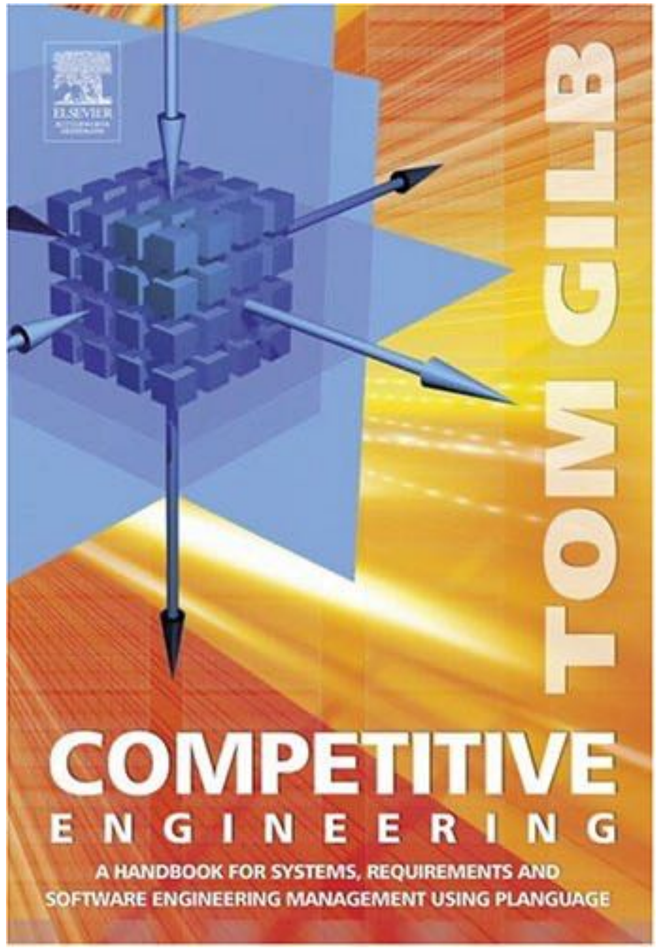
Documentation: the course is based on our recent digital textbook
"Value Design:
How to get the Qualities you need to win and succeed,
using advanced design thinking."
Book Price normally = £1,000 (as of April 1 2020)
Digital copy free to course participants.
Book Knowledge Value > £1,000,000,000

https://tinyurl.com/FreeValueDesign

The ‘Design Theory’ Icon

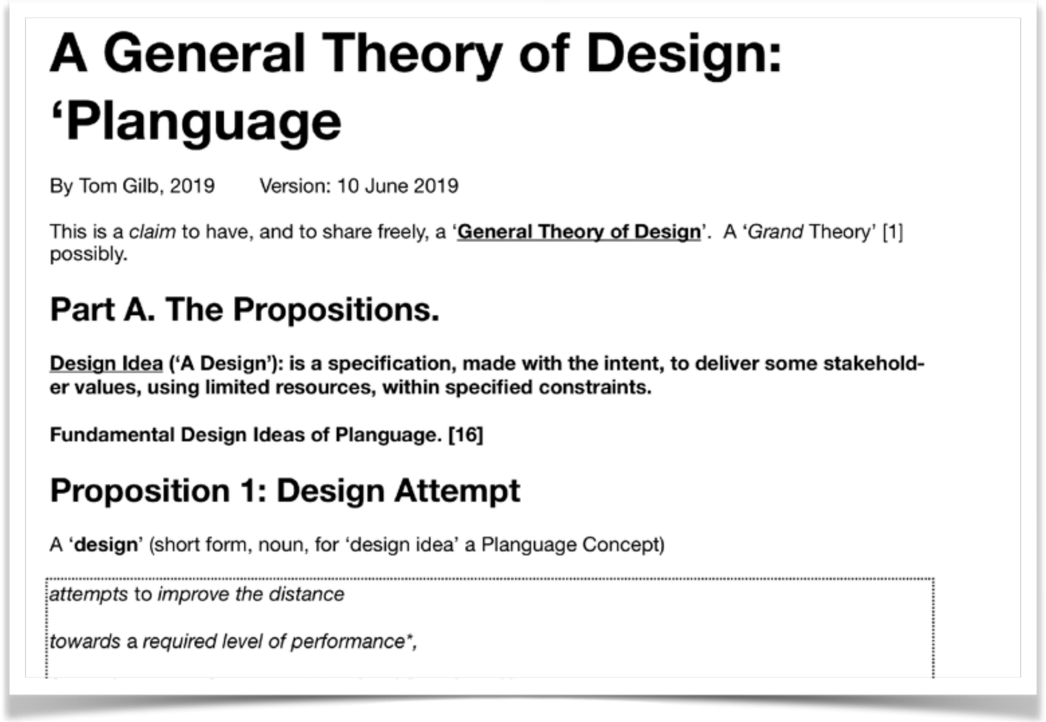


"A General Theory of Design: 'Planguage' "



Get a free e-copy of 'Competitive Engineering' book.
https://www.gilb.com/p/competitive-engineering

(The 'Planguage' design language standard



A General Theory of Design: 'Planguage

By Tom Gilb, 2019 Version: 10 June 2019

This is a *claim* to have, and to share freely, a '**General Theory of Design**'. A '*Grand* Theory' [1] possibly.

Part A. The Propositions.

Design Idea ('A Design'): is a specification, made with the intent, to deliver some stakeholder values, using limited resources, within specified constraints.

Fundamental Design Ideas of Planguage. [16]

Proposition 1: Design Attempt

A '**design**' (short form, noun, for 'design idea' a Planguage Concept)

attempts to *improve the distance*

towards a *required level of performance**,

http://concepts.gilb.com/dl956

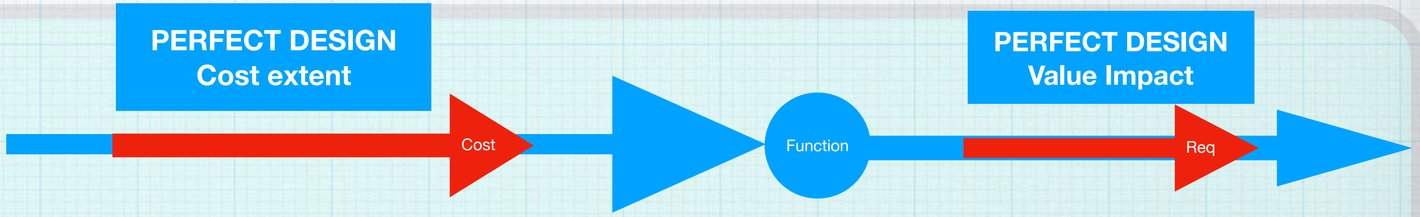
51 Page paper

What is Design ?

A design is

a suggested solution
to the problem specified
by the design requirements.

It might not turn out to be valid, or as efficient as *alternative* solutions.



The Design costs
Do not exceed
Resource constraint,
Budgets

The design impacts
Values
So that all Goal
levels are reached

A perfect valid design specification would

- deliver all value requirement levels within their deadlines
- While not exceeding any resource budgets
- And not exceeding any other specified constraints

'Design' basic concept synonyms:
Architecture, Technology, Means,
Solutions, Strategies

Management has these leadership and organizational roles: with DESIGN

- Identify necessary value and cost ambition levels (requirements)
- Get these ambitions translated into specific actions (design)
- Make sure the designs are brought to life, to deliver the values (results)
- Make sure we see the big picture
  - (all critical values, all critical stakeholders, all critical resources, all available design options)
  - is dealt with.
  - Avoid sub-optimization. (do Systems thinking)

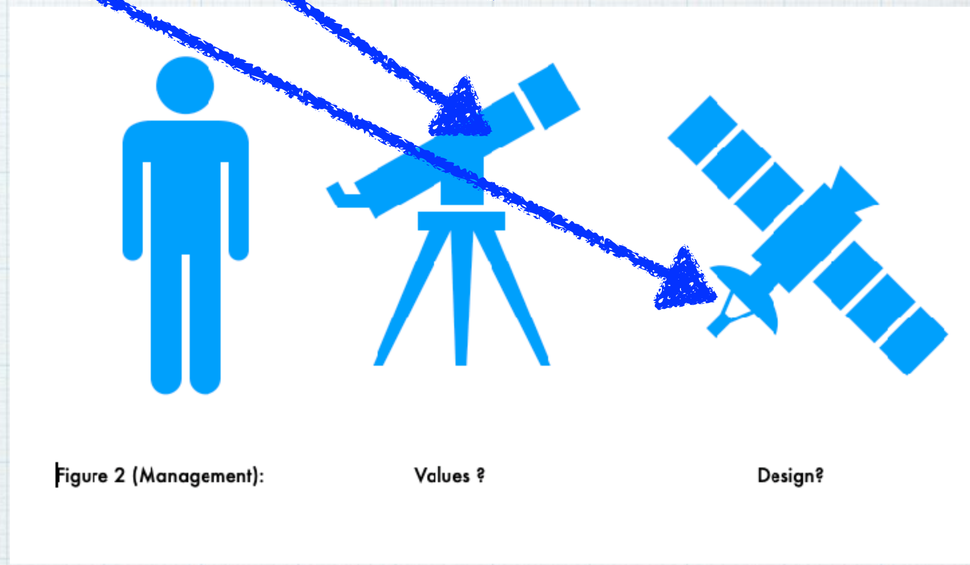


Figure 2 (Management):

Without these management roles being done well,

stuff will happen,

or stay as it is:

but it will not be competitive, improved,

and sooner or later, there will be a need for better management.

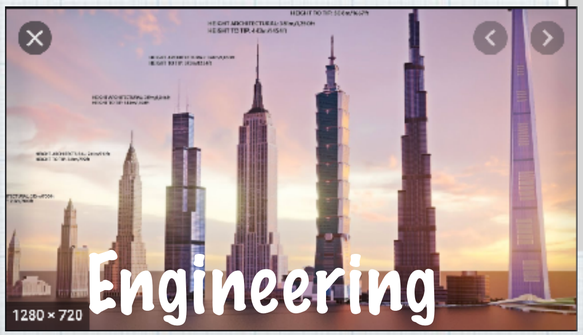
So what are the conditions that need, perhaps demand, a formal design process? Maybe an engineering approach.

- Large size (many people involved)
- Long duration (months, years)
- High capital costs
- Potentially high operational and maintenance costs
- Many stakeholders
- International scope
- Political pressure
- Responsibility, legal, financial, political
- Complexity (many related systems together)
- New technology (you have not used it, few or none have)
- Very high quality requirements (safety, security, availability, usability)
- And many other factors

Table 1 - Project Complexity Formula

Complexity Dimensions	Project Profile		
	Independent	**Moderately Complex**	**Highly Complex**
Solution Clarity Level of IT Complexity	· Solution is readily achievable using existing, well-understood technologies · IT complexity low	· Solution is difficult to achieve or the technology is proven but new to the organization · Moderate IT complexity and legacy integration	· Solution is likely to be using immature, unproven or complex technologies provided by outside vendors · IT complexity & legacy integration high
Requirements Volatility and Risk	· Strong customer/user support · Basic requirements understood, straightforward, stable	· Adequate customer/user support · Basic requirements understood, but are expected to change · Moderately complex functionality	· Inadequate customer/user support · Requirements are poorly understood, volatile, and largely undefined · Highly complex functionality
Strategic Importance Political Implications Multiple Stakeholders	· Strong executive support · No political implications · Straightforward communications	· Adequate executive support · Some direct mission impact · Minor political implications · 2-3 stakeholder groups · Challenging communication and coordination effort	· Mixed/inadequate executive support · Affects core mission · Major political implications · Visible at highest levels of the organization · Multiple stakeholder groups with conflicting expectations

https://www.projecttimes.com/articles/introducing-the-new-project-complexity-model-part-i.htm



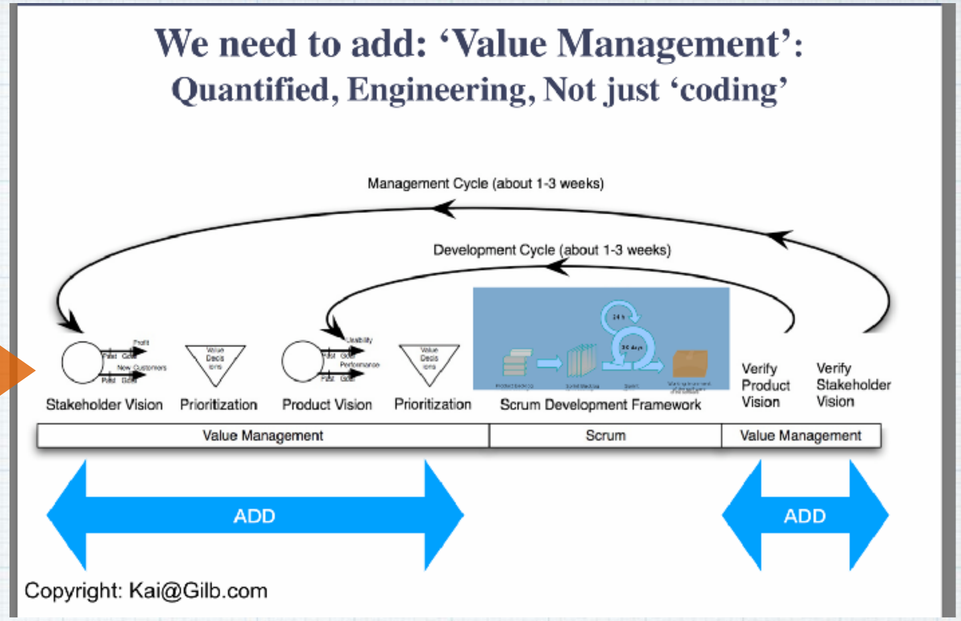
Pseudo Design- Pseudo Architecture
"Calling something by a name does not make it so" (Gerry Weinberg)

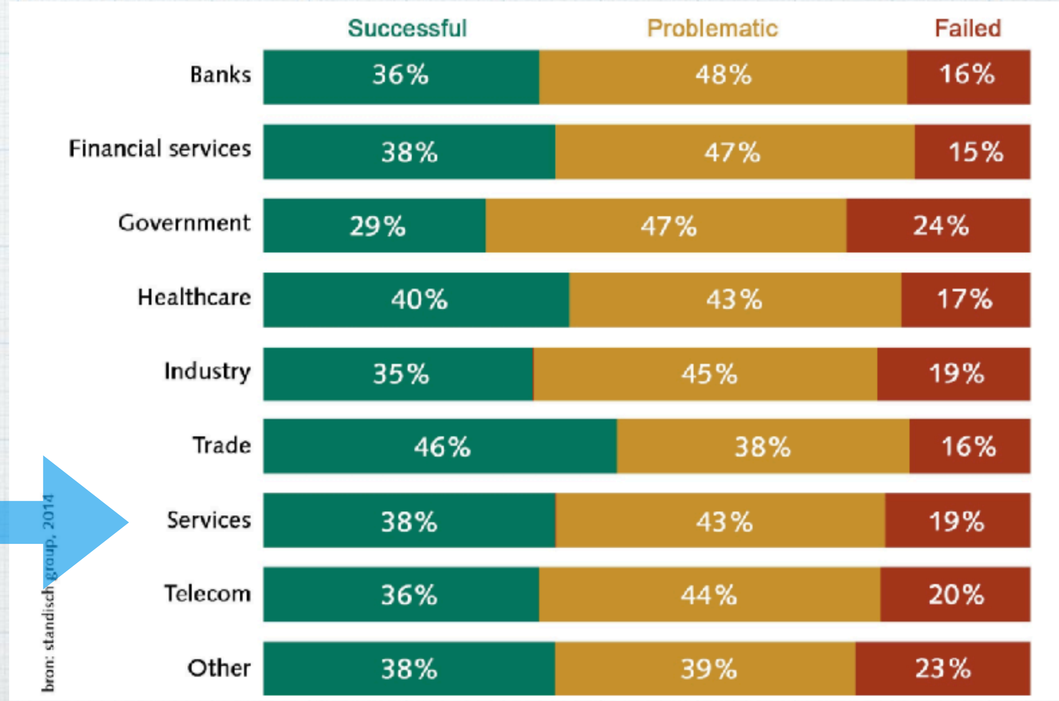
Many IT Architecture processes do spout designs, architecture, and strategies, but they lack:

- Any serious stakeholder analysis (they do 'users')
- Any serious clear and quantified specification of value requirements (including quality requirements), and constraints (including time, money, operational costs, and legality).
- Any serious numeric analysis, of any design suggested; in terms of its impacts, on any of the many values or costs.
- They often have no sense of the larger system, and live in a narrow domain. (like 'dev', 'programming')

* I find it mind-boggling that I see large expensive projects, with these enterprise architects, who have no concern for critical values and costs.
* They do not even feel ashamed or embarrassed.
* Of course the root problem is that *managers* permit these dangerous beasts to exist, and to determine values-and-costs results, blindly.
* We have a very *high project failure rate,* and this is one reason.
* Do not allow these practices, this witchcraft, to persist on your watch. Take responsibility for critical values and costs.

+ Design





Value Design Management. The right stuff.

Value Design Management. The Right Stuff.

So, what does a manager need to do, to manage the design process?

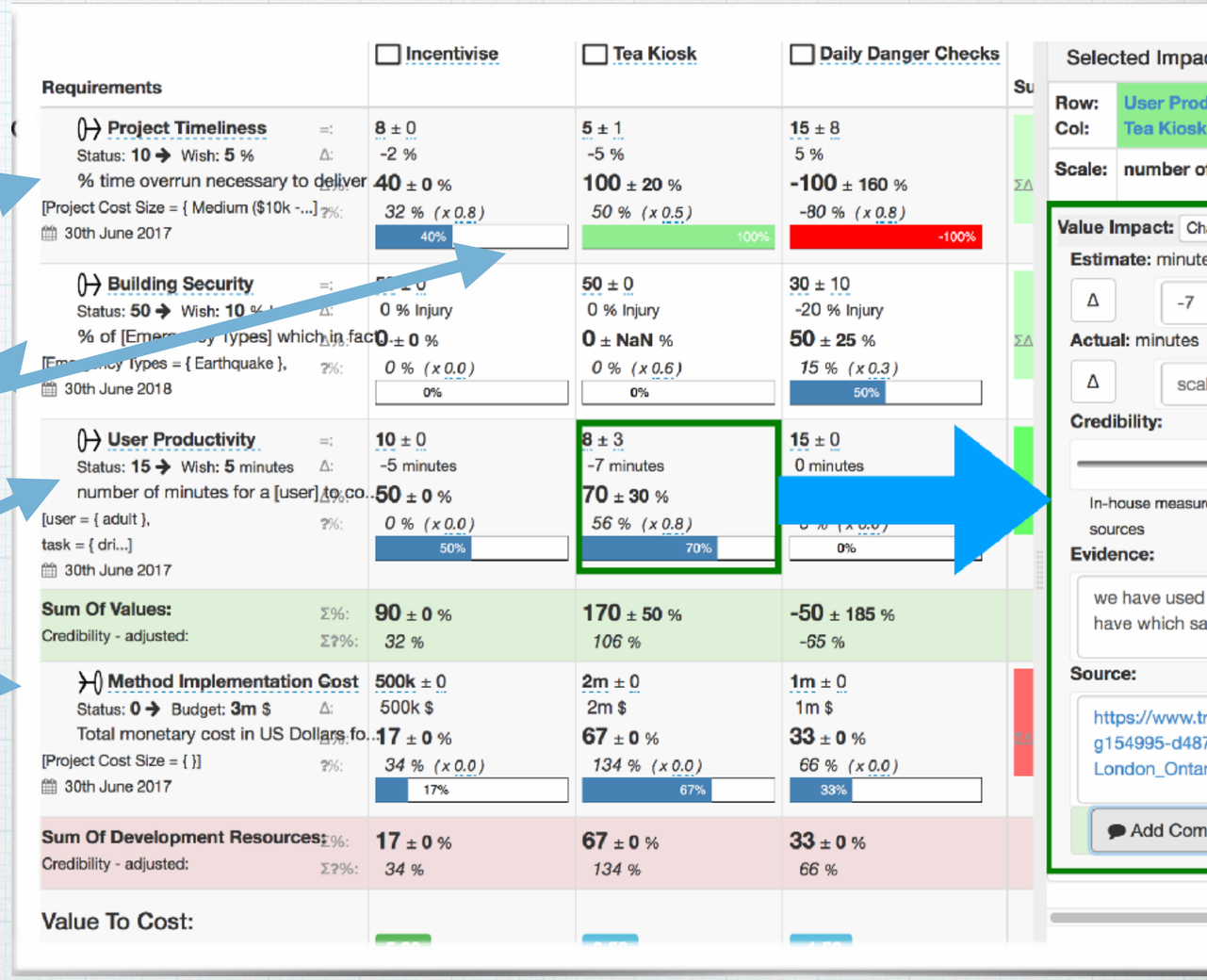
In simple terms:

These pre-design steps were discussed above.

- Make sure reasonably thorough *stakeholder analysis* is done.
- Make sure Critical 'design' *requirements*, for values and resources are specified: as a minimum, that they are all *quantified*.

Now we are ready for the 'design' steps

- Make sure that all design *options are estimated*, before choice and prioritization
  - The main asserted critical design *value impact is estimated*
  - All other critical values (the other 9 in top 10) are estimated for the impact of the design (*side-effects*)
  - All budgeted *resources* impacts (time, money, operational costs) are *estimated*
  - All designs evaluated for all stated legal/cultural/contractural/stakeholder *constraints*.
  - All estimates of value and cost with respect to ± *uncertainty range*
  - All estimates have *evidence* stated, and rated (credibility level)
  - All designs and estimates specs are *quality controlled* (Spec QC)



One approach to this, is to keep on asking the Tough Questions.

"So you tell me this strategy is more secure?

* Can you give me a number for experience,
* you can cite,
* of how secure it is?"

* "To what degree would your design or architecture
* be sure to meet our numeric Goal levels,
* within our short deadlines and miserly budgets?"

If you know everything, then you can tell me what you do not know?

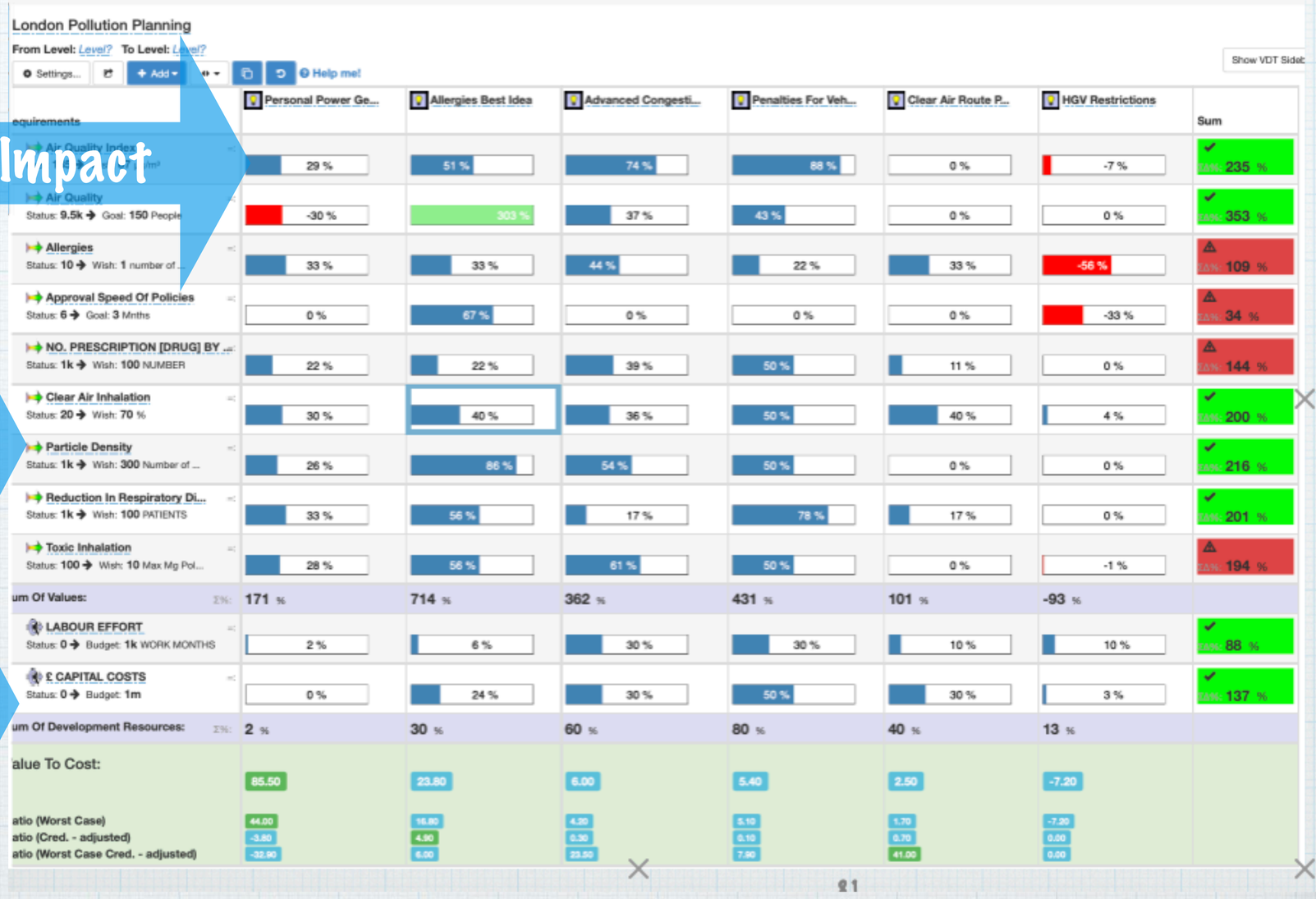
Reply to my 6 year old grandson (He 'got' it)

'12 Tough questions questions' paper. http://www.gilb.com/dl24

see also 20 Tough Questions 2016, http://concepts.gilb.com/dl876

Principles for design

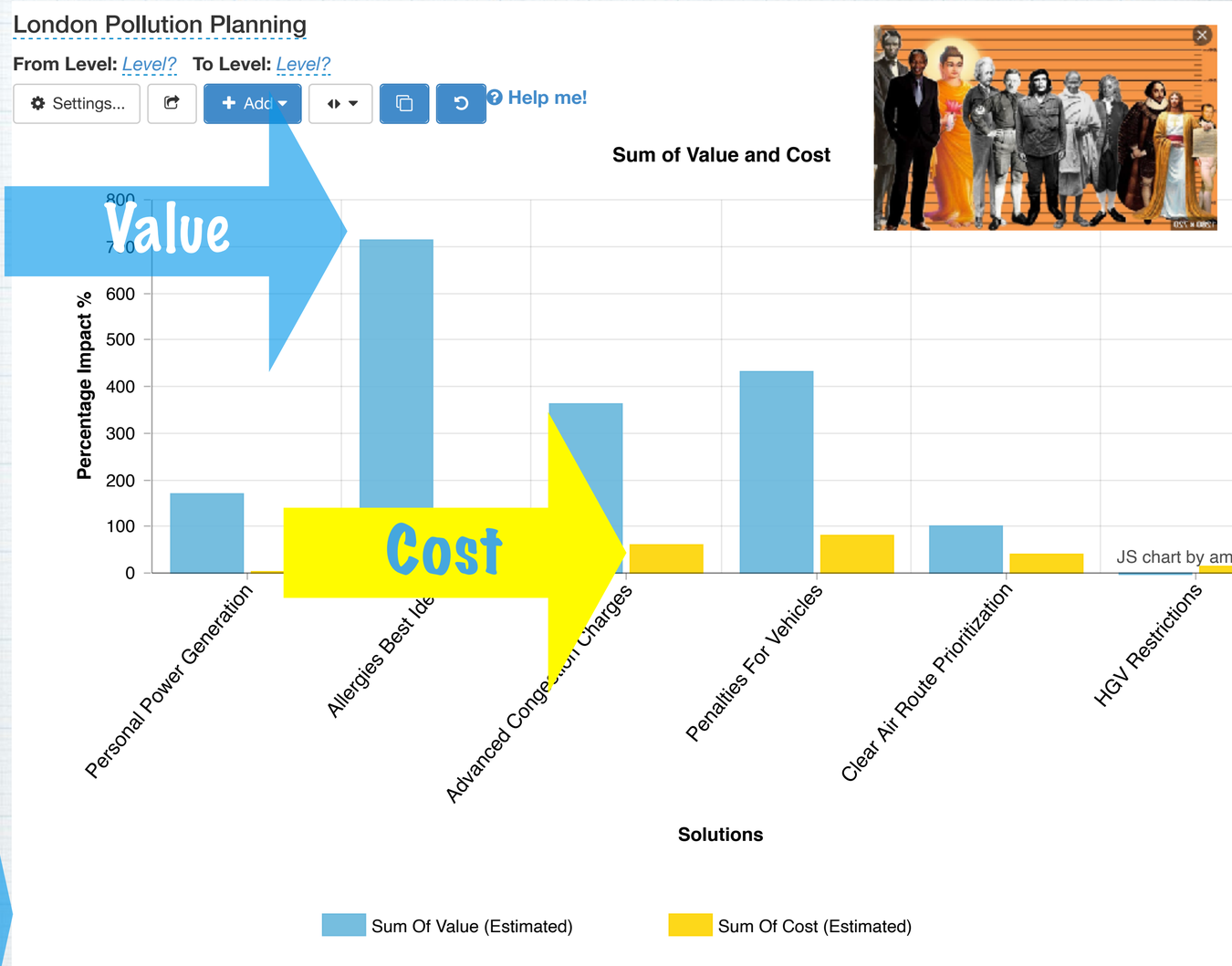
1. Your designs must contribute substantially to your value objectives at low costs: estimate and measure the levels
2. If you *try out your design ideas in small increments*, you can adjust designs, and never fail, on a large scale.
3. All designs have at least 9 side-effects on your critical values, and at least 6 cost aspects, some very negative; so you need to try to *discover these, as soon as you can, estimate, then measure their delivery incrementally.*
4. Your designs need to be *tried out in practice, in small increments*, so if they disappoint, you can *dump them fast*.



Policies for design: management attitude.
This should be obvious. I expect you to do this. I will ask about it,

1. Design's *impacts* will be estimated *before* their selection.
2. Designs will be tested in *practice*, *before* keeping them in place.
3. The preferred designs will be those with *'high values' and 'low costs'*.
4. Our designers will *justify* their design suggestions with numeric *facts*, and *evidence*, and practical demonstrations.
5. The name of the sponsoring designer/ *architect*/strategy sponsor will be annotated and *public*.

From Table Data



Checklists for design: feel guilty if you cannot say yes!

1. Is each design idea specified in *enough detail,* to enable us to *understand* its value and cost ranges ?

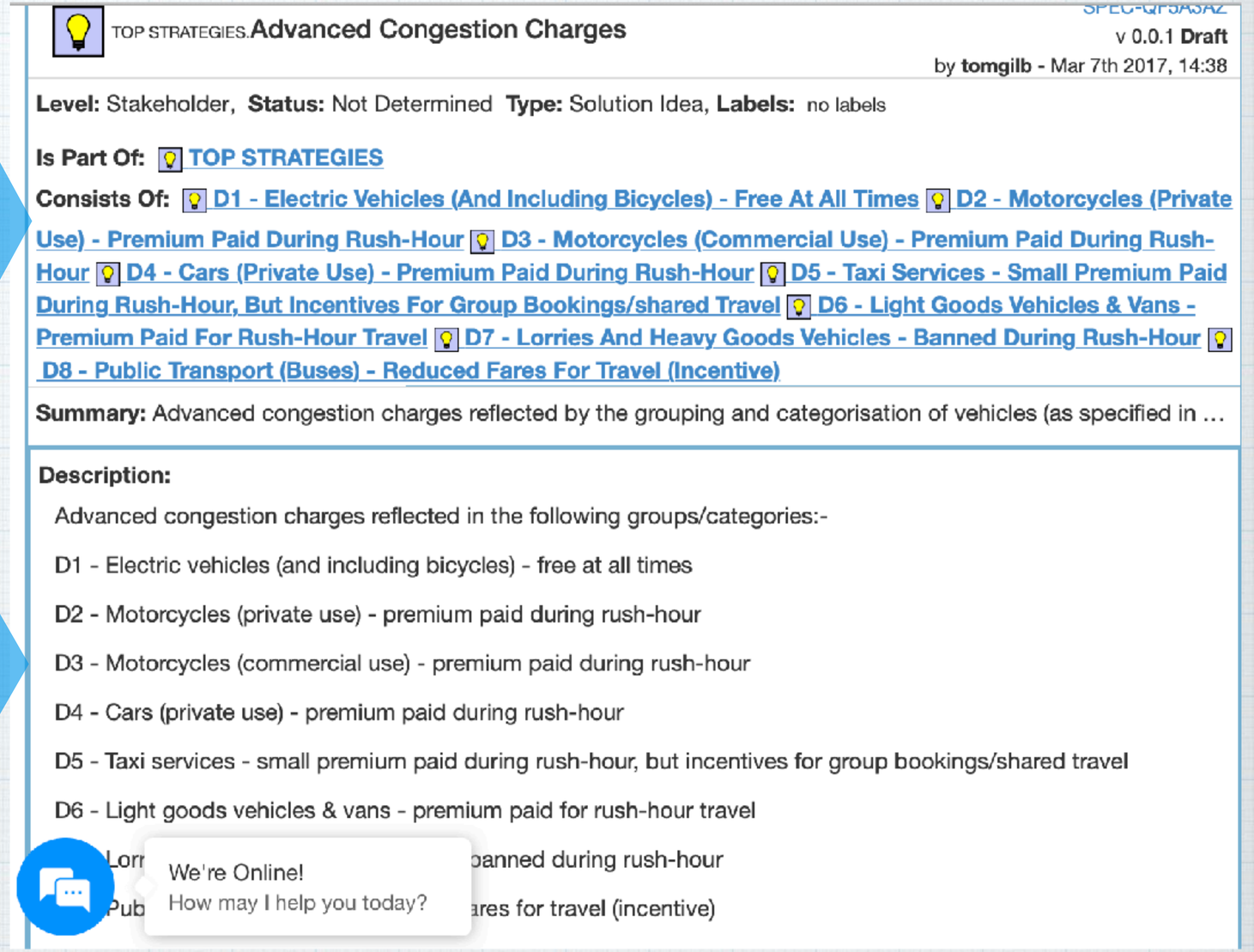
Detailed design links

2. Are there any estimates, with *evidence*, for the design values and costs.

3. Are large designs *decomposed*, into *small implementable* increments-to-existing systems?

Decomposed

4. Who exactly is *name-responsible* for the *failure* of any design/architecture/strategy?



End Part 2
Design Management

Part 3.
Managing quality assurance.

Some Basic QA Definitions.

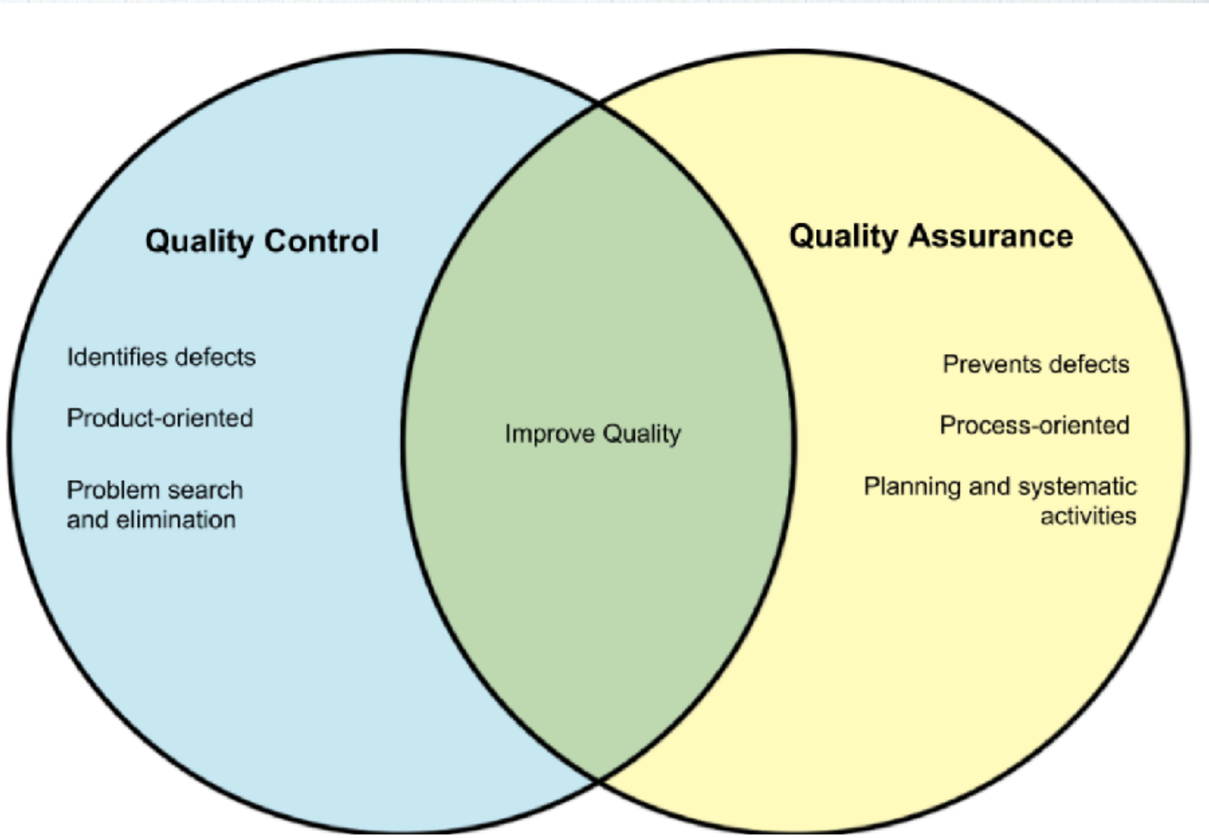
Quality: ***How Well*** a function functions. Often ending in '-ility'

Quality Assurance (QA): any process that contributes to meeting and maintaining **quality levels,** including all value levels.

* The primary emphasis should be on ***prevention*** of lower qualities than planned, not merely on detection and correction.
* Typically these are organizational and technical improvement processes, so that work **processes** lead to better quality.
* For example QA is engineering, architecture, design, process improvement. Avoid misuse of this term to simply mean 'testing' .

Quality Control (QC): checking any type of quality level, and related factors to make sure they are OK.

* **Removal and correction** of bad quality artifacts.
* Typically QC is **reviews** and **testing** processes.

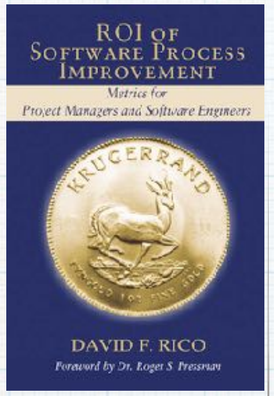


https://whyunlike.com/quality-control-vs-quality-assurance/

Designing Your Own QA Process

<u>The Organizational Improvement Process.</u>

1. Define your *organizational improvement objectives quantitatively*. Improve definition periodically
2. Find candidate *organizational QA strategies*, and estimate their *effectiveness, and costs.*
3. *Decompose* big strategies, into smaller implementable strategies, and *try them out* in practice. Keep if good. Modify if necessary
4. Continue this process until all your organizational *value objectives are reached.*



http://davidfrico.com

Source: Ericsson Case

Improvement ROI:
"Engineering Process Improvement Profitability"
Ambition: Order of magnitude return on investment in process improvement.

Scale:

The average [annual OR defined time term] Return on Investment in Continuous Improvement as a ratio of [Engineering Hours OR Money]

Note: The point of having this objective is to remind us to think in terms of real results for our process improvement effort, and to remind us to prioritize efforts which give high ROI. Finally, to compare our results to others. <-TsG

Record

[Shell NL, Texas Instruments , Inspections] 30:1 <- Independently published papers TsG

Past

[IBM RTP, 1995, DPP Process] 13:1 <- Robert Mays, Wash DC test conference slides TsG

[Raytheon, 1993-5, Inspection & DPP] $7.70:1 <- RDE Report page 51 ($4.48 M$0.58M) Includes detail on how calculated. PK has copy.

[IBM STL, early 1990's] Average 1100% ROI (11:1) <- IBM Secrets pp32. PK has copy. NB Conservative estimate. See Note IBM ROI below.

Goal Next Year. 10:1 ROI

ERICSSON CASE: Software Engineering Productivity

* These objectives (Software Productivity ... Profitability)
  * were the CTO support objectives
  * of the higher-level corporate objectives (Profit and survival).

* The CTO objectives were themselves supported by a set of 'Means Objectives'

* (see ref gilb.com/dl559 for detail, including real detailed quantification of all these values)

Ericsson of Sweden, Mobile Base Stations needed organizational improvement, so that they could produce more product faster, for an eager international marketplace.

This involved many simultaneous critical value improvements, including quality and productivity. Both highly related.

Support
the **Fundamental** Objectives (Profit, survival)
Software Productivity:
Lines of Code Generation Ability
Lead-Time:
Predictability.
TTMP: Predictability of Time To Market:
Product Attributes:
Customer Satisfaction:
Profitability:

See slide 25

Productivity Slides incl Ericsson

http://www.gilb.com/dl559

Specific Strategy Options (only the best survive)

We only prioritised and implemented strategies with a proven evidence-based record

Strategies: (total brainstormed list)
'Ends for delivering Strategic Objectives'

Evo [Product development]:

DPP [Product Development Process]: Defect Prevention Process.

Inspection?

Motivation.Stress-Management-AOL

Motivation.Carrot

DBS

Automated Code Generation

Requirement -Tracability

Competence Management

Delete-Unnecessary -Documents

Manager Reward:?

Team Ownership:?

Manager Ownership:?

Training:?

Clear Common Objectives:?

Application Engineering area:

Brainstormed List (not evaluated or prioritized yet)?

Requirements Engineering:

Brainstormed Suggestions?

Engineering Planning:

Process Best Practices:

Brainstormed Suggestions?

Push Button Deployment:

Architecture Best Practices:

Stabilization:

World-wide Co-operation?

The Decision

- Three specific processes were selected based on credible cost-effectiveness,
- and we got, in the CTO presentation meeting,
- acceptance to start using 2 of them, the _.

The One Page Top Management Summary
(after 2 weeks planning)

The Dominant Goal

Improve Software Productivity in R PROJECT by 2X by

Next 3 Years

Dominant (META) Strategies

Continual Improvement (PDSA Cycles)
.DPP: Defect Prevention Process
.EVO: Evolutionary Project Management

Long Term Goal

Next 3 Years

DPP/EVO, Master them and Spread them on priority basis.

Short Term Goal [Next Weeks]

DPP [RS?]
EVO [Package C ?]

Decision: {Go, Fund, Support}

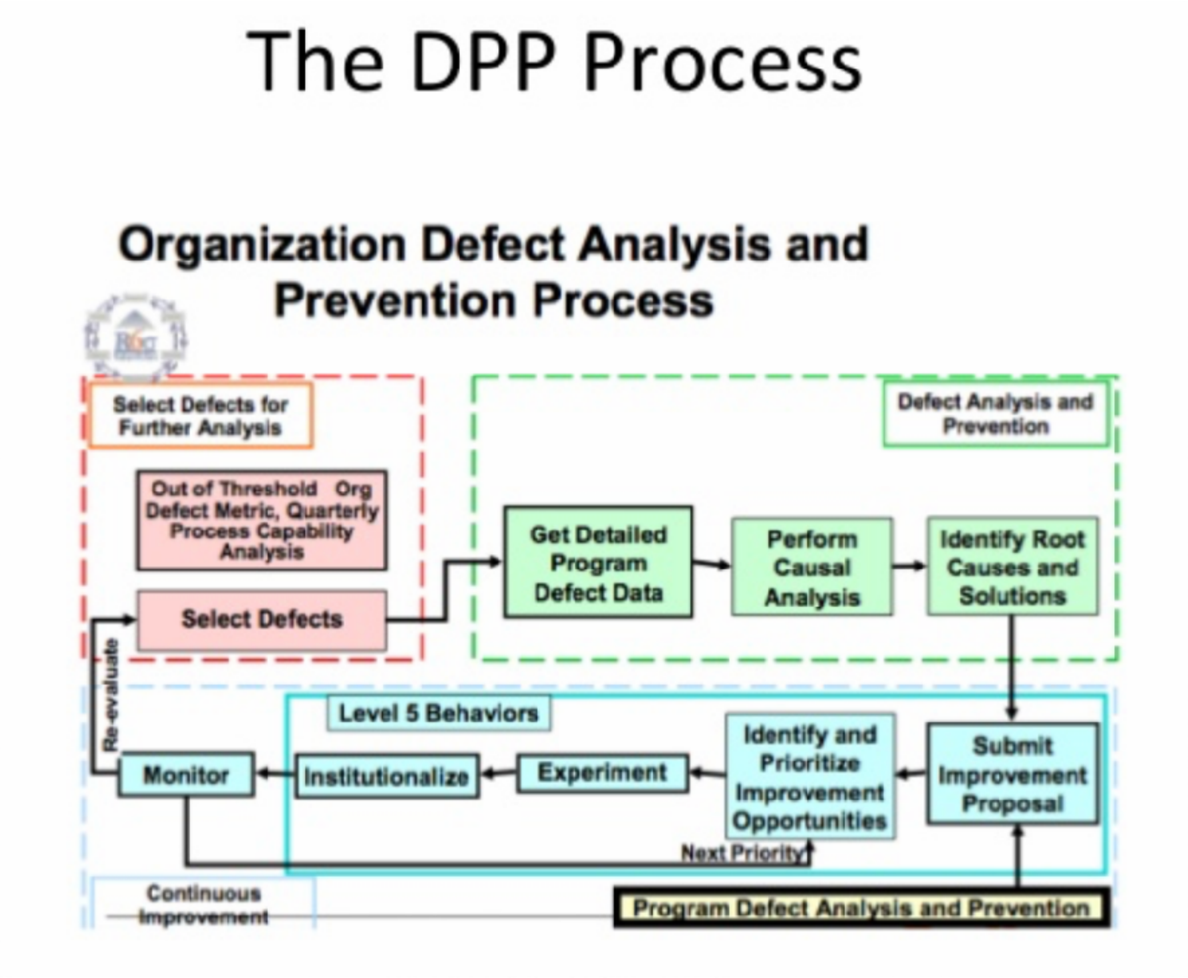
The DPP, Defect Prevention Process
Raytheon Case (E)

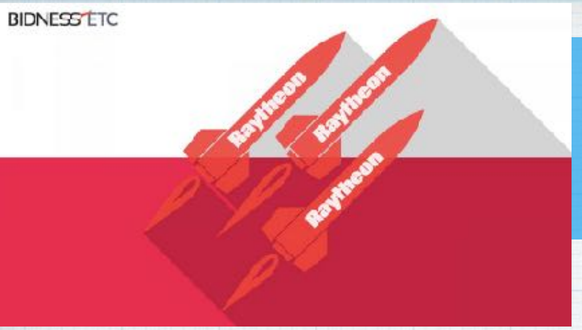
DPP is simple:

- Grass-roots professionals analyze their own everyday faults and problems
  - (not analyzed by managers or consultants)
- They suggest changes to prevent the problems re-occurring
- They can try out the suggested changes, before scaling up
- The process is measured against current critical value improvement objectives
  - (qualities of product and services)

The key DPP successful idea was

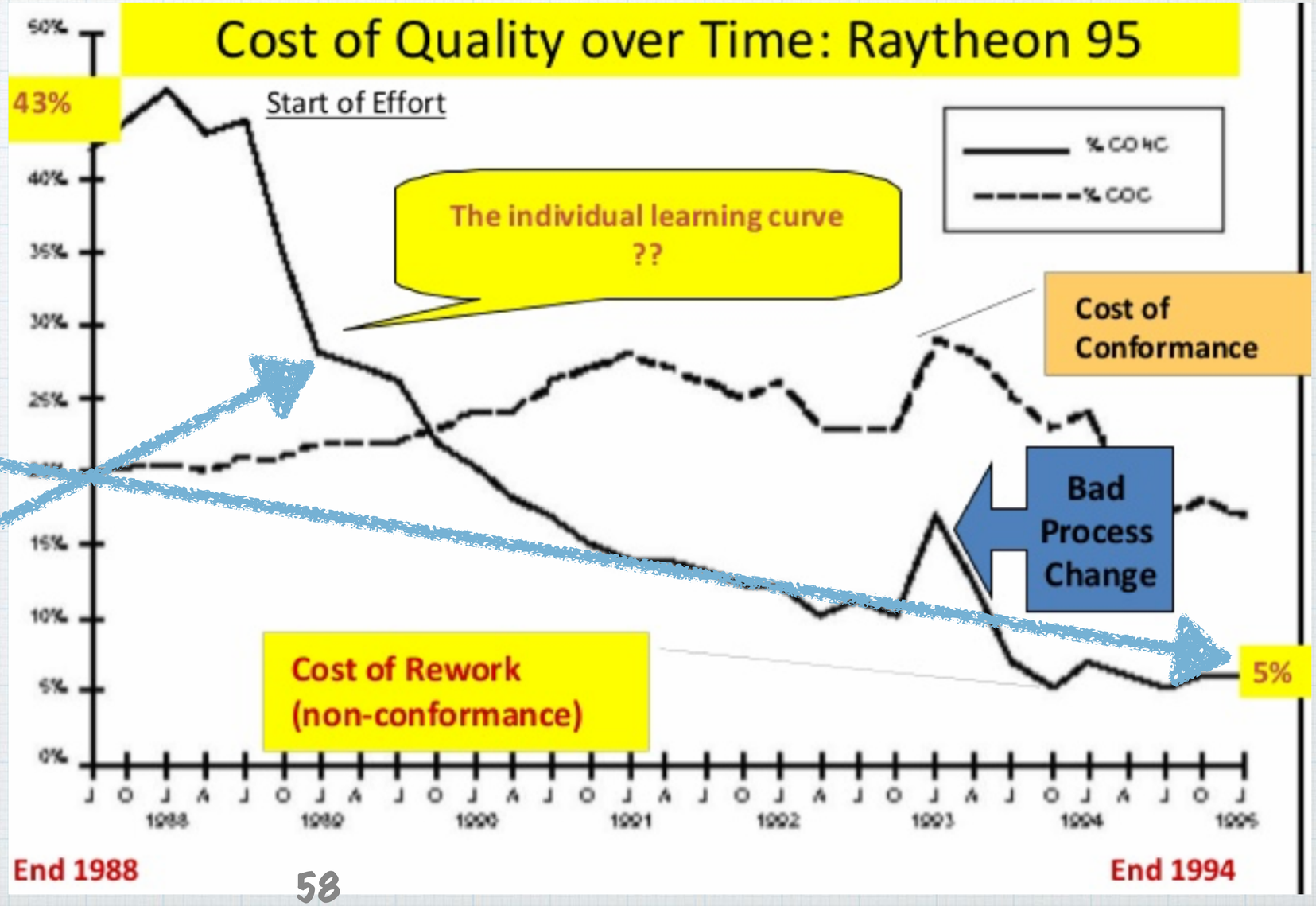
- *delegation of power* to analyze
- and be creative,
- delegation to the troops,
- not to top management
- and their consultant corporations
  - (who always have a big idea, that fails).





DPP & QC, Reduced Rework by 10X
For 1,000 Software Engineers

- **Cost of Bad Quality Reduced**
- **As a result of DPP&QC,**
- **the wasted effort, 'Cost of Rework',**
- **went down 10X**
  - **In 8 years, &**
  - **By 50% in 1 year**
- **And there are more results .... Next slides**



Raytheon Paper (2019 link)
https://figshare.com/articles/Raytheon_Electronic_Systems_Experience_in_Software_Process_Improvement/6582863

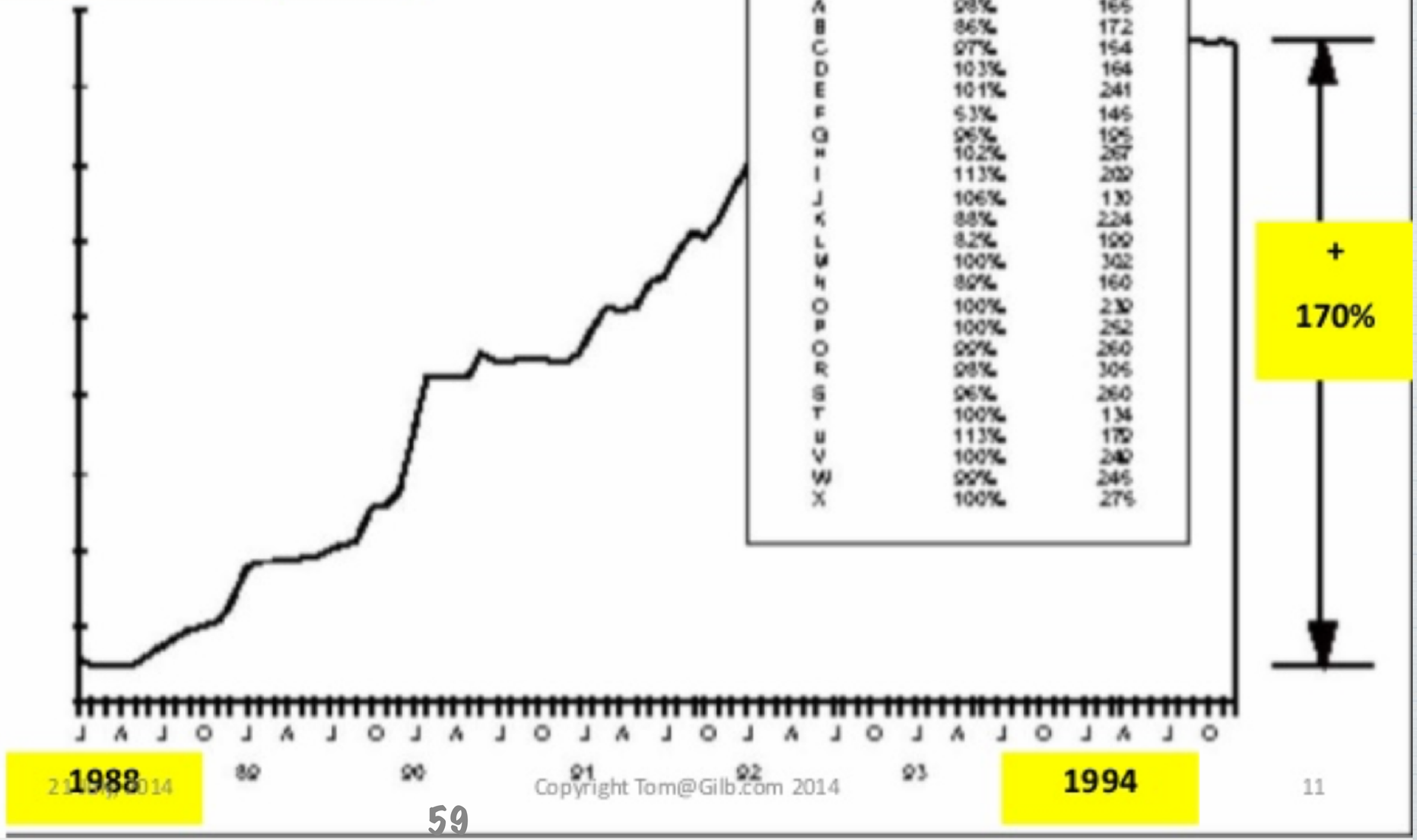
Software Productivity x 2.7 (Programmers produce more code)

- 'Productivity'(lines of code).
  - as a result of DPP, 'productivity' went up by average 2.7 X
  - Based on 24 projects. (A-X)
- Because much less re-work, of their own errors.
  - Errors prevented
  - 'Lean' (D P P)
  - Corporate Learning in practice

Software Productivity 2.7X better

Productivity

Project	CoCoMo II	Productivity
A	98%	166
B	86%	172
C	97%	154
D	103%	164
E	101%	241
F	53%	146
G	96%	195
H	102%	267
I	113%	202
J	106%	130
K	88%	224
L	82%	199
M	100%	302
N	89%	160
O	100%	230
P	100%	252
Q	99%	260
R	98%	305
S	96%	260
T	100%	134
U	113%	179
V	100%	240
W	99%	246
X	100%	276

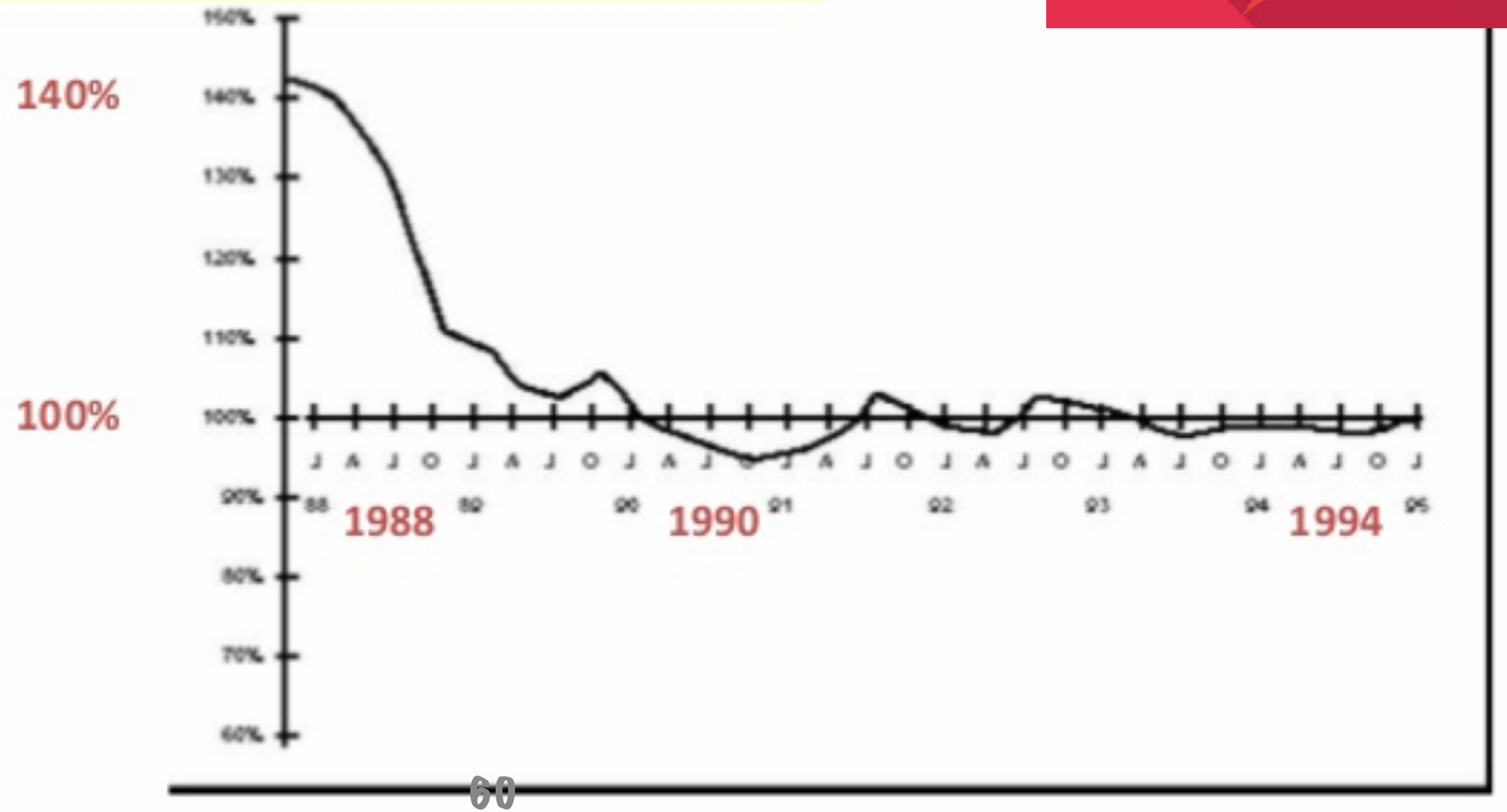


Project Predictability: being 'On Budget'

- 'Predictability'.
- As a result of DPP, the ability to deliver on budget
  - got much better.
- Why?
  - Reduced errors (errors caused by bad corporate processes).
  - (1,000 programmers)
  - Less need to clean up bad soft-crafting

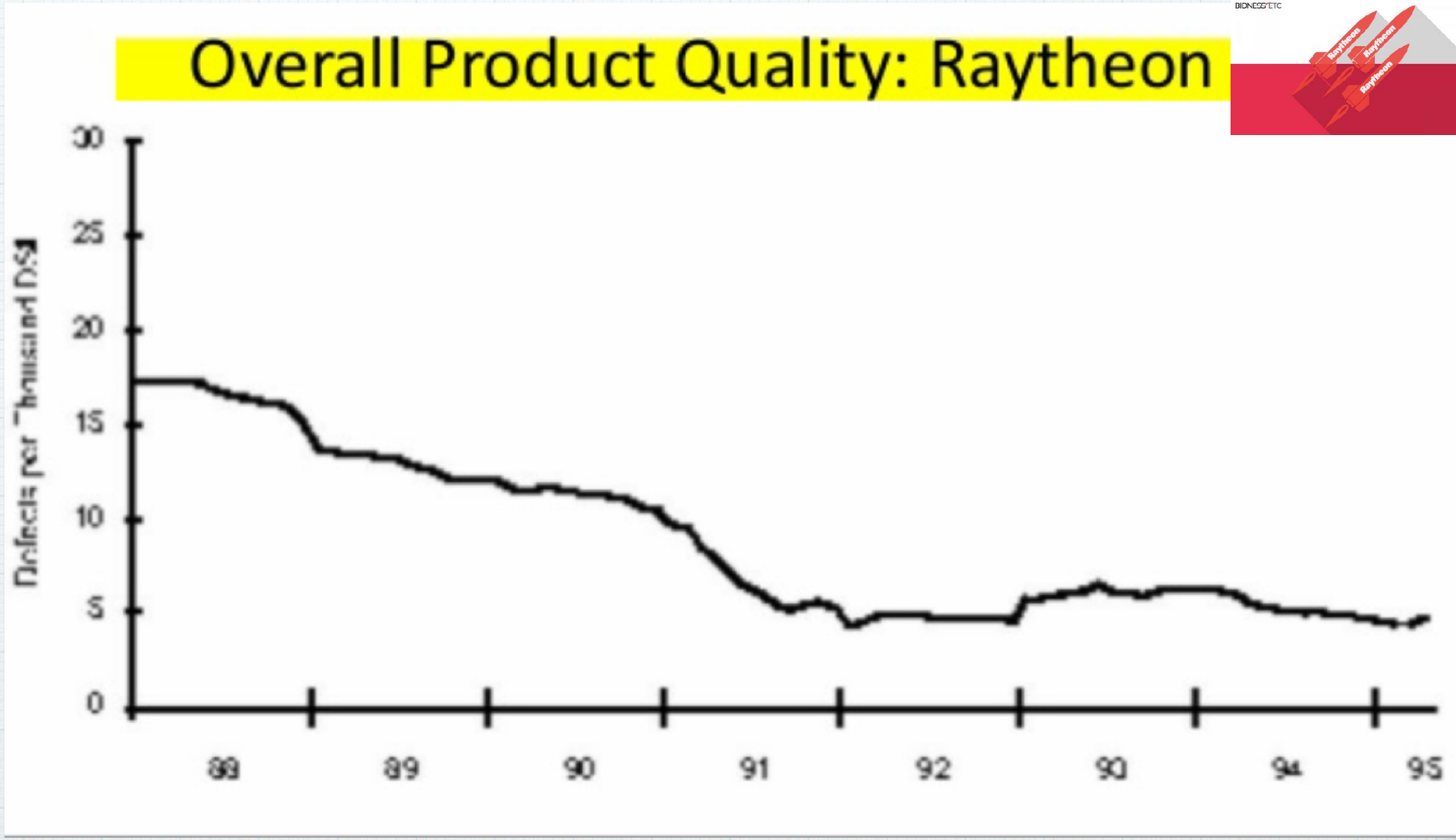
Achieving Project Predictability: Raytheon

Cost At Completion / Budget %



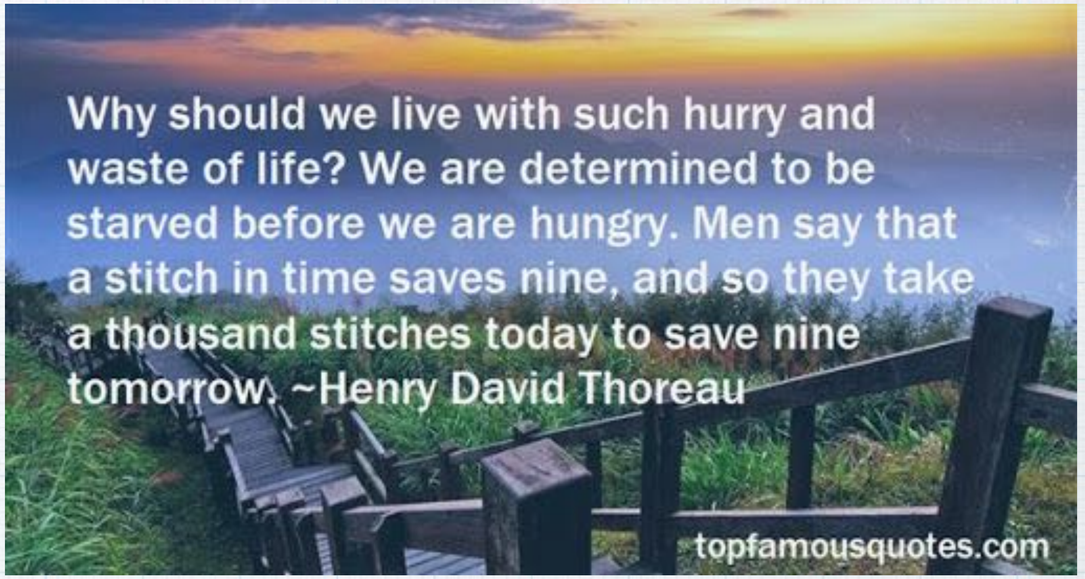
Software Bug Density, went way down

- 'Product Quality' (bug density).
- As a result of DPP the software bug released rate
  - went down substantially.
- Because: root causes of bugs were detected (QC and Test) and removed from official processes (DPP)



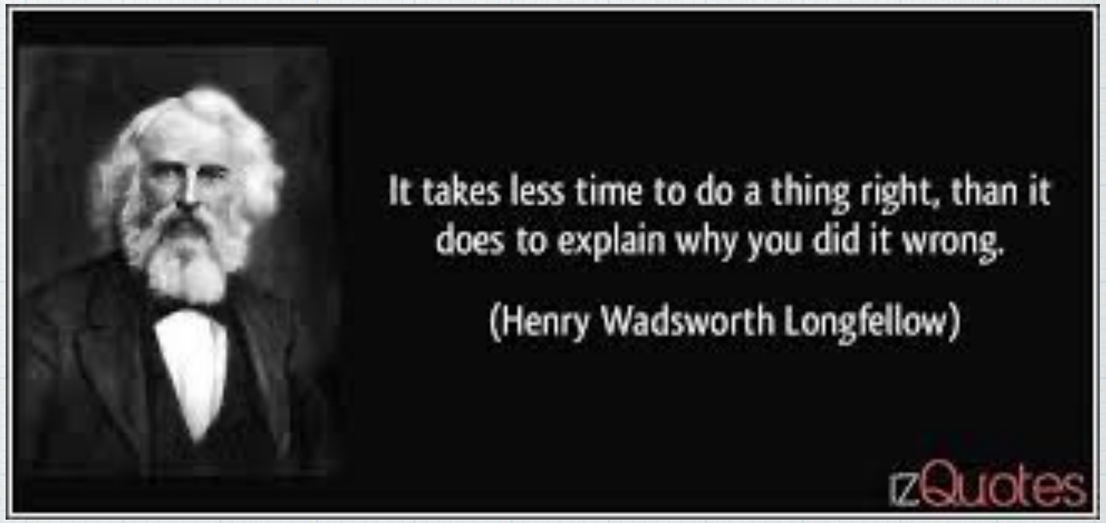
QA Principles: The Henry Principles

1. *A stitch in time, saves nine'*. It pays off to tackle problems upstream, easily. And to prevent bad stuff, rather than to detect it late.



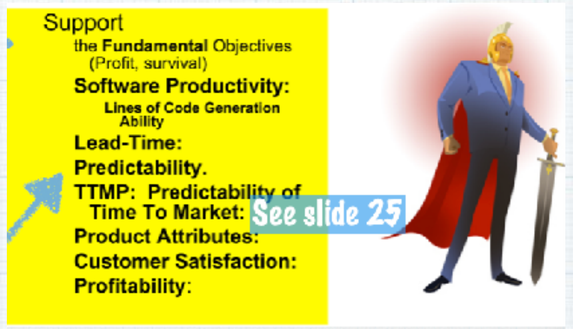
2. QA improvement is *very many small* organizational tweaks, which need to be *incremented gradually*.
   - The 'big idea' is to manage this process (like with DPP) of a *stream of incremental improvements*.

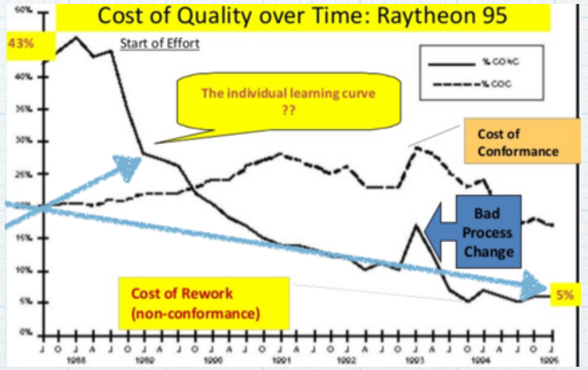
3. QA is *not* about 'testing',
   - it is about *doing things right the first time;*
   - not just as an individual,
   - but because the *organization* makes that possible,
   - enables good work, for all people.

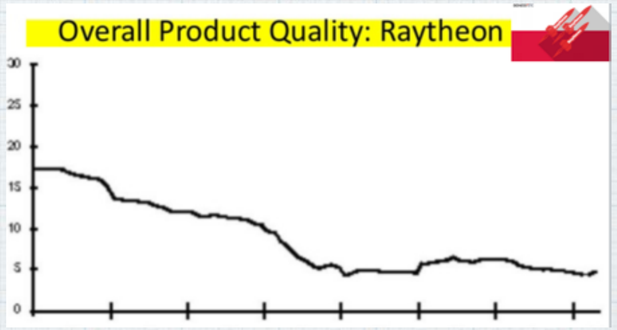


QA Policies, for managers: your attitude

- We will guide Quality Assurance tactics
  - by means of our top ten critical numeric value objectives.
  - Our definition of our quality values. 'Values Assurance' !
- We will make good use of our grass roots professionals
  - to *find* process improvements for QA,
  - and to *try* them out initially.
- We will base our QA improvement efforts
  - on a long-term process of systematically getting better,
  - and *measuring* that we got better,
  - and making sure the new methods are *really embedded* in the organization widely, and for the long term.

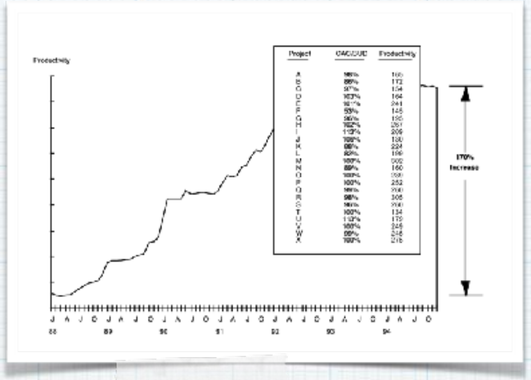


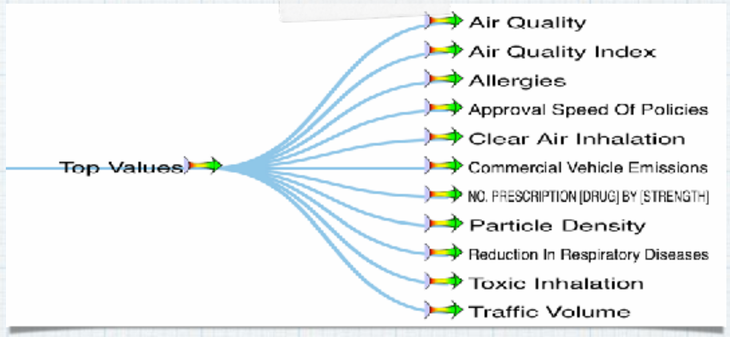


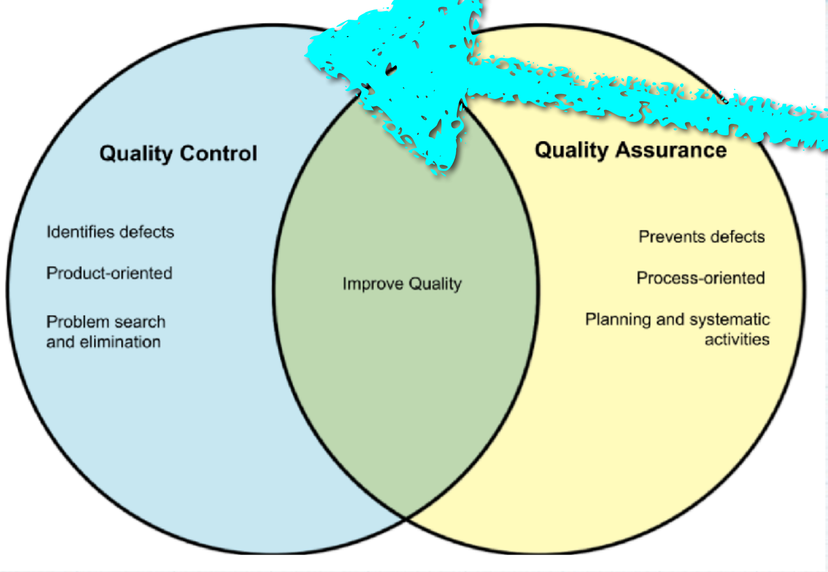


QA Checklists: have you done this yet?

- Can you trace a long-term improvement curve
  - towards your critical multiple *organizational* quality values?
- Is CTO Level management onboard with
  - *quantification* of critical project and product objectives for quality values?
  - Is there long-term sustained support and real action, to reach the goals?
- Do you have a *high rate* of embedding the many small organizational improvements,
  - like at least one a week?







The 'Spec-Reviewing' QC Process

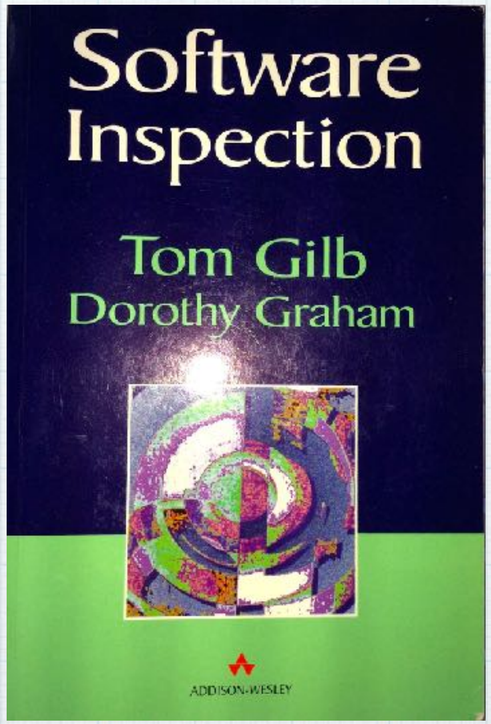
a subset of a QA Process.

* Most review processes I know about are weak to worthless.

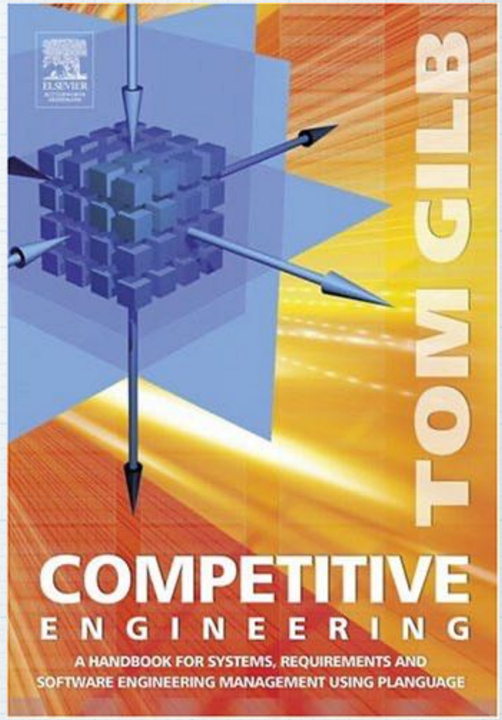
* But a few are very effective, and measurably so. I am going to focus on our best known single process (SQC), but if you find better ones, let me know.

* We call it Specification Quality Control (Spec QC).

* It works well because
  * it is simple,
  * it is quantitative,
  * and it can be as powerful as you want to make it.



1993



2005

https://www.gilb.com/lean-specqc
Online Training. Kai Gilb

Clarity, then Relevance

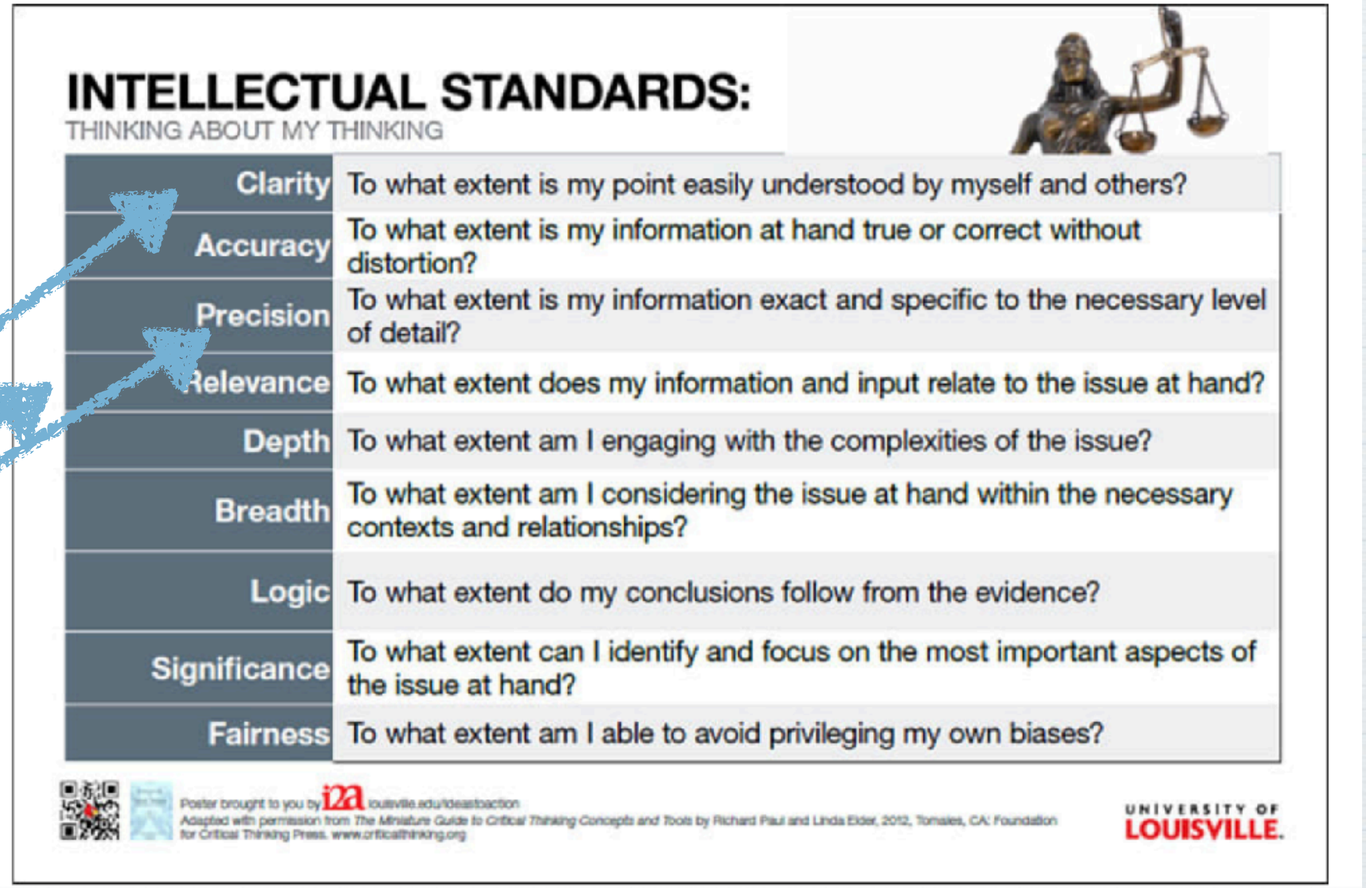
- Spec QC can be used to measure any written plans,
  - and is especially useful on critical things
  - like requirements. (and contracts, designs, test plans, software code, everything)

- It has two basic modes:
  - Checking that something is *clear*
  - Checking that something is 'right'

- The first test, 'clear', is a *prerequisite* for checking that the clear stuff
  - is also *correct*; consistent with the facts and truth.

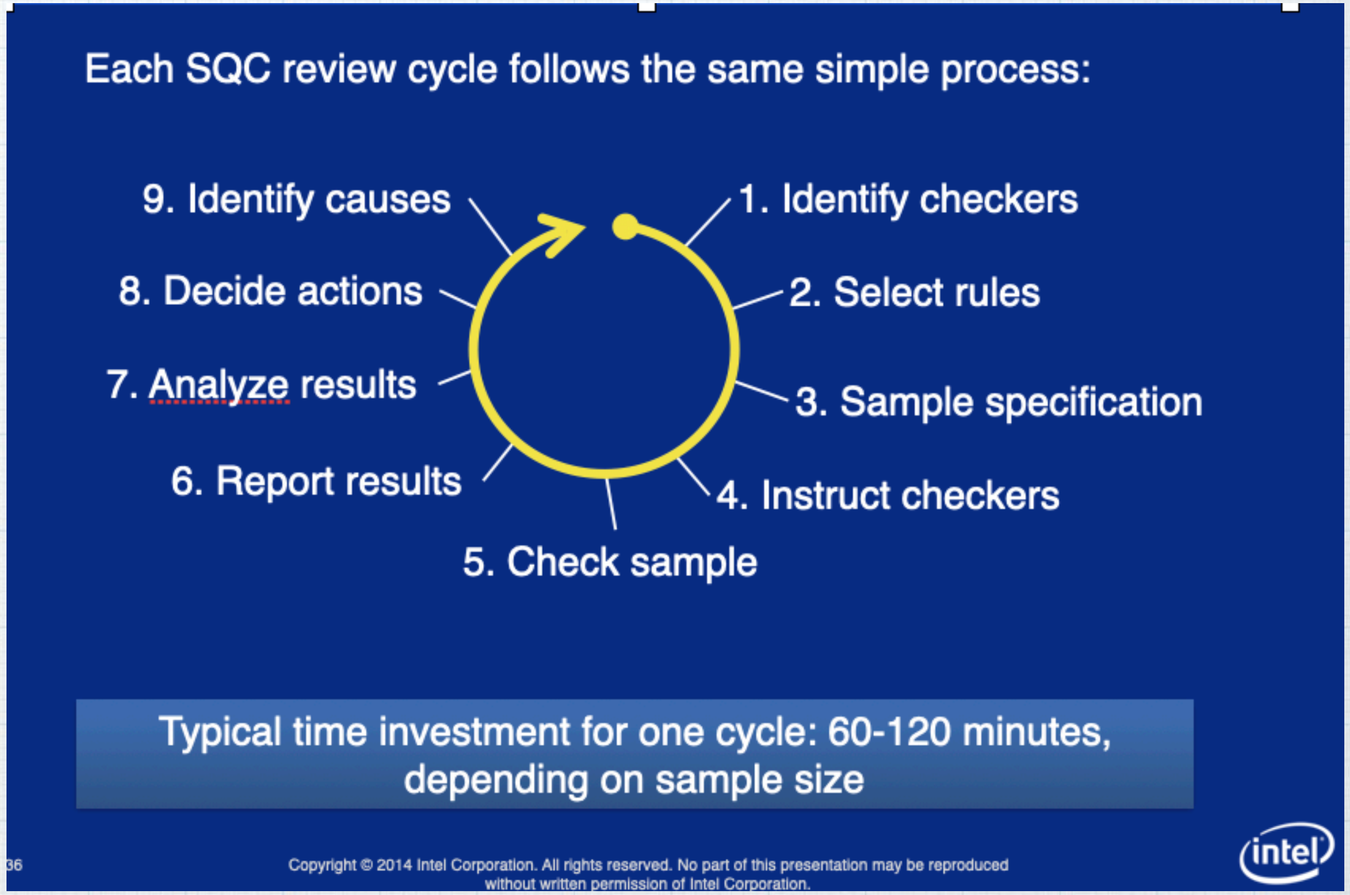
dentalcare.com/en-us/professional-education/ce-courses/ce551/the-intellectual-standards

Table 4. Intellectual Standards.



Basic 'Spec QC'

- The basic SQC process is simple,
  - we sample a document,
  - and count the 'specification defects' (Rule violations) we find.
- This gives us a measure of pollution
  - (of not following our organizations own rules).
  - For example if your one Rule is 'must be clear', then all unclear words are counted as 'defects'.
- If there are 'too many' defects; we have to do something.



Spec QC Process

Clear? Right?

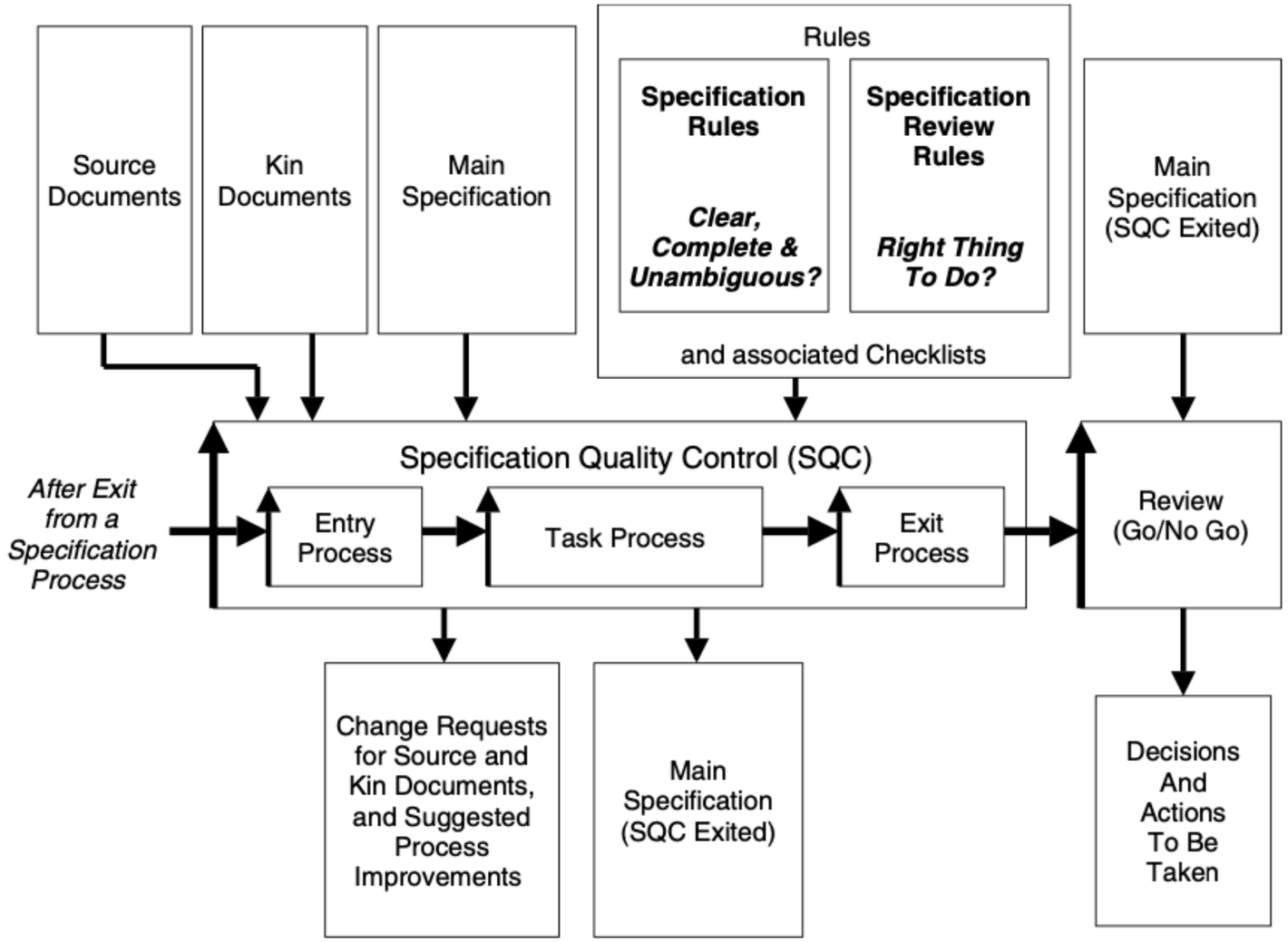
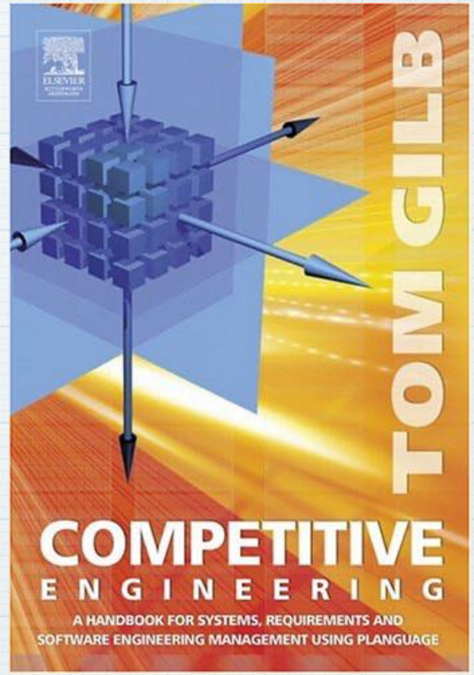


Figure 8.8
Overview of the SQC process showing how Specification Review Rules fit alongside Specification Rules.



http://www.gilb.com/dl541

Intel Measures of Gilb Methods 2013

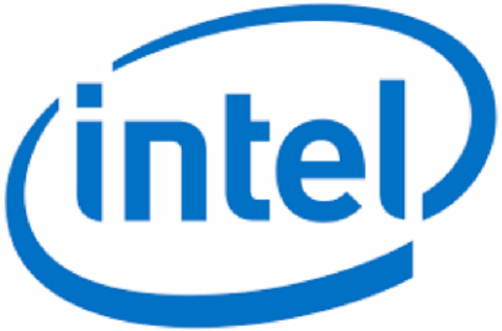


TABLE I: GEN 2 REQUIREMENTS DEFECT DENSITY

PRD Revision	# of Defects	# of Pages	Defects/ Page (DPP)	% Change in DPP
0.3	312	31	10.06	-
0.5	209	44	4.75	-53%
0.6	247	60	4.12	-13%
0.7	114	33	3.45	-16%
0.8	45	38	1.18	-66%
1.0	10	45	0.22	-81%
Overall % change in DPP revision 0.3 to 1.0: ***-98%***				

The Impact of Requirements on Software Quality across Three Product Generations

John Terzakis

Intel Corporation, USA
john.terzakis@intel.com

***Abstract*—In a previous case study, we presented data demonstrating the impact that a well-written and well-reviewed set of requirements had on software defects and other quality indicators between two generations of an Intel product. The first generation was coded from an unorganized collection of requirements that were reviewed infrequently and informally. In contrast, the second was developed based on a set of requirements stored in a Requirements Management database and formally reviewed at each revision. Quality indicators for the second software product all improved dramatically even with the increased complexity of the newer product. This paper will recap that study and then present data from a subsequent Intel case study revealing that quality enhancements continued on the third generation of the product. The third generation software was designed and coded using the final set of requirements from the second version as a starting point. Key product differentiators included changes to operate with a new Intel processor, the introduction of new hardware platforms and the addition of approximately fifty new features. Software development methodologies were nearly identical, with only the change to a continuous build process for source code check-in added. Despite the enhanced functionality and complexity in the third generation software, requirements defects, software defects, software sightings, feature commit vs. delivery (feature variance),**

ays from project
he second to the

, requirements
quality, multi-

ort paper [1] that

II. Product Backgrounds

The requirements for Gen 1 that existed were scattered across a variety of documents, spreadsheets, emails and web sites and lacked a consistent syntax. They were under lax revision and change control, which made determining the most current set of requirements challenging. There was no overall requirements specification; hence reviews were sporadic and unstructured. Many of the legacy features were not documented. As a result, testing had many gaps due to missing and incorrect information.

The Gen 1 product was targeted to run on both desktop and laptop platforms running on an Intel processor (CPU). Code was developed across multiple sites in the United States and other countries. Integration of the code bases and testing occurred in the U.S. The Software Development Lifecycle (SDLC) was approximately two years.

After analyzing the software defect data from the Gen 1 release, the Gen 2 team identified requirements as a key improvement area. A requirements Subject Matter Expert (SME) was assigned to assist the team in the elicitation, analysis, writing, review and management of the requirements for the second generation product. The SME developed a plan to address three critical requirements areas: a central repository, training, and reviews. A commercial Requirements Management Tool (RMT) was used to store all product requirements in a database. The data model for the requirements was based on the Planguage keywords created by Tom Gilb [2]. The RMT was configured to generate a formatted Product Requirements Document (PRD) under revision control. Architecture specifications, design documents and test cases were developed from this PRD. The SME provided training on best practices for writing requirements, including a standardized syntax, attributes of well written requirements and Planguage to the primary authors (who were

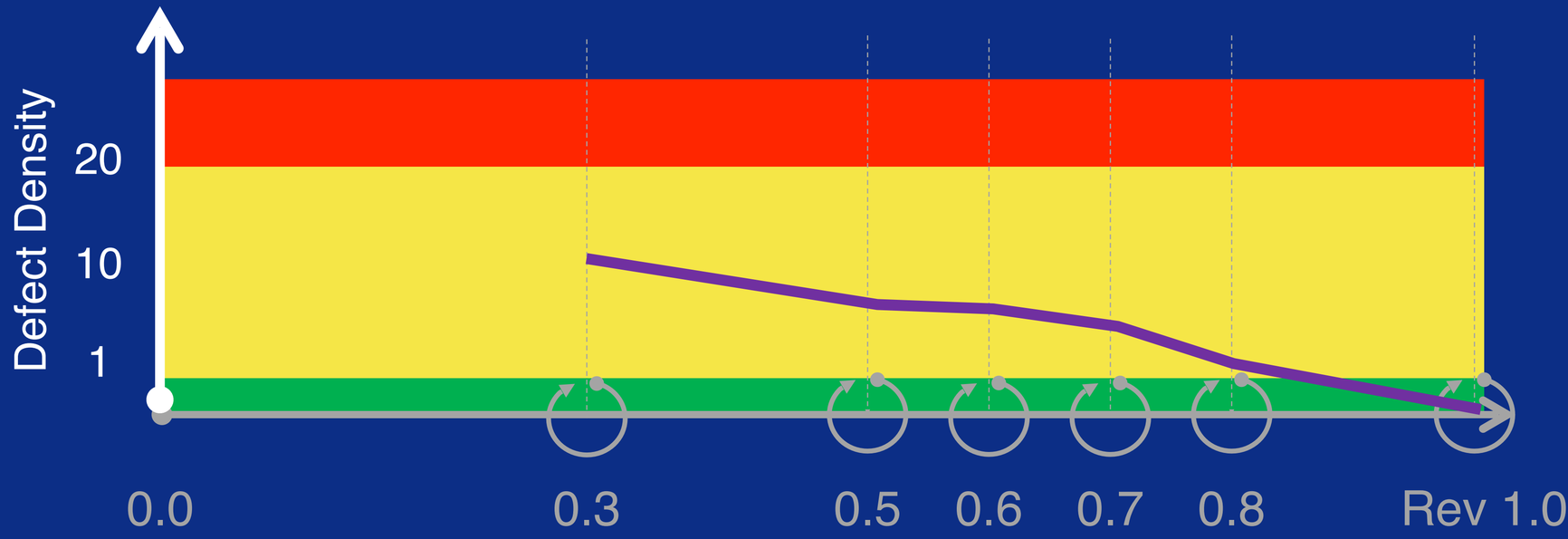
Intel Case Study

J. Terzakis, "The impact of requirements on software quality across three product generations," 2013 21st IEEE International Requirements Engineering Conference (RE), Rio de Janeiro, 2013, pp. 284-289.https://www.thinkmind.org/download.php?articleid=iccgi_2013_3_10_10012

Notice the team 'learning process' for tech standards

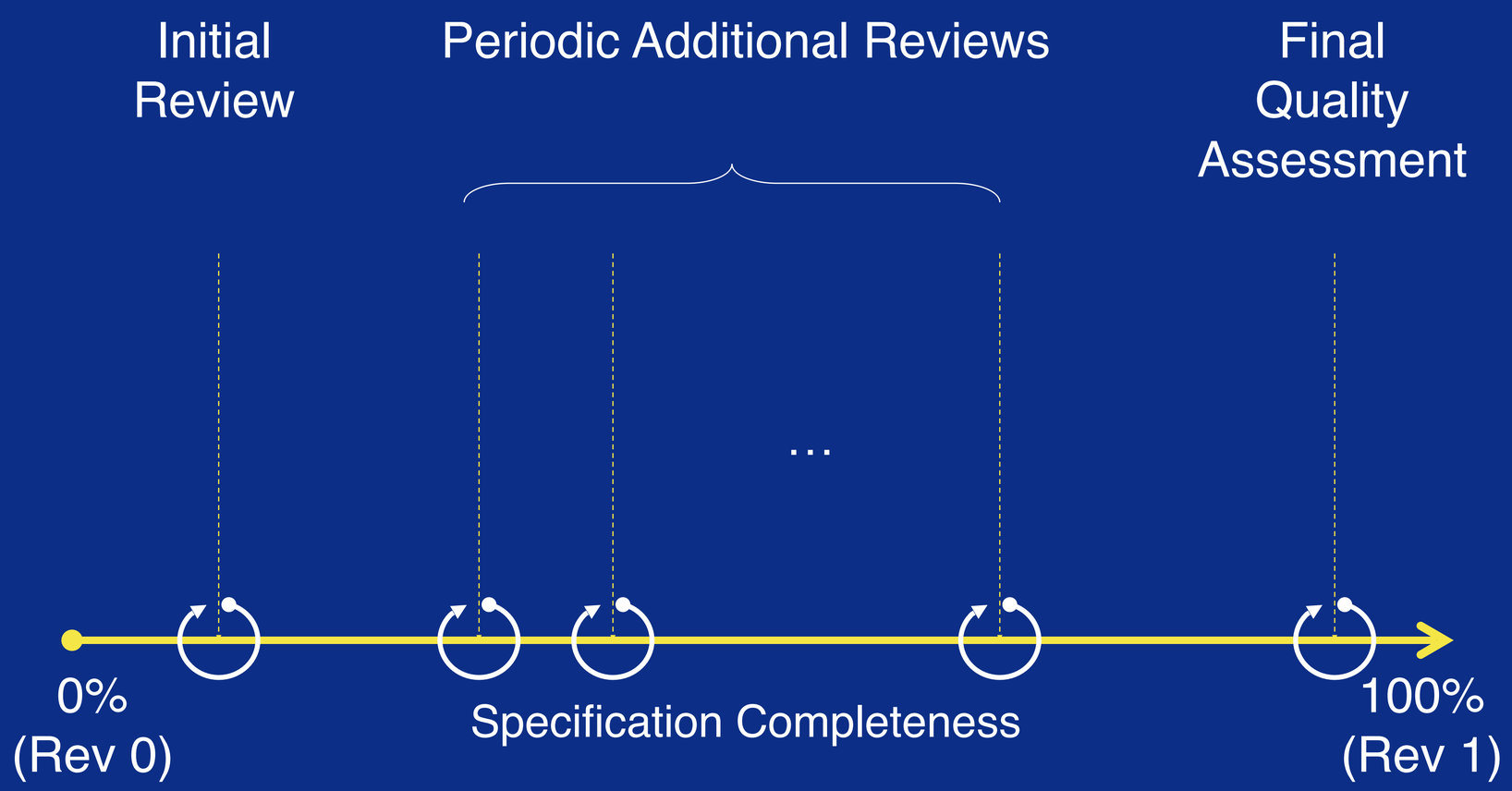
Example

A team in Client BIOS used SQC to reduce requirements defect density by 98% over six cycles:



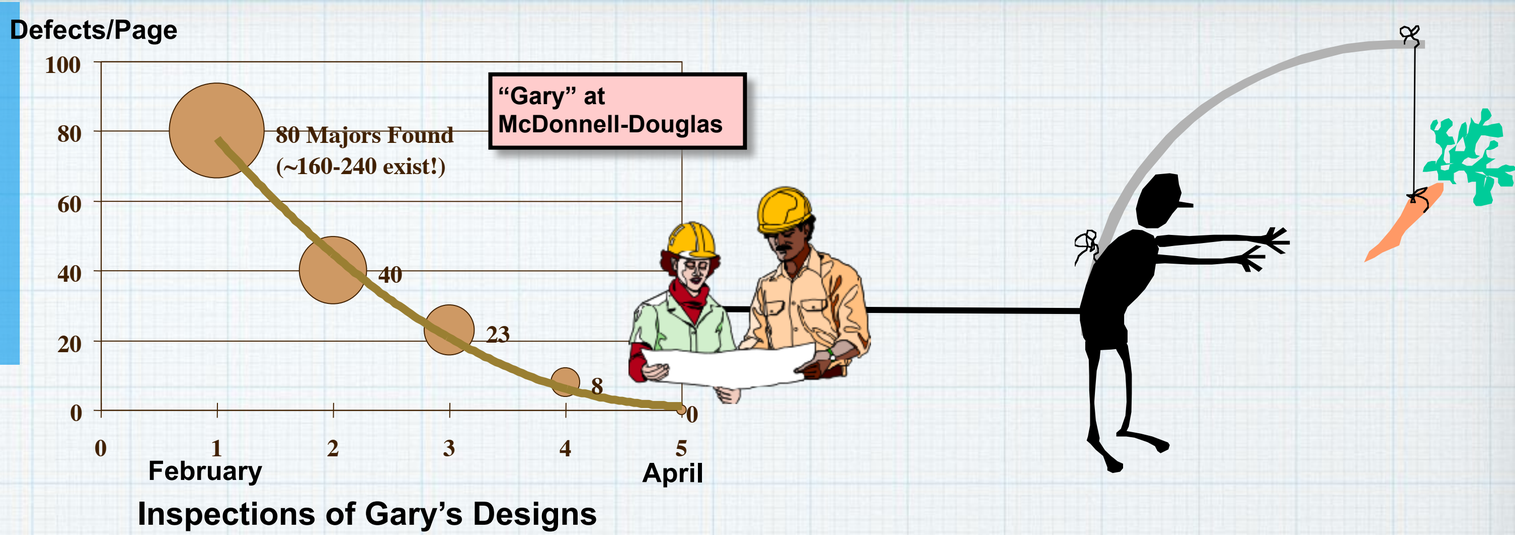
This effort had *significant* benefits to downstream work, including improved productivity (233%), time to test, and customer quality

SQC At a Glance



Specification Quality Control consists of a series of short, intense reviews that measure the defect density of a specification

Spec QC 'measurement'
achieves the following:



- The team is *motivated to learn* the corporate standards in practice
- Bad quality specification does *not 'get approved*, and then pollute the rest of the working process'.
- Intel reported 233% *productivity* improvement using these methods
  - (they are not fighting pollution at late costly stages of work).
- The document can be released with a *98% reduction in defects* compared with first submission.
  - This is a *learning* degree, and it persists.
  - People learn and remember how to do it right.

"We find an hour of doing Inspection is worth ten hours of company classroom training."

A McDonnell-Douglas line manager

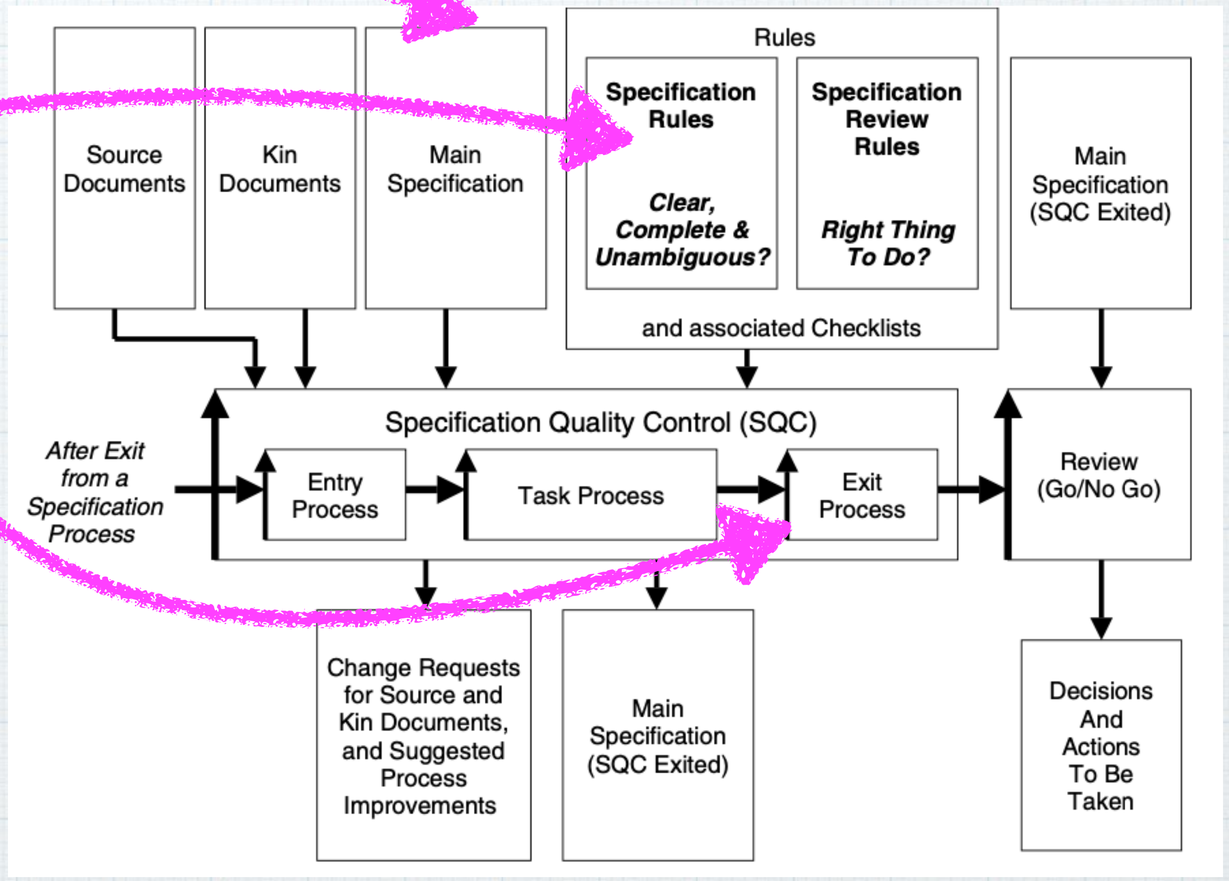
"Even if Inspection did not have all the other measurable quality and cost benefits which we are finding, then it would still pay off for the training value alone."

A McDonnellDouglas Director

The Management Responsibility for QC: No GIGO

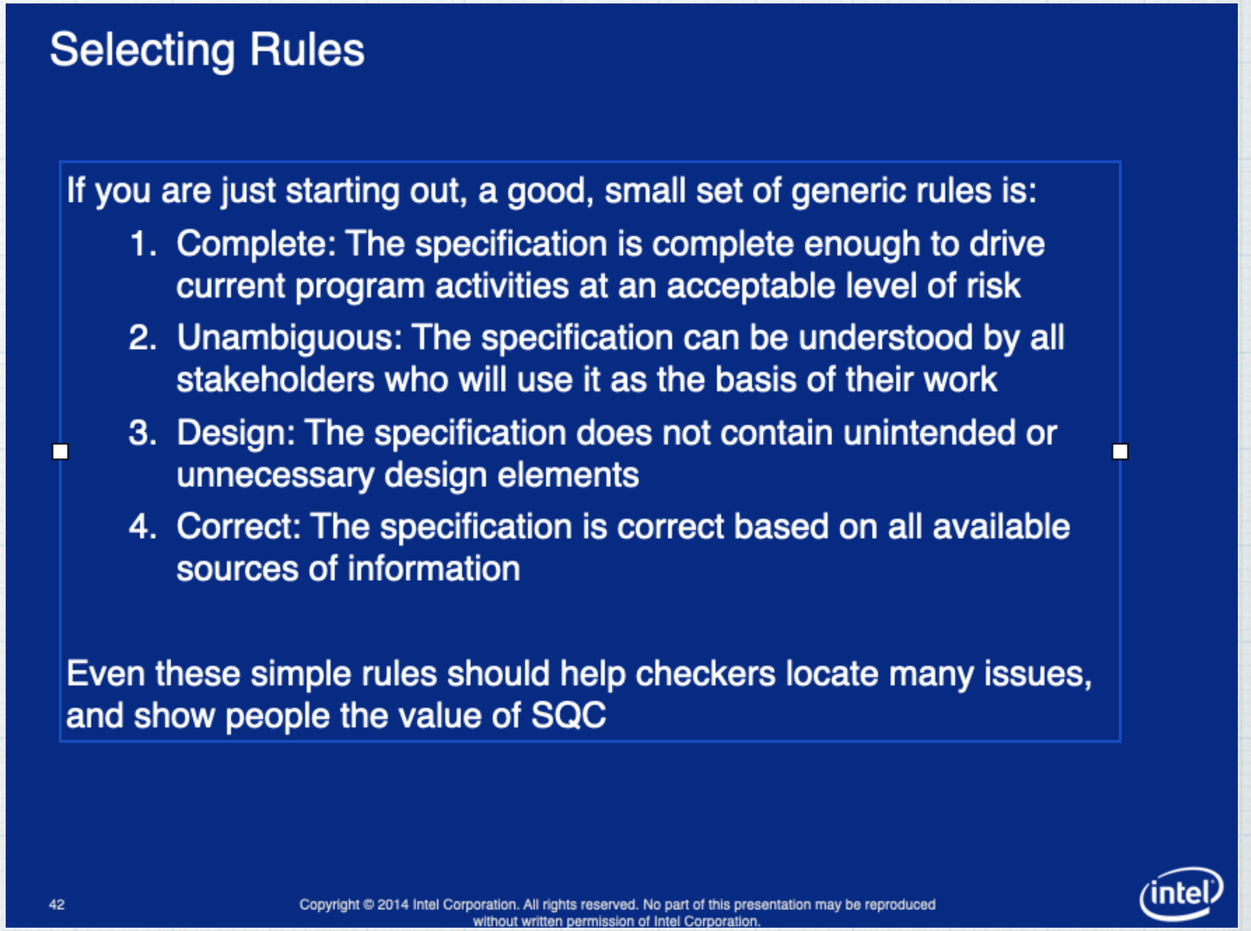
- Make sure all serious work is quality controlled
- Make sure standards are defined for that type of work, which work effectively in practice (corporate learning)
- Make sure bad quality-level work does not escape to the next process, even under pressure of time (you will lose *more* time if you do allow premature escape)

- None of these things will happen just because you employ educated professionals. You the management will have to lead and make sure it is in place.



The beauty of this SQC method is that

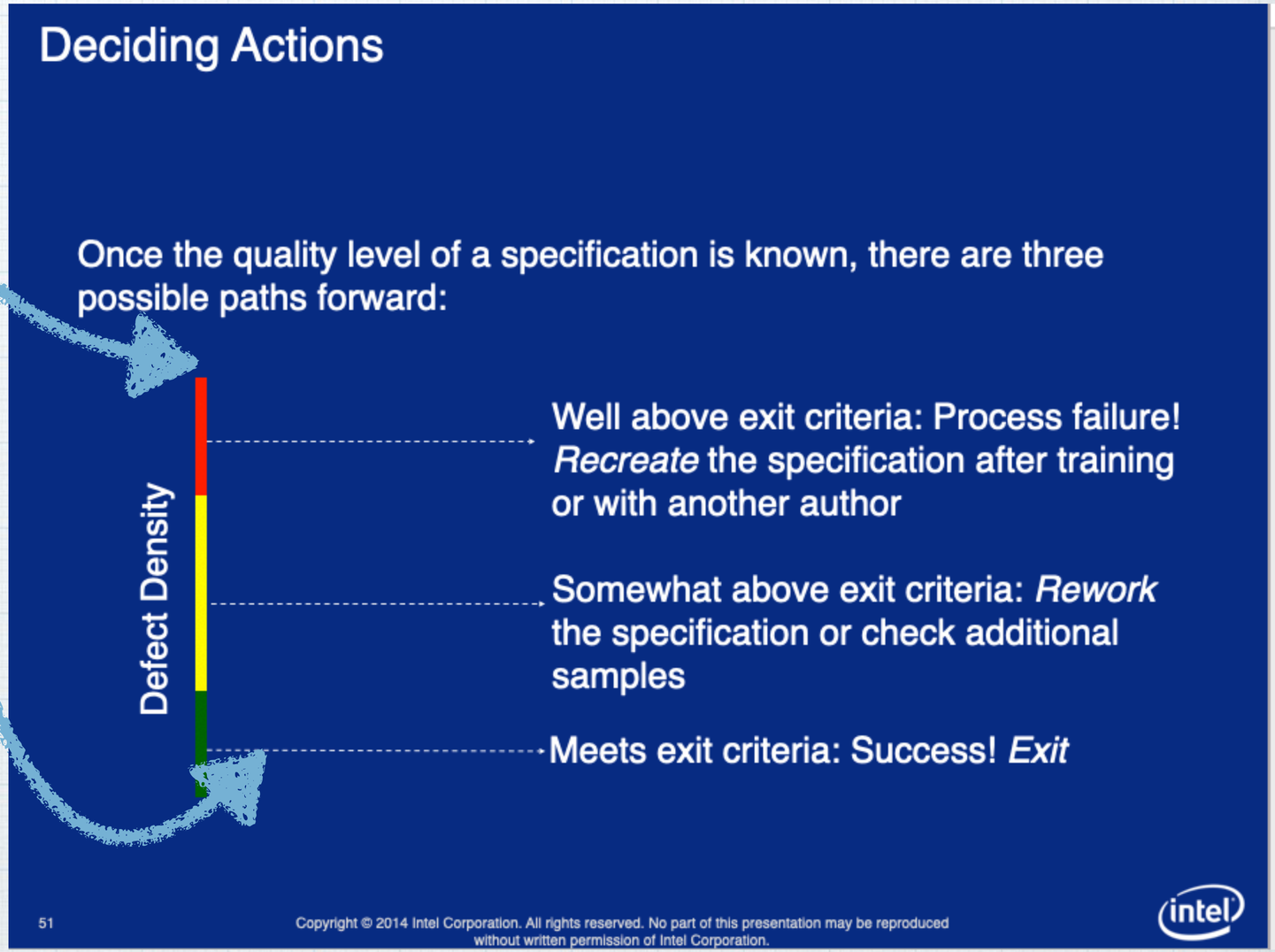
- The methods are *free*, and well-documented
- The methods *can* be done on a small scale locally, until you see how well they work, and decide to *scale up*
- The methods are *well-proven*, and documented for decades: you are not taking chances on somebody's sudden insight
- The method is *universal* and can be applied to any written specification, management, IT, engineering, legal, civil service.
- The SQC methods are *quite inexpensive*, an hour or two for a team of 2 people to sample, even a very large document.



Example of simple basic clarity Standard for requirements, as adopted by Intel

QC Principles: if you do X you get Y

1. MEASURE: If you measure work quality, you might get quality work.
2. STANDARDS: If you measure specification quality against standards, you might achieve those standards in practice.
3. TAILOR: SQC can check anything you actually put in your standards, clarity and correctness, for example



QC Policies: Management Attitude

1. MEET STANDARDS: All written work will be at least checked by sampling, against our standards
2. AVOID GIGO: High-density defects will not be allowed to exit. to next process
3. STITCH IN TIME: QC at *early* stages of planning, will be preferred over late testing of final product; because it is about 10X more cost-effective.

Case Study 2 – Results

First round	6 major defects per test case
Second round	0.5 major defects per test case

Time investment: 2 hours in initial review, 36 hours total in inspection, excluding rework (2 inspections; 4 hrs, 4 checkers, plus prep and debrief)

Test plan in use yielded over 1100 software defects with only 1 defect (0.1 %) closed as "functions as designed"

Historical rates were closer to 25% of all defects, with 2-4 hrs. spent on each. Time saved on the project: 500 -1000 hrs.

Defect Prevention in action: First inspection of this tester's next test plan: 0.2 major defects per test case

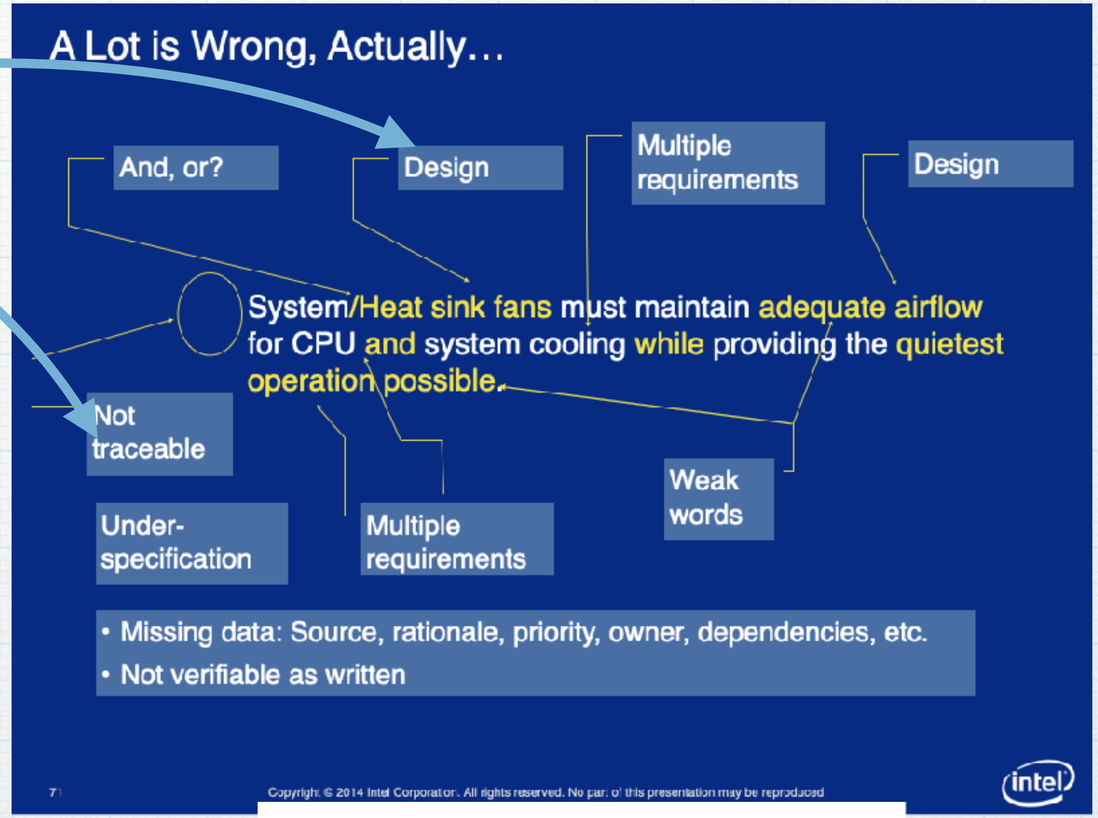
62

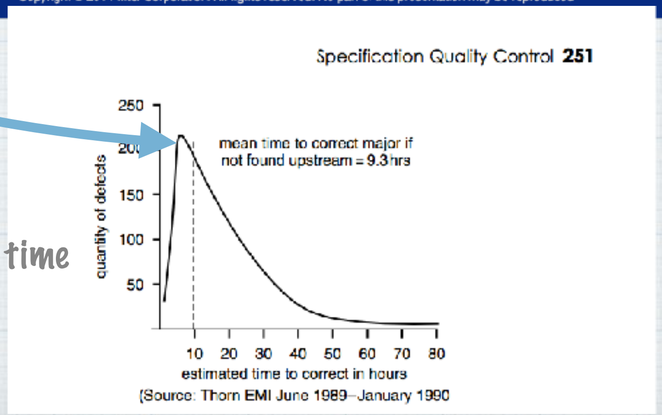
intel

Personal learning, of how to 'Do it right first time'. Preventing future defects

QC Checklist: ask yourself

1. REALLY-USEFUL, USED, RULES: Do you actually have a useful set of Rules for specifications, that people have to use in practice?
2. COURAGE TO SAY NO: Do you actually reject bad-quality work, based on Spec QC defect levels?
3. A STITCH IN TIME SAVES NINE (9.3!): Have you figured out the economics of having various densities of defects in specifications, when they are released?





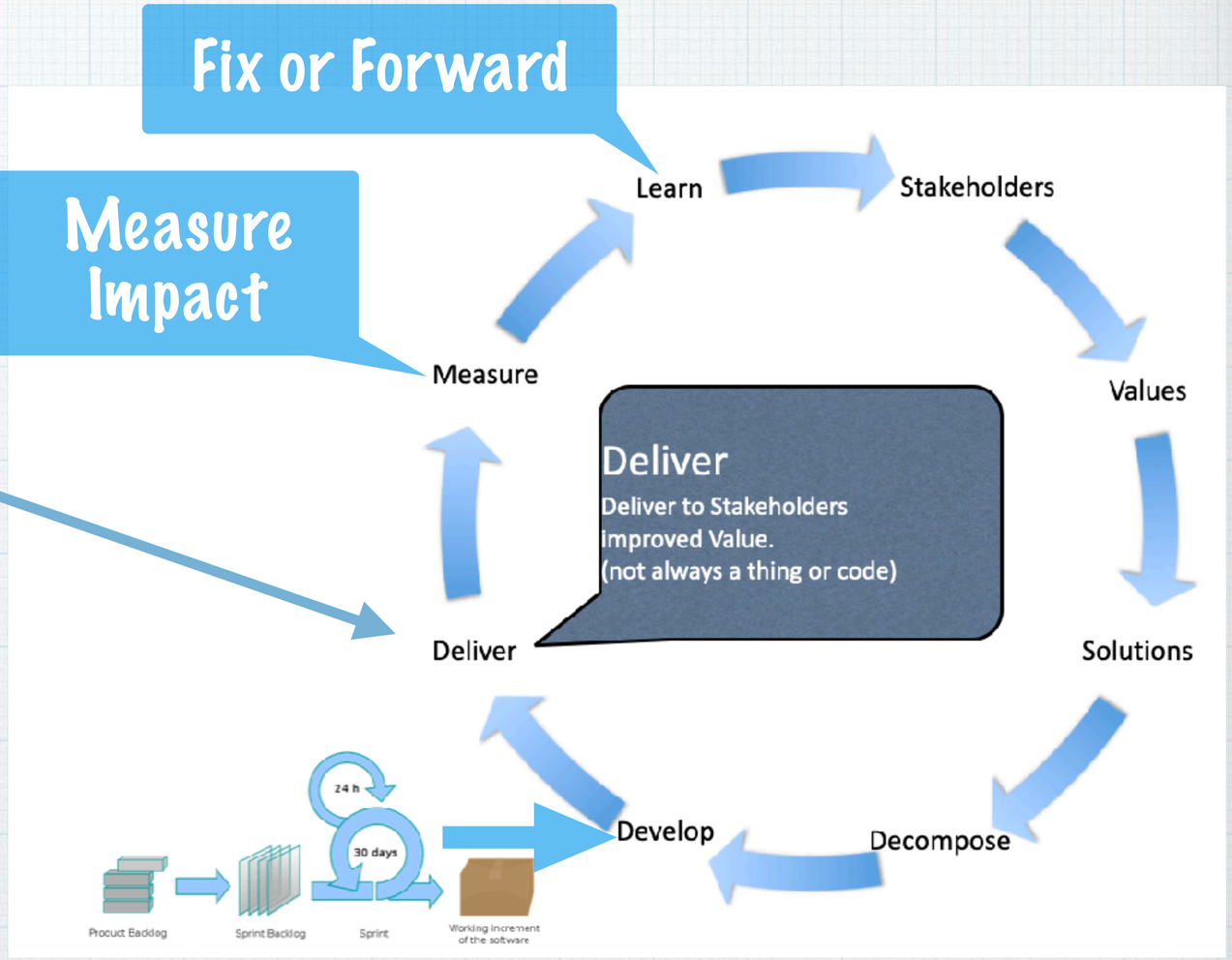
The Case: my client collected estimates for over 2,000 SQC defects, for probable future costs, if not caught early. The median future cost was 9.3 times more than catching defects at SQC time Versus test or field. NOTE, THE CEO was actively behind success here.

End Part 3
QA QC

Part 4.
Managing 'value delivery'

The Process of Making It Happen for Real.

- PREREQUISITES: The stakeholder critical requirements, the design, the design decomposition to small delivery steps, the Quality Assurance, is *done*. Develop or procure your design.
- DELIVER AND STUDY: The *next* step to manage is to integrate the design step into the existing system ('deliver'), and see how it works in practice.
- FIX IS NECESSARY: Adjusting step design if necessary.
- MANAGEMENT ROLE: What is the role of management at this stage?
  - Make sure these things really get done properly!
  - Do <u>you</u> make sure Values are measured at each delivery cycle, and relevant action is taken?



Here are the prerequisites for
MANAGING VALUE DELIVERY
management makes sure this set of things is in place

- One or more (parallel implementation is a possibility) *small steps are selected*, based on
  - *Value-to-cost* is good, and *risks* are low
  - The step has priority, to help reach a *Tolerable value level*, then later the success '*Goal*' levels.
  - The step is decomposed to be small (one week, or 2% maximum of total resource budgets)
- If step is successful, we can scale up
  - (example more cities or countries),
  - and we can have plans to do so.
- If not successful, as measured by value-level delivery,
  - a suitable specialist designer is ready to analyze root cause,
  - and suggest a new improved design to try out quite soon (Quinnan, IBM, Cleanroom)
  - (Dynamic Design to Requirements, to Value)

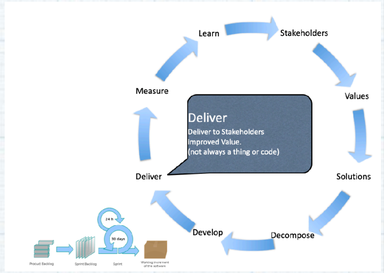
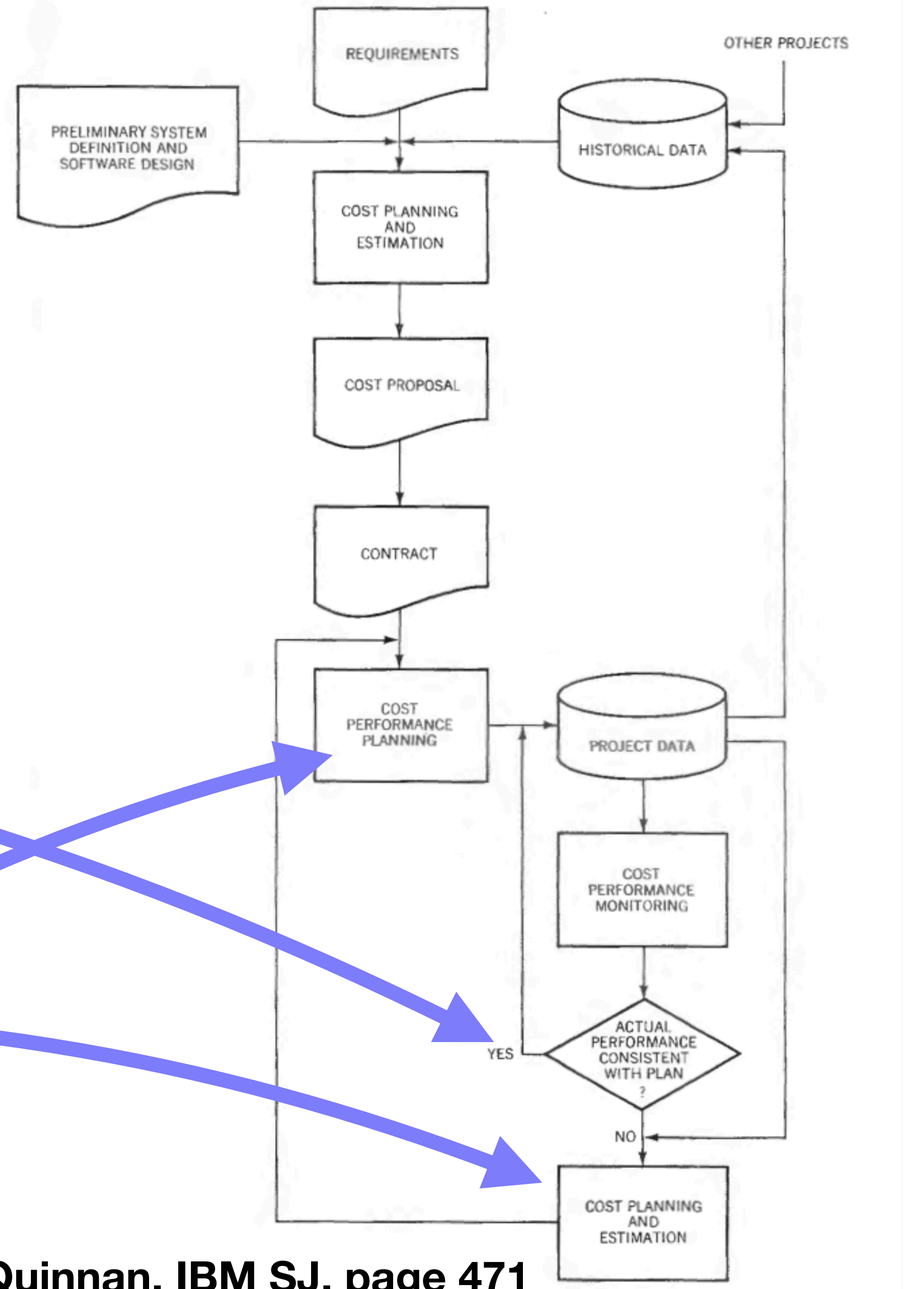
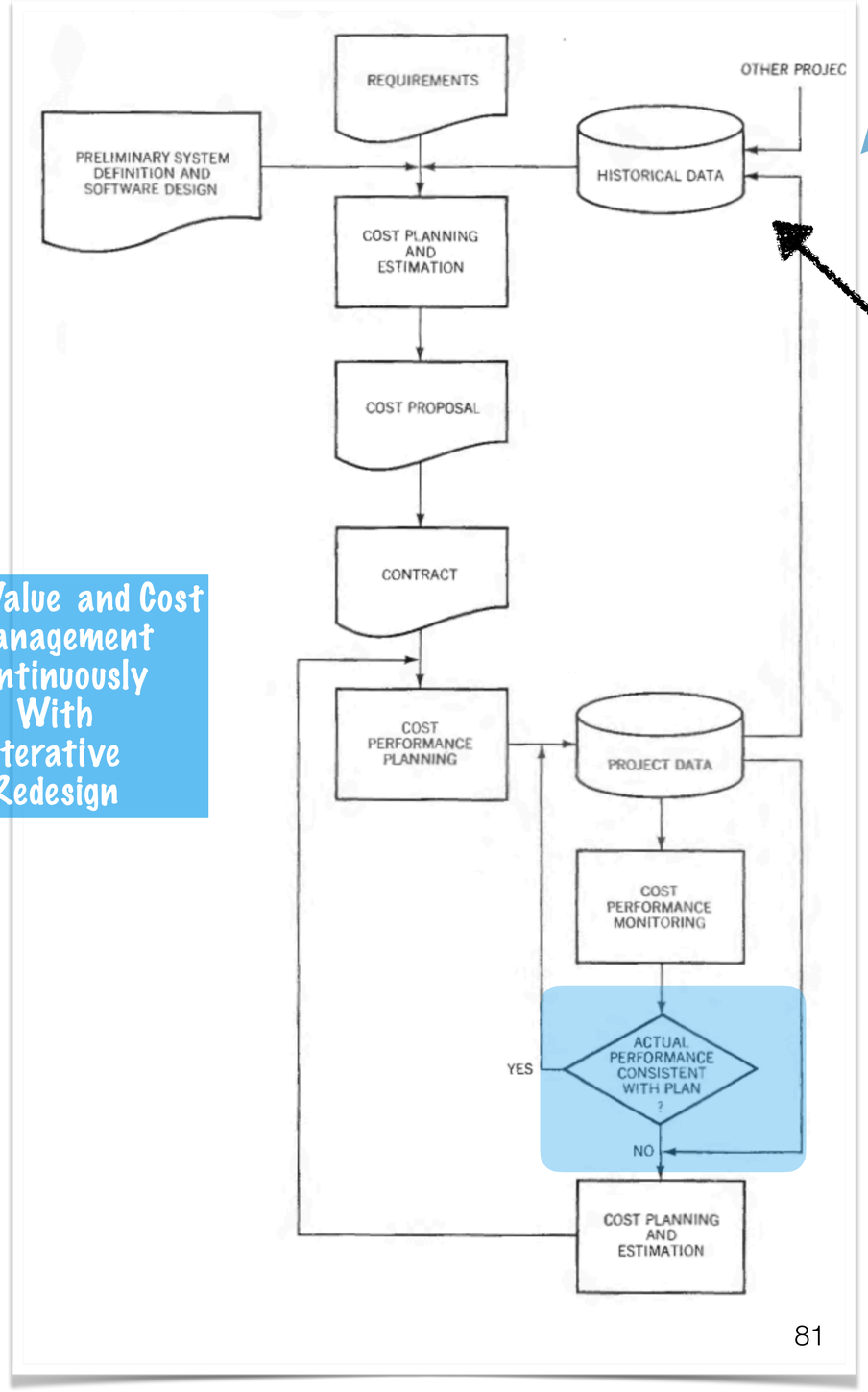


Figure 1 Cost management process



Source: Quinnan, IBM SJ, page 471

http://trace.tennessee.edu/cgi/viewcontent.cgi?article=1004&context=utk_harlan

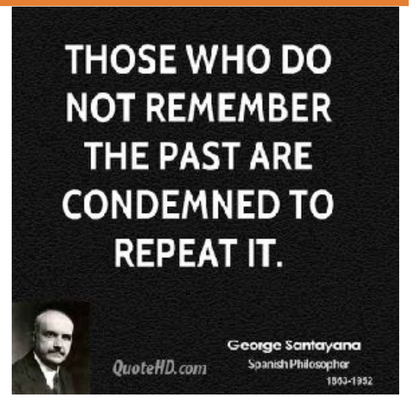


Agile Value and Cost Management Continuously With Iterative Redesign

Think: Fighting Covid-19 Virus by data collection

IBM 'Cleanroom' Cost Management Process

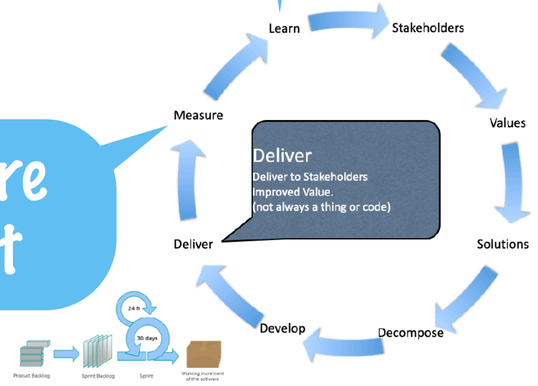
IBM FSD had a very advanced detailed collection of historical data from previous projects.
Published in IBM SJ, Walston and Felix
About 20 pages of data per project were collected



"ensuring that the design is cost effective"

Fix or Forward

Measure Impact



Source: Quinnan, IBM SJ, page 471
http://trace.tennessee.edu/cgi/viewcontent.cgi?article=1004&context=utk_harlan

Management opportunity: systematically collect lots of data and process parameters about all projects, as basis for future estimation

Here is management action after the value delivery step. For a self-managing team! But a CTO needs to put this in place (Trond J. Did!)


Trond Johansen

- ***Publish*** **the value increment results, in relation to requirement levels**
  - ("Security level is now 85% of the distance to the Goal for next month").
  - Motivate and celebrate with numbers.
- **Check that one-or-more steps are ready, by now, for delivery on the next delivery cycle.**
  - At least a specific idea,
  - even though the team might make a final decision at beginning of next cycle.
- You might well have several parallel teams,
  - so keep an eye on them all,
  - and make sure they get help when necessary.
  - Like when they fail at one increment, to get hoped-for results.

EVO Plan Confirmit 8.5

4 more product areas were attacked concurrently

Impact Estimation Table: Reportal codename "Hyggen"

Current Status	Improvements		Reportal - E-SAT features		
Units	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	50,0	14	13	12
			Usability.Productivity (minutes)		
20,0	45,0	[illegible]		35	25
			Usability.ClientAcceptance [illegible]		
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		

(Confirmit) 4 parallel teams delivering values independently

Notice 75% of time used (9 of 12 weeks).

Notice the left side color-codes. Red means high priority to do better.

Source http://concepts.gilb.com/dl33

Value Delivery Principles

- *You* do not not need to *micromanage*,
  - if your teams can manage themselves,
  - when they have *quantified values*,
  - and quick measurement
  - and quick feedback cycles.
- If you measure value-delivery in small increments,
  - you can also see a need to correct bad designs immediately,
  - you scale them up.
    - (Dynamic Design to Requirements).

		Current Status	Improvements		Goals			Step9 Recoding			
								Estimated impact		Actual impact	
		Units	Units	%	Past	Tolerable	Goal	Units	%	Units	%
6					Usability.Replacability (feature count)						
7		1,00	1,0	50,0	2	1	0				
8					Usability.Speed.NewFeaturesImpact (%)						
9		5,00	5,0	100,0	0	15	5				
10		10,00	10,0	200,0	0	15	5				
11		0,00	0,0	0,0	0	30	10				
12					Usability.Intuitiveness (%)						
13		0,00	0,0	0,0	0	60	80				
14					Usability.Productivity (minutes)						
15		20,00	45,0	112,5	65	35	25	20,00	50,00	38,00	95,00
20					Development resources						
21			101,0	91,8	0		110	4,00	3,64	4,00	3,64

Estimates

Weekly Testing

Priority

Next week

Warning metrics based

Cumulative weekly progress metric

Constraint

Target

4 x 4 person teams
Self managing

Value Delivery Policies
A Self-managing Team policy

- We will measure value-delivery levels, in small increments,
  - and allow our delivery teams to manage the resulting 'action' themselves.

- Management will not intervene,
  - as long as Value increments are reasonable,
  - in relation to time used, towards release deadlines.

Technical Debt

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

This value delivery cycle was based on Evo at Hewlett Packard And was extended with right 2 columns CTO and QA by Confirmit

Delivery Process: Checklist (some ideas to improve projects)

- Have you quantified the value requirements
  - so that your teams can self-manage,
  - step by step?
- Have you made sure that incremental measurement of value delivery is
  - planned, practiced and practical (economic, credible).
  - Maybe even measured independently of the implementation team? (Microsoft did it for Confirmit on Usability)
- Have you considered setting up two-or-more parallel value-delivery teams,
  - so that there is a sense of *friendly competition,*
  - when the incremental results are published?

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.	60 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

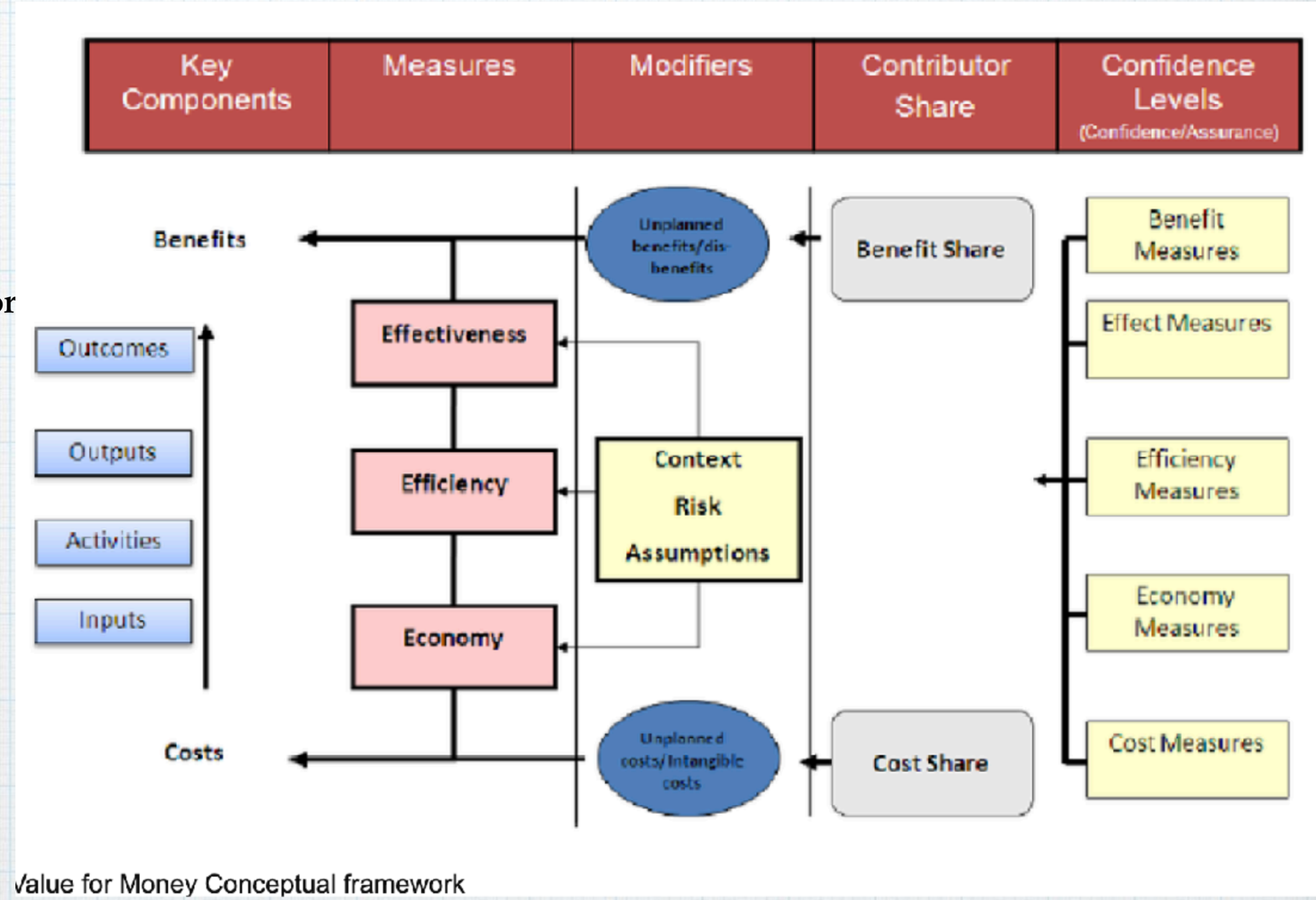
At the Confirmit end
of the first 12 weekly cycles of using Evo,
25 Product qualities were improved,
So much as to destroy competitors
Here are there top 5 improvements.
Way beyond wildest expectations

Part 5.
Managing suppliers and Contracts

Managing Suppliers and Value Delivery.

- Your suppliers do not know about and do not care about your success or failure, in delivering Values, or in any sense.
- They are quite happy to take your money, until you fail, or are bankrupt.
- An analysis of the huge quantity of failed IT projects reported in the internet is clear about this point.
- Or, did you even hear of a supplier who is reported to have
  - returned all payments, after a clearly failed project?
  - Never. Why should they. It is your fault!

- It is management's responsibility to make suppliers care about their customer's success and failure.
  - But do *you*? Do you know *how*?



Value for Money Conceptual framework

Why is 'Value Contracting' a problem?
Why should you allow payment with no corresponding value for your company?

- Managers do not know how to articulate their values quantitatively
- There is little to no culture or tradition to putting those values into contracts.
- People do not seem to know how to measure their quantified values as they are delivered
- People (the professionals involved, their managers) have the misunderstanding that, the technology, and the strategies needed, are the 'real' value drivers. Wrong!
  - That is, '*if we only build this new technology, then the values we want will automatically and guaranteed, be delivered*'. Latest fads, 'digitization', 'AI'.
- People do not understand how to decompose their strategies and technologies into small deliverable value increments.
  - They understand some forms of decomposition, like bill of materials decomposition.
  - But they have no training or theory in decomposition into *value delivery increments.*

Figure 5 (Contract Framework).

Source: Flexiblecontracts.com

Here is the management job.
To get Value-for-Money, 'No Cure-No Pay' contracts

- Write your contract so that there is a relationship between measured value delivered, and supplier payment
  - It does not have to be pure-pay-for-value, but it should be enough to focus your supplier's attention on *your* value objectives.
  - Payment should be incremental, as value objectives head towards your Goal levels.
    - But whether is it weekly, monthly or quarterly are options.
- If you cannot deliver value with a given supplier, then maybe a change of supplier is called for.
  - Make sure the contract allows you to terminate a contract at any increment.
- It is possible to have more than one parallel supplier, delivering values.
  - This motivates the suppliers, a healthy competition.

SOTO Specification
(from contract template)
short-term Statements Of Target Outcomes

SOTO Completion Date	*NOTE: Please state not applicable if this is not being used.*
The problem or opportunity to be addressed	
The Business Objectives	
The Target Outcomes	*NOTE: These should be in line with the Business Objectives. They should be bullet points only and listed in order of priority.*
The Constraints	*NOTE: Examples include design constraints, minimum quality constraints, budget constraints, schedule constraints, resource constraints.*
Customer responsibilities	*NOTE: This should include any support, facilities and information, including any requirements for execution of the Options, which are to be provided by the Customer.*
Time frame for provision of feedback by the Customer	
Early termination payment	

Figure 5 Contract Framework. Source: flexiblecontracts.com

Specifying your critical Values for a supplier

Air Quality

Level: Stakeholder, **Type:** Value, **Labels:** critical Edit

Is Part Of: Top Values

Status	Status	Tolerable	Tolerable	Goal	Goal	Stretch
9.5k	1.5k	200	100	950	150	0

Tolerable [Persons = **Senior**, Time = **5 years**, Pollution = **NO2**, Area = **Greater London**] @ 2022 : **200** People

AQ.Ambition Level: Drastically improve air quality in London to acceptable legal levels as stated in the Paris Accord (Paris Agreement)

Scale: Number of [Persons] who reside in London Boroughs dying from exposure over [Time] to [Pollution] per year in [Area]

Meter: Recent Hospital records from London hospitals for deaths by Pollution Exposure related illness

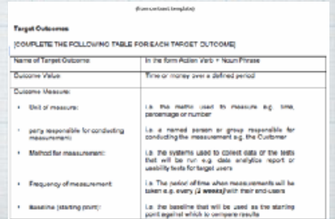
Status: 9.5k People [Persons = **<All>**, Pollution = **<All>**, Area = **<All>**] When 2019

Status: 1.5k People [Persons = **Senior**, Pollution = **NO2**, Area = **<All>**] When 2019

Tolerable: 200 People [Persons = **Senior**, Time = **5 years**, Pollution = **NO2**, Area = **Greater London**] When 2022

Tolerable: 100 People [Persons = **Child**, Time = **1 Year**, Pollution = **{NO2, Carcinogens}**, Area = **Greater London**] When 2020

Goal: 950 People [Persons = **<All>**, Pollution = **<All>**, Area = **<All>**] When 2028



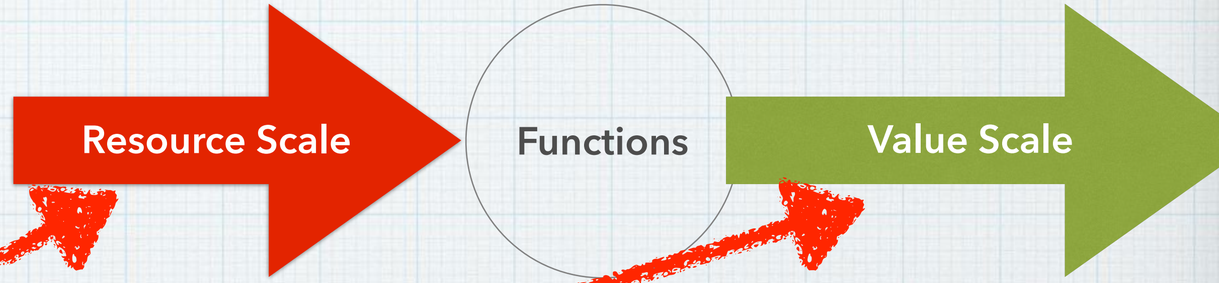
Part 6.
Managing Risks

Risk: a risk is something that can go *wrong*.

An *opportunity* is by contrast, something that can 'go right', get better.

Risk Dimensions: The risk *dimensions* are all **critical** *values*, and all **critical** *resource* limitations.

Figure 6 (Resource and Value) the two areas of risk concern: bad quality and high costs.



- **The term 'Critical', by definition, is**
  - **a value or cost area,**
  - **which can become negative,**
  - **to the potential extreme**
  - **of total system failure, as a result.**
- **No matter how good**
  - **all other values and costs are.**

Managing Risk:
the managers responsibility for the Value aspects , especially.

- Everything discussed thus far
  - is somehow a part
  - of what managers can do
  - to manage value risks.

- At the same time there is some element of risk
  - in every idea,
  - every requirement,
  - every design,
  - every estimate,
  - every measurement,
  - every prioritization.

- Risk is pervasive.

- Maybe so pervasive, that we do not notice it sometimes.

- That's life.

"Risk comes from not knowing what you're doing"

— Warren Buffett (1930-)

Case: Ericssons Decentralized Risk Management Policy

* Our wonderful client Ericsson of Sweden, mentioned in Productivity case above, had a wonderful and insightful risk policy.
  * They said that, I paraphrase it here,
  * *'risk is the job of every engineer, at all times'*
* The responsibility of management is therefore
  * to make sure that the professional staff,
  * those who encounter risks and can do something about them;
    * including engineers, IT specialists, lawyers and not least managers,
    * are trained, updated, tooled, motivated, rewarded
    * for dealing with risks in a responsible way.
* Even under pressure for delivery times, sales, cost cutting, conventional thinking, and other pressures, which are there, even in the best of organizations.
* Notice what Ericsson is saying. There is no 'special risk analysis process' as such. There is no risk department, director or specialized entity. Everybody, all the time, in every little detail deals with risks: because that is where big critical risks can occur and hide. Everywhere, all the time.

Risk management is integrated with the business and its operational processes

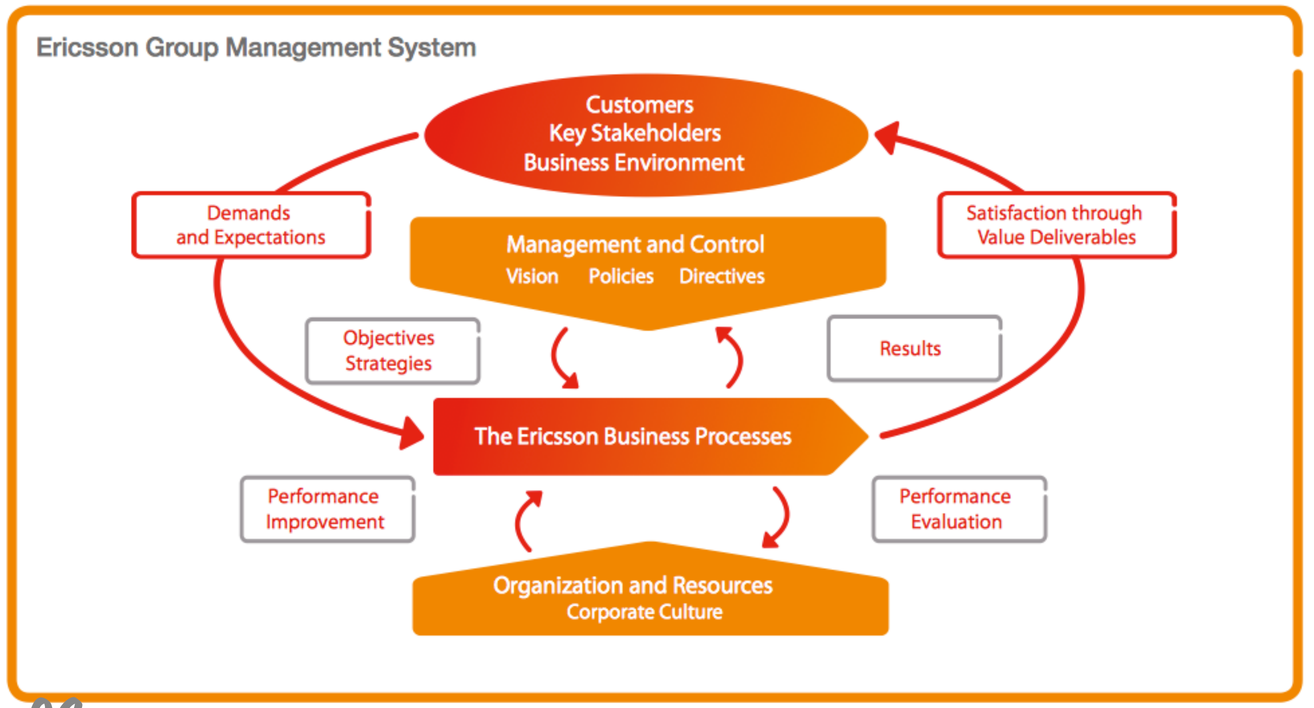
and is a part of the Ericsson Group Management system

to ensure accountability, effectiveness, efficiency, business continuity and compliance with corporate governance,

legal and other requirements.

the Board of Directors is also actively engaged in the company's risk management." [Ericsson Management Policy]

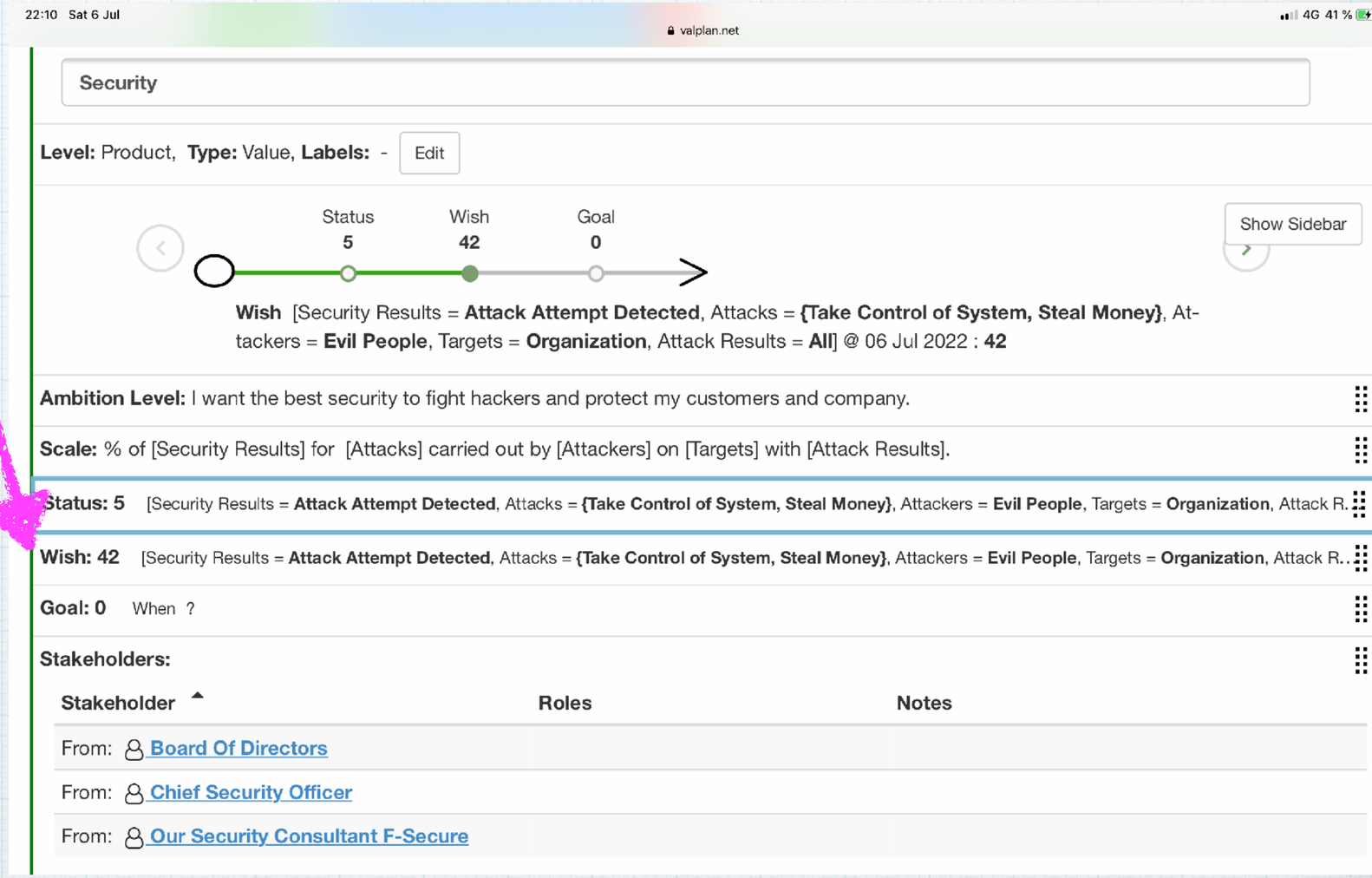
ERICSSON



‘Every little planning detail’ is a risk

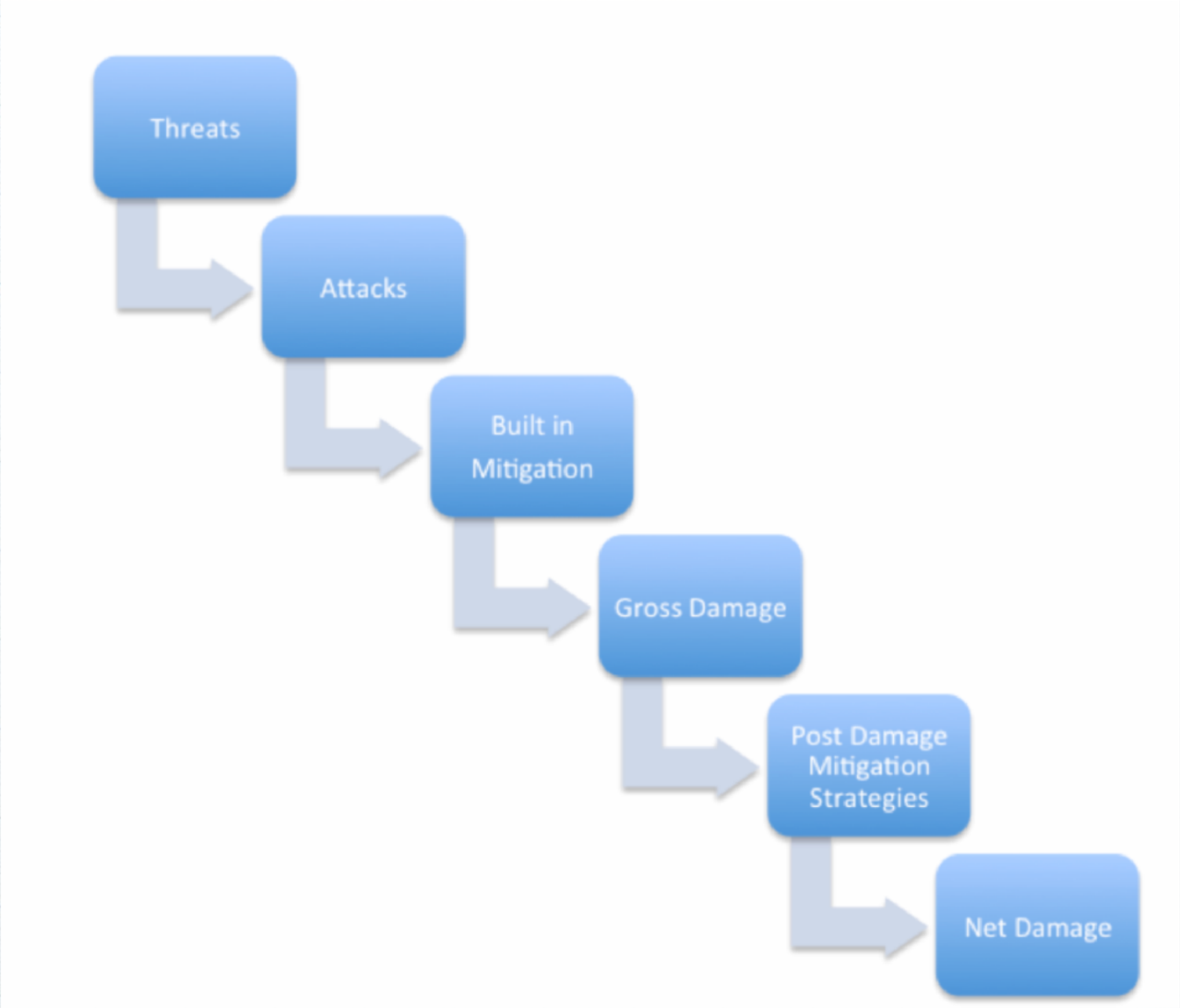
I had a little classroom game I played at Ericsson.

- I would put on the screen a detailed requirement, which was our subject matter.
  - I would ask them to tell me which details they thought were related to risk management.
- They would suggest something obvious, at first, like the *Wish* level.
  - If that were not good enough, they reasoned, the entire project risks wasting its time, on the wrong requirement level, and worse;
  - not actually achieving the correct intended requirement level.
  - Maybe the whole project would fail, based on that detail alone.
  - Good start to answers.
- Then I would ask if they could see any other details related to risk,
  - and to make a long iterative story shorter,
  - I did not give up until *absolutely every line, and every detail in every line*
    - (like ‘Security Results = ...’ in the Wish statement).
    - Try it yourself: any parameter (Ambition, Scale, Status, Stakeholders,...) any detail (Chief Security Officer)



‘Risk discussion’ at meetings

- Another way for a manager to keep reminding people that the ‘devil is in the details’,
  - is to ask questions at meetings.
  - Show interest, and explain
  - (from your experience of problems)
  - what could go wrong,
  - and why a particular risk is worth paying attention to.
- “are there any other critical stakeholders, we might have forgotten?”
- “Are there other Security Attack types we might consider, however improbable?”
- “If I extended the deadline by 1 working day, could you do a much better quality job of documenting risks, in this?”
- “Which risks are we failing to document here, perhaps because they are documented elsewhere?”
- “We missed listing EU Laws as a stakeholder on the last project, and they threatened us with €10 million fine and exclusion from the EU market”

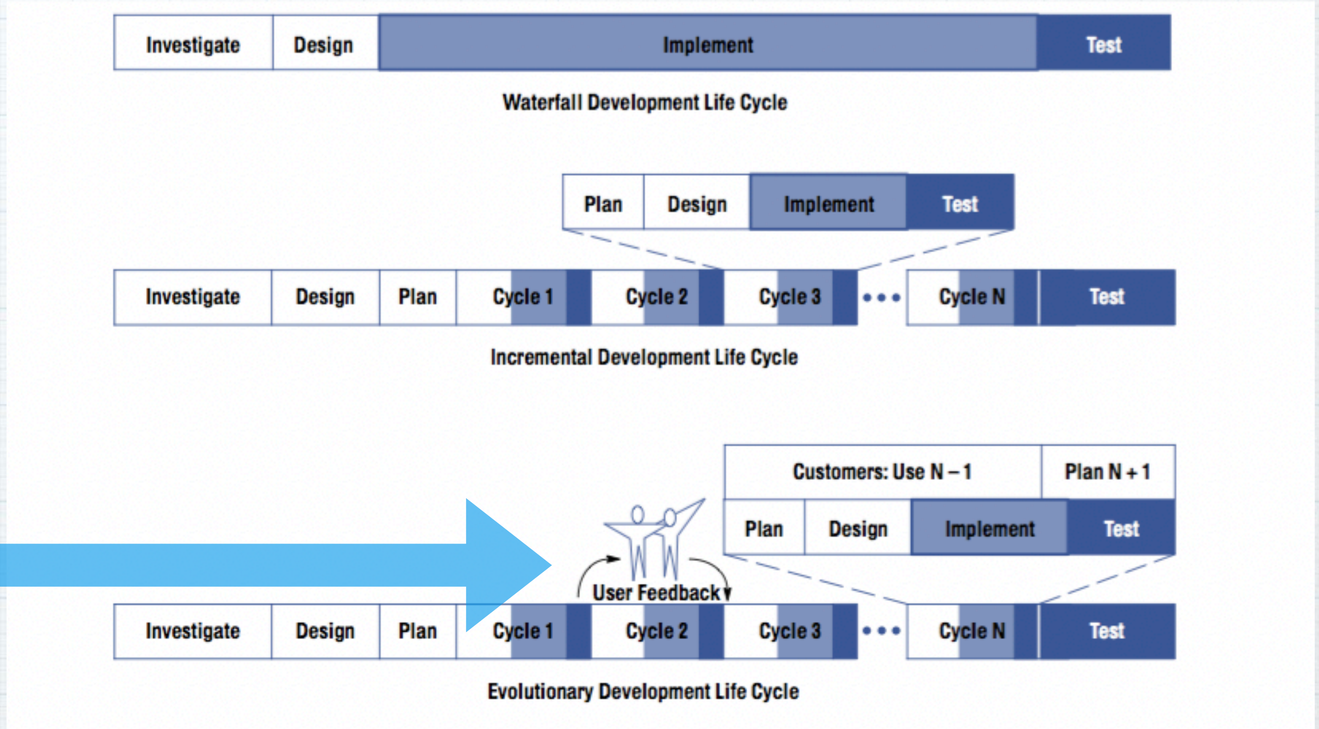


Principles for Managing Risk (= smart high-leverage ideas)

1. INDIRECT RISK MANAGEMENT: You can never cover all risks directly, there are too many and too many unknowns, but you can *cover risks indirectly and generally*, with things like insurance, mitigation, and contracting.

2. MINIMIZE DAMAGE: The point with risk management cannot be to foresee and prevent all risks; the point is to *minimize lifecycle damage* reasonably, and to try to mitigate or compensate for catastrophic risks.

3. VALUE FOR 'RISK MANAGEMENT COSTS': There needs to be a *reasonable balance* between risk management investments in your professional planning, and the possible and probable costs of not planning for some risks.



[HP] Another big user of our methods. Very good at Evo: Incremental with feedback.
Three approaches to delivering projects.
The Evolutionary cycle (bottom) is characterized by stakeholder feedback and change.
The others just hurry up to be late with bad quality.

http://www.gilb.com/DL35,
http://www.gilb.com/DL65,
http://www.gilb.com/DL67,
http://www.gilb.com/dl825

Policies for Managing Risk
Let your professionals know you want them to be 'fanatic about risks'

1. PERVASIVE RISK: Risk responsibility is
   1. the personal responsibility of every professional employee,
   2. at all times when their work 'touches' risk elements:
   3. they must sense risk,
   4. document risks,
   5. mitigate risks,
   6. and then mitigate the effects of damages, *caused* by risks.

2. ENABLING: Management will ensure that
   1. all training,
   2. all standards,
   3. all tools
   4. will be constructed to help the professional staff and suppliers to deal with risks,
   5. especially critical risks.

3.RISK RESPONSIBILITY: managers at all levels will motivate, enable and show interest in risk management.

	->	->	->	->	->	->	
Planning	Error (in Planning)	Specification Defect	Spec QC	System Defect	Malfunction or System Failure	Measure Value Delivery	Net Damage
Security	Threat	Attack	Mitigation (Built in)	System Defect	Damage (Gross)	Post Damage Mitigation (Recovery)	Net Damage
Program -ming	Coding Error	Code Bug	Inspection	Bugs Remain	Bug Encountered	Remove Bug	Net Bugs Remaining
Pattern	Fault	Fault	FIX	Fault	Fault	FIX	Fault

Table 7.8 The planning chain of areas of threats and mitigation, compared to 2 similar security planning and software reliability planning scenarios. © gilb.com 2014.

Checklists for Managing Risk
Things you can do as a manager

1. What can you do,
   1. in terms of <u>training, and standards</u>,
   2. to improve the practical everyday risk management work?

2. Have you regularly done something
   1. to *motivate* your staff
   2. to <u>care</u> about risks?

3. Have you demonstrated repeatedly to your staff
   1. that it is better to take <u>more time</u>,
   2. than to take more risks, within reason.

"I not only could not stifle controversy among your readers. I welcome it.

This Administration intends to be candid about its errors;

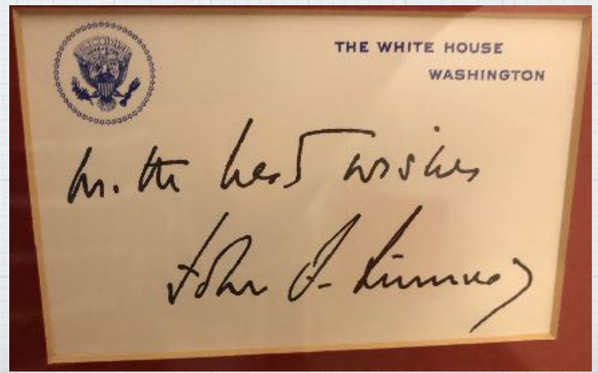
for as a wise man once said:

'An error does not become a mistake until you refuse to correct it.'

We intend to accept full responsibility for our errors;

and we expect you to point them out when we miss them."

John F. Kennedy (1917-1963)

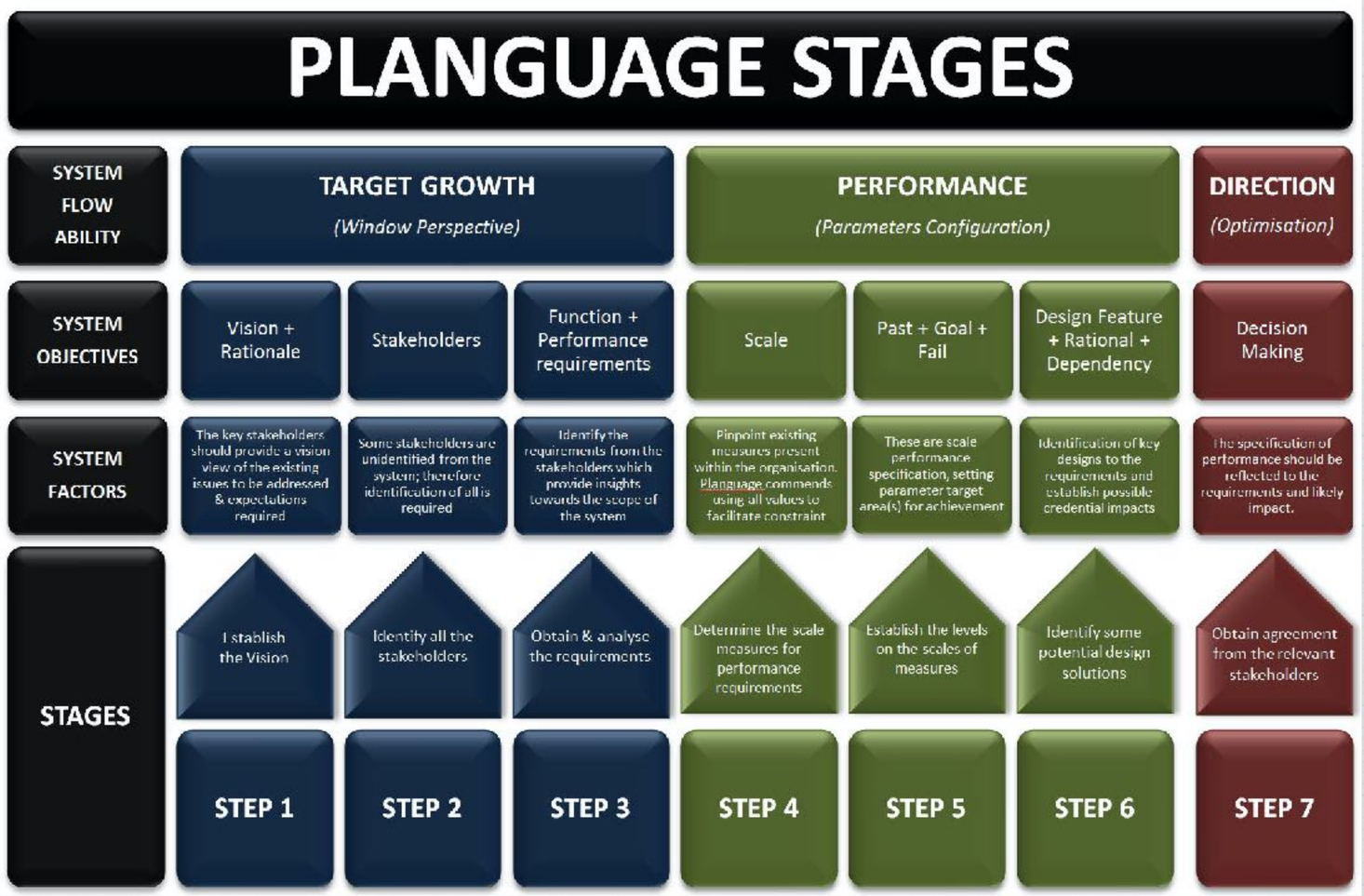


Part 7.
Managing priorities

Bad prioritization methods include:

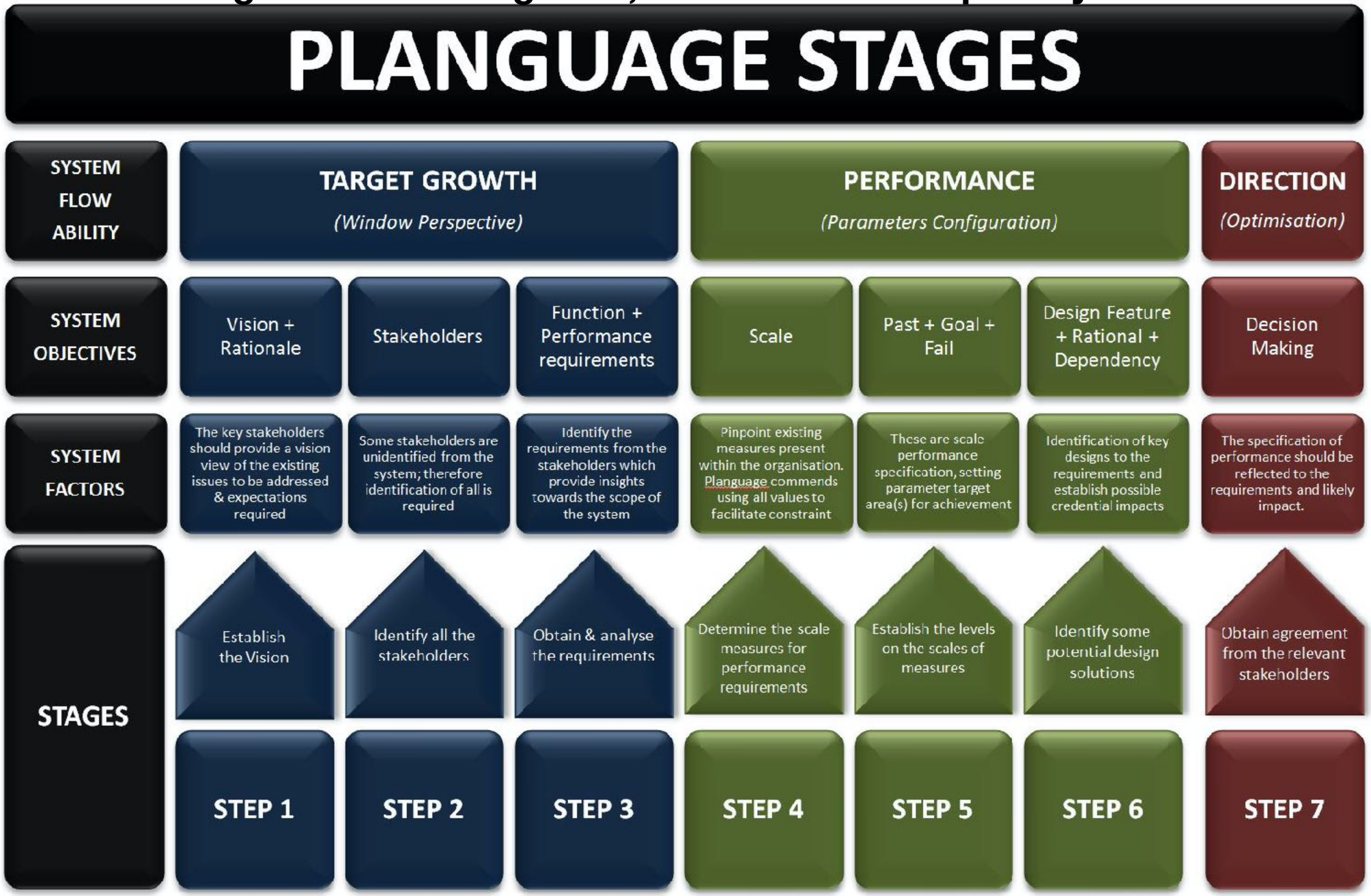
- Priority weighting, fixed in advance
- Source of weights not given, no responsibility
- Reason for weights not documented. Why?
- Weights not based on Values, and resources, requirements
- Weights not based on *remaining* resources, and *remaining* distance to value requirement levels.
- No consideration (or documentation of such consideration)of the quality of underlying facts, for the prioritization weights.
    - Is the fact basis 'wild guesses' or 'solid facts'?
- Total lack of dynamics, or re-prioritization when things change in the incremental measures of progress, or in the outside world.

Tse and Kahlon **Stages of collecting data, in order to make priority decisions**



Value Planning book, Chapter 6 Prioritization
https://www.dropbox.com/sh/34llx1a7ckyagxl/AAA0pDzSxN5WmoP9IOKR0Mpca?dl=0

Stages of collecting data, in order to make priority decisions



If you do not prioritize intelligently, then : These things go bad

- you cannot then expect a *series of quick wins*
  - at the *beginning* of your project value delivery stream.
  - Quick measurable Value improvements
    - will create credibility,
    - and willingness to co-operate
    - on next steps from users and stakeholders.
- You might end up funding a bunch of loser ideas
  - (high cost, low value),
  - leading to lack of <u>your</u> credibility, losses of time and money.
- You might get involved in a 'digitization' project,
  - where the new technology is the main point,
  - and values are forgotten.
  - These are often big scandal failures.
  - There are too many of these.
- You may miss getting critical things delivered
  - by critical deadlines:
- for goodness sakes,
  - do the important stuff *early*, and
  - do not waste time rebuilding systems
    - and functionality that in fact already exist.
  - (=large costs, long time, no value, maybe 'ever')

The highest priority for human survival is:

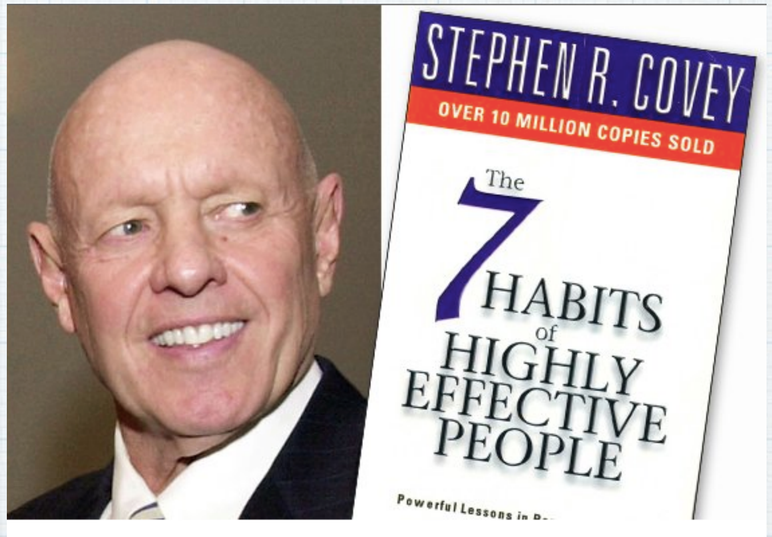
- Water
- Air
- Food

Hint: 'It depends' (on unknowns)
See Presenter Notes.

What is the job of the manager, regarding prioritization?

- Review the above lists of bad and good prioritization practices:
  - read references,
  - if you need a deeper understanding.
- Be prepared for your current culture to have old and 'informal' prioritization methods
  - (I did not write 'bad')
- Take a position on what you feel is necessary and logical
  - (Good practices, we hope)
- Announce that your teams are going to use
  - *"Value and Cost Based Dynamic Prioritization"*
- Make sure your teams are trained and equipped
  - (tools, templates, Rules, Checklists, Coaches)to do so in practice.
- Keep on asking questions, to make sure people are taking this seriously.
  - *"Show me the detailed basis for your priority next week".*

> ***"The bottom line is, when people are crystal clear about the most important priorities of the organization and team they work with and prioritize their work around those top priorities, not only are they many times more productive, they discover they have the time they need to have a whole life."***
>
> *Stephen Covey (1932 - 2012), Utah, USA*



Covey makes a great point:
Smart prioritization and delivery of
Critical value early and
Incrementally, takes the deadline pressure off your Organization. It gives people 'a life'!

Good prioritization: smarter ways to prioritise (are quantitative, and dynamic)

Good prioritization methods are the opposite of the bad ones, above. They are built into the planning language Planguage.

The Good Prioritization Methods

- Prioritize our critical required value levels
  - First priority the survival (not-failure level) constraints (Tolerable)
  - The achievement of the Goal (Success)
  - Then priority for possible optional Stretch levels
  - Then NO PRIORITY (no resources) for going *beyond* requirement levels.
- Good prioritization is based on step-by-step incremental measurement, of the degree to which we are achieving our requirement levels. Priority is dynamic, not 'static up front'.
- Good prioritization considers remaining budgeted resources
  - It will prioritize designs using mainly the resources, that we *still have enough of.*
  - And will *not* choose, or at least will avoid, designs requiring resources we have 'too little left'.
- Good prioritization will consider the quality of the many facts we use, to reason about priority, such as;
  - Which *stakeholders* are interested, and why, and consequences
  - Which *assumptions*, *issues* unresolved, known *risks* are involved

All of this is good common sense. All of this is already baked into the methods we refer to.

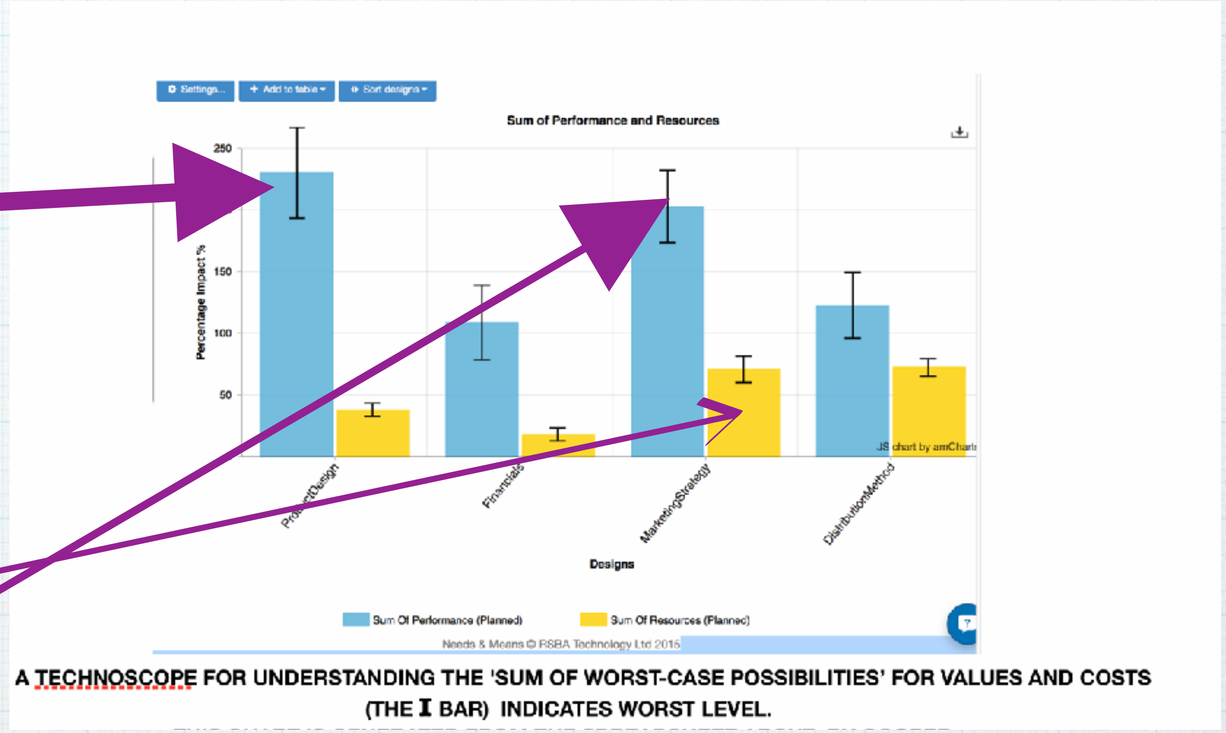


Figure 6 (Values Costs Uncertainties) This bar chart is generated from estimations of the sum of all critical values, and the sum of all critical costs, along with a notion of 'worst potential case' levels: the lower part of the I bar for values, upper part for costs. It gives a logical basis for prioritizing one of the 4 strategies. I would give priority to 'Product Design' because it seems to give a lot of overall value delivery, worst case, at low overall costs. Source Geoff Cooper, Startup Plan, on my BCS course London.

Prioritization Principles
(do not oversimplify priority)

1.INFORMED PRIORITIZATION:

- Deciding what to do next, needs to be done, when 'more cards' are on the table. More recent facts.
- When we know what *really* happened, at the *latest* increments of value delivery.

2. MANY FACTORS:

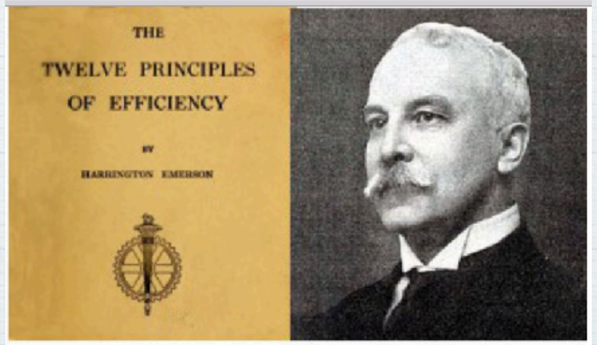
- Intelligent prioritization needs to consider *many factors*, (other values, stakeholders, resources, risks)
  - not the uninformed opinion of an anonymous 'weight setter'.

3. VALUE-FIRST PRIORITIZATION:

- Intelligent prioritization will be focussed
  - on *your* critical values first,
  - and secondarily your budgeted *resources*

"As to methods, there may be a million and then some, but principles are few. The person who grasps principles can successfully select their own methods".

Harrington Emerson



Book Cover

Harrington Emerson

Prioritization Policies
Your formally-stated 'culture'

1. SUCCESS PRIORITIZATION:
   - We will prioritize,
     - so that we *maximize* our chances,
     - to *succeed* in delivering critical values,
     - on *time*.
2. FACT-BASED PRIORITIZATION:
   - We will prioritize *incrementally*,
     - so that we gradually have the best facts
     - to prioritise wisely and *intelligently*, available.
3. COST-EFFECTIVE PRIORITIZATION:
   - We will prioritize so that we
   - incrementally deliver high *cost-effectiveness* to our organization.
4. SUPPLIER PRIORITIZATION:
   - We will enable our suppliers
     - to make priority decisions *locally*,
     - towards our values and resources.

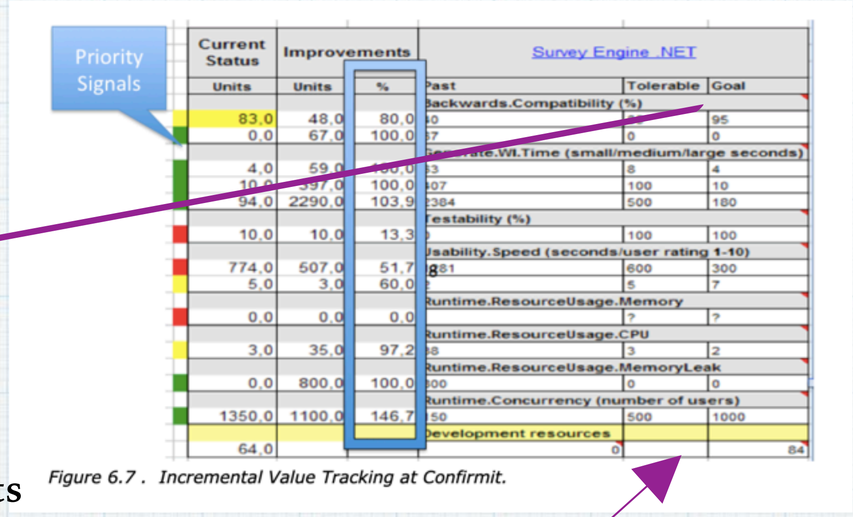


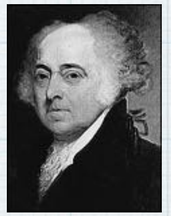
Figure 6.7 . Incremental Value Tracking at Confirmit.

"Facts are stubborn things;

and whatever may be our wishes, our inclinations, or the dictates of our passions,

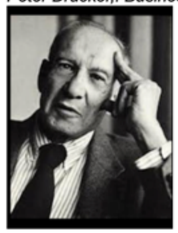
they cannot alter the state of facts and evidence."

--John Adams (1735-1826), 2nd President of US



"Work implies not only that somebody is supposed to do the job, but also accountability, a deadline and, finally, the measurement of results —that is, feedback from results on the work and on the planning process itself,"

Drucker wrote in Management: Tasks, Responsibilities, Practices. [19]
Peter Drucker,: Business Author (1909 Vienna - 2005 Claremont, California).



Prioritization Checklist
Some questions that might provoke some action

1. Have you quantified
   - all your critical values and resources,
   - so that you have a rational basis for prioritizing incrementally?
2. Are you set up so that a Value Decision Table (or IET)
   - can help you automatically calculate
   - your current priority?
3. Do you have a clearly-stated prioritization policy?
   - Like "Maximize critical values delivered"
4. Do you truly understand
   - what is wrong with
   - many conventional
   - 'fixed weights up front' methods?

Service Availability:
Type: Critical Product Line Objective.
Stakeholders: Service Personnel, Product Users, Consumer Associations.

Ambition: highest service availability in our business.

Scale: % of [**Instances**] for defined [**Users**] for defined [**Products**] and defined [**Problems**] where service can begin within the hour, and be completed within an hour.

Tolerable [Instances = Life Threatening, Users = Medical, Products = Body Monitors, Problems = Total Malfunction] 99.0%

OK [Instances = Life Threatening, Users = Individual Patients at Home, Products = Body Monitors, Problems = Total Malfunction] 99.5%

Goal [Instances = Unable to Modify, Users = Medical, Products = Body Monitors, Problems = Pain or Discomfort] 99.98%.

Stretch [Instances = Health Threatening, Users = Medical, Products = Body Monitors, Problems = Any Malfunction] 99.99%.

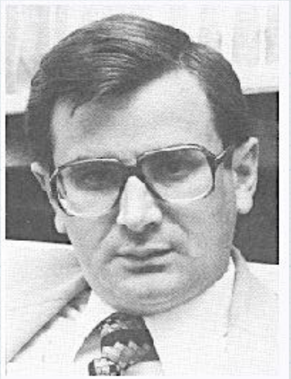
This value requirement gives a lot of information about priority levels of the value

Part 8.
Motivating people

Motivating Value Delivery

- Value delivery only happens
  - when management motivates people
  - to care about value delivery seriously
  - (quantified, measurable, responsible).

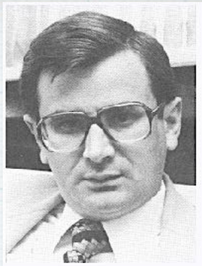
- 'Tom Is Right.
- We Are all Spouting Platitudes at Each Other,
  - And I Might Be the Worst One of You All.
- But This Is Going To Stop Immediately.
  - You Will Each Quantify the Top Values That You Are Working Towards, by End of This Week.
- If They Look Good,
  - And You Continue To Deliver Them,
  - You Will Still Have a Job and a Budget.
- If Not, Guess What'.



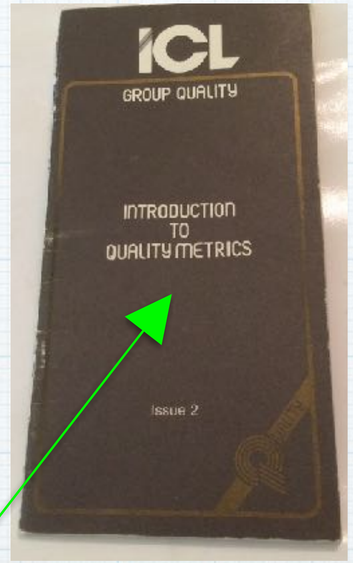
Robb Wilmot, ICL's energetic new MD

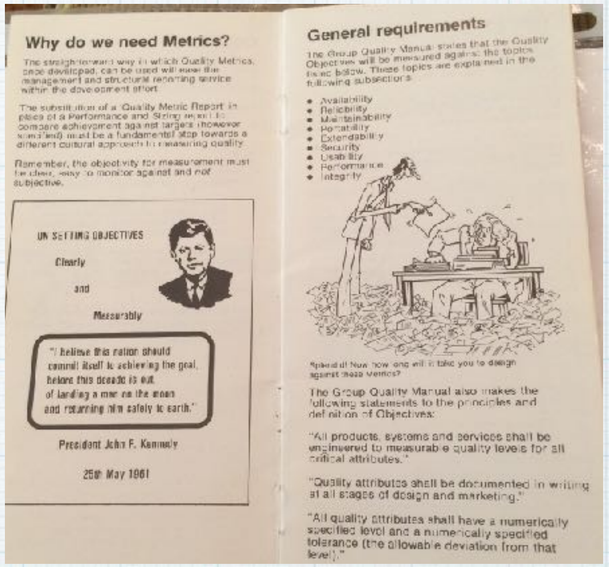
Motivation in practice.
Result: 15 years of profit. 2 Knighthoods

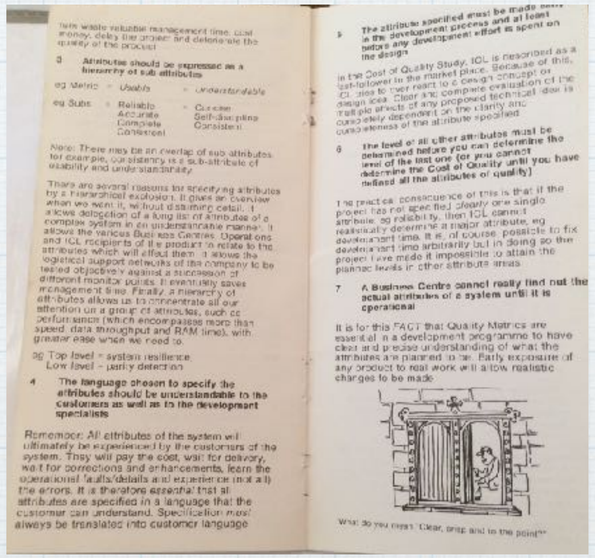
Case of Running a Corporation on my Quantified Values Ideas

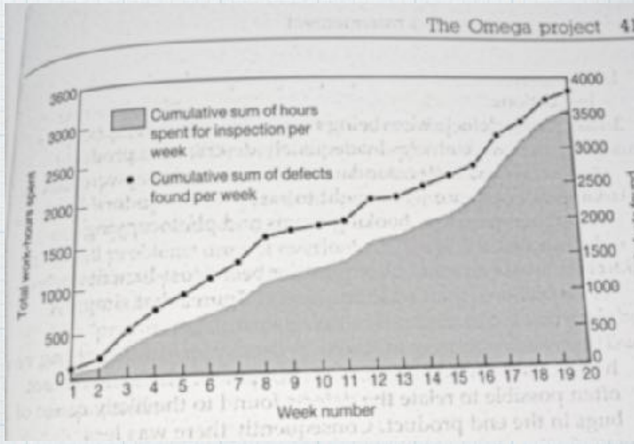

Robb Wilmot, ICL's energetic new MD

- Once, long ago, at a 20,000 to 33,000 employee computer company,
  - I managed to convince the to-be-successful turnaround CEO, Robb,
  - that he needed to run the show using *quantified* top-level values;
  - as opposed to shouting nice-sounding slogans, as management was doing.
- I helped quantify the corporate level values, like Viability. As well as many product qualities.
- The Board loved this stuff, and approved it (Gilb Methods) as one Corporate Strategy.
- They had 'never seen such clarity at a Board meeting', they said.









http://www.gilb.com/dl846

Bill of Rights for Clearer Company Communication

- At one stage, with this same large corporation,
  - I was trying to work out some standards for Value communication amongst the 20,000-33,000 employees.
- The people were highly-educated professionals,
  - and they did not like being told what to do!
- So I tried 'turning it around' to 'empowerment'.
- I put everything in terms of the powers and rights every employee
  - in fact had,
  - to question silly plans,
  - and to make better plans themselves
- The CEO loved this and made it public everywhere.

Bill of Rights

- You have a right to *know* precisely what is expected of you.
- You have a right to *clarify* things with colleagues, anywhere in the organization.
- You have a right to *initiate* clearer definitions of objectives and strategies.
- You have a right to get objectives presented in *measurable, quantified* formats.
- You have a right to *change* your objectives and strategies, for better performance.
- You have a right to *try out* new ideas for improving communication.
- You have a right to *fail* when trying, but must kill your failures quickly.
- You have a right to *challenge* constructively higher-level objectives and strategies.
- You have a right to be *judged* objectively on your performance against measurable objectives.
- You have a right to *offer* constructive help to colleagues to improve communication.

Original version in (Gilb 1988 Page 23)

You had have my permission to rip off my 'Rights' ideas. Tom@Gilb.com

http://www.gilb.com/dl846

Another Case, UK Software Corporation: the reluctant Directors Serious 'Motivation' at the top

- Later, in another 800 employee UK corporation, the CEO, David, decided to do the same as Robb.
  - He 'encouraged' his direct reports to quantify their top values to him.
  - He personally spent a whole day learning how to quantify and build Scales of Measure, tutored by me and Kai. He was very good at it.

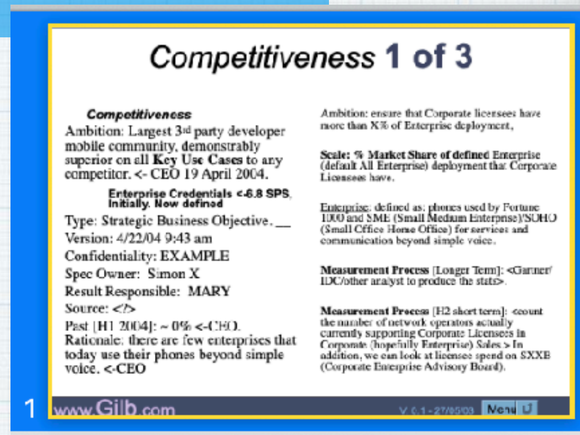
- All the directors played along, and gave lip service to this.
  - But half of them did nothing, and played for time,
  - hoping it would go away.

- A few months later, David simply fired about 5 of the Directors,
  - who did not act on his message.
  - And promoted people who were really good at using our quantified Value methods, to positions of power.

- I suspected some of those who 'left',
  - were happy to go elsewhere,
  - and sabotage their new employer, by unclear communication.

- It does demand a little bit of extra effort (half a day to a day) to write clear value goals.
- But the scary part for *incompetent* directors is that their inability to actually deliver their own level of value objectives will become clear, quickly.

Objective set by CEO personally —>


Competitiveness 1 of 3

Competitiveness
Ambition: Largest 3rd party developer mobile community, demonstrably superior on all **Key Use Cases** to any competitor. <- CEO 19 April 2004.

Enterprise Credentials <6.8 SPS. Initially. Now defined
Type: Strategic Business Objective.__
Version: 4/22/04 9:43 am
Confidentiality: EXAMPLE
Spec Owner: Simon X
Result Responsible: MARY
Source: </>
Past [H1 2004]: ~ 0% <-CEO.
Rationale: there are few enterprises that today use their phones beyond simple voice. <-CEO

Ambition: ensure that Corporate licensees have more than X% of Enterprise deployment.

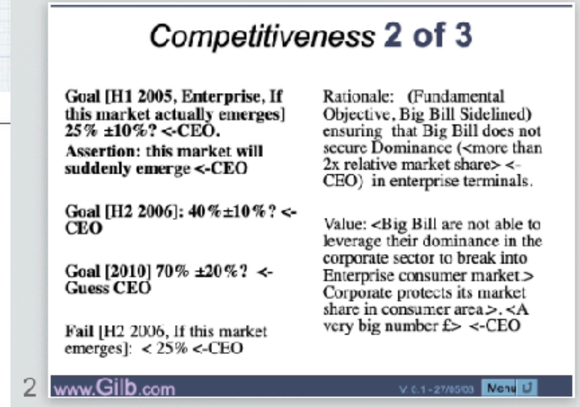
Scale: % Market Share of defined Enterprise (default All Enterprise) deployment that Corporate Licensees have.

Enterprise: defined as: phones used by Fortune 1000 and SME (Small Medium Enterprise)/SOHO (Small Office Home Office) for services and communication beyond simple voice.

Measurement Process [Longer Term]: <Gartner/ IDC/other analyst to produce the stats>.

Measurement Process [H2 short term]: count the number of network operators actually currently supporting Corporate Licensees in Corporate (hopefully Enterprise) Sales > In addition, we can look at licensee spend on SXXE (Corporate Enterprise Advisory Board).

1 www.Gilb.com


Competitiveness 2 of 3

Goal [H1 2005, Enterprise, If this market actually emerges] 25% ±10%? <-CEO.
Assertion: this market will suddenly emerge <-CEO

Goal [H2 2006]: 40%±10%? <-CEO

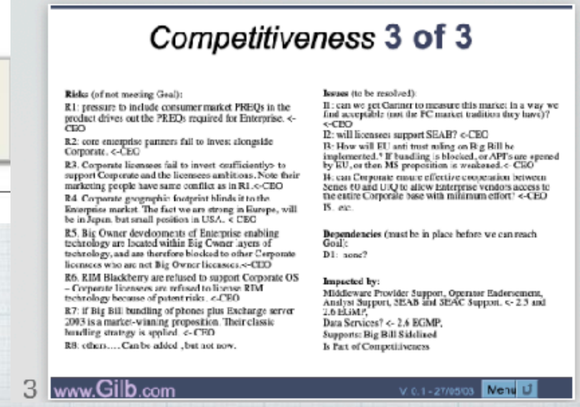
Goal [2010] 70% ±20%? <-Guess CEO

Fail [H2 2006, If this market emerges]: < 25% <-CEO

Rationale: (Fundamental Objective, Big Bill Sidelined) ensuring that Big Bill does not secure Dominance (<more than 2x relative market share> <-CEO) in enterprise terminals.

Value: <Big Bill are not able to leverage their dominance in the corporate sector to break into Enterprise consumer market.> Corporate protects its market share in consumer area>. <A very big number £> <-CEO

2 www.Gilb.com


Competitiveness 3 of 3

Risks (of not meeting Goal):
R1: pressure to include consumer market PREQs in the product drives out the PREQs required for Enterprise. <-CEO
R2: core enterprise partners fail to invest alongside Corporate. <-CEO
R3: Corporate licensees fail to invest sufficiently to support Corporate and the licensees ambitions. Note their marketing people have same conflict as in R1.<-CEO
R4: Corporate geographic footprint blinds it to the Enterprise market. The fact we are strong in Europe, will be in Japan, but small position in USA. < CEO
R5: Big Owner developments of Enterprise enabling technology are located within Big Owner layers of technology, and are therefore blocked to other Corporate licensees who are not Big Owner licensees.<-CEO
R6: RIM Blackberry are refused to support Corporate OS – Corporate licensees are refused to license RIM technology because of patent risks. <-CEO
R7: If Big Bill bundling of phones plus Exchange server 2003 is a market-winning proposition. Their classic bundling strategy is applied <-CEO.
R8: others.....Can be added ,but not now.

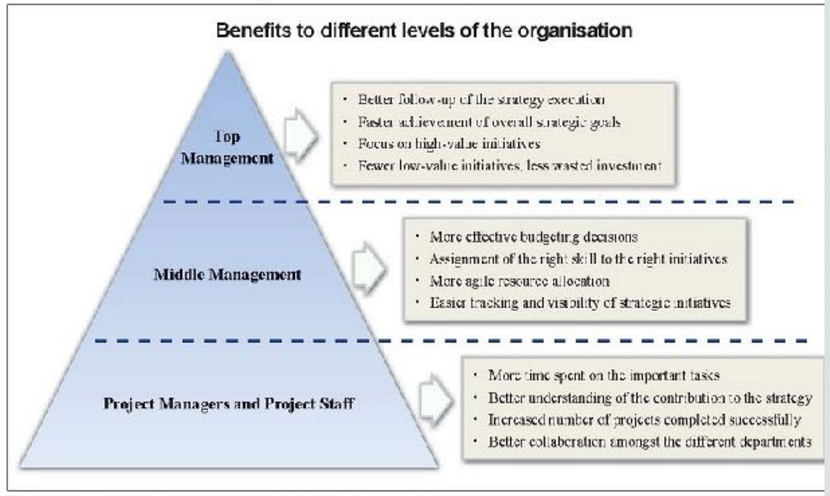
Issues (to be resolved):
I1: can we get Gartner to measure this market in a way we find acceptable (not the PC market tradition they have)? <-CEO
I2: will licensees support SEAB? <-CEO
I3: How will EU anti trust ruling on Big Bill be implemented.? If bundling is blocked, or APIs are opened by EU, or then MS proposition is weakened.< CEO
I4: can Corporate ensure effective cooperation between Series 6 and UIQ to allow enterprise vendors access to the entire Corporate base with minimum effort? <-CEO
I5: etc.

Dependencies (must be in place before we can reach Goal):
D1: none?

Impacted by:
Middleware Provider Support, Operator Endorsement, Analyst Support, SEAB and SEAC Support. <- 2.5 and 2.6 EGMP.
Data Services? <- 2.6 EGMP.
Supports: Big Bill Sidelined
Is Fact of Competitiveness

3 www.Gilb.com

Figure 1: Business Case—Qualitative Benefits

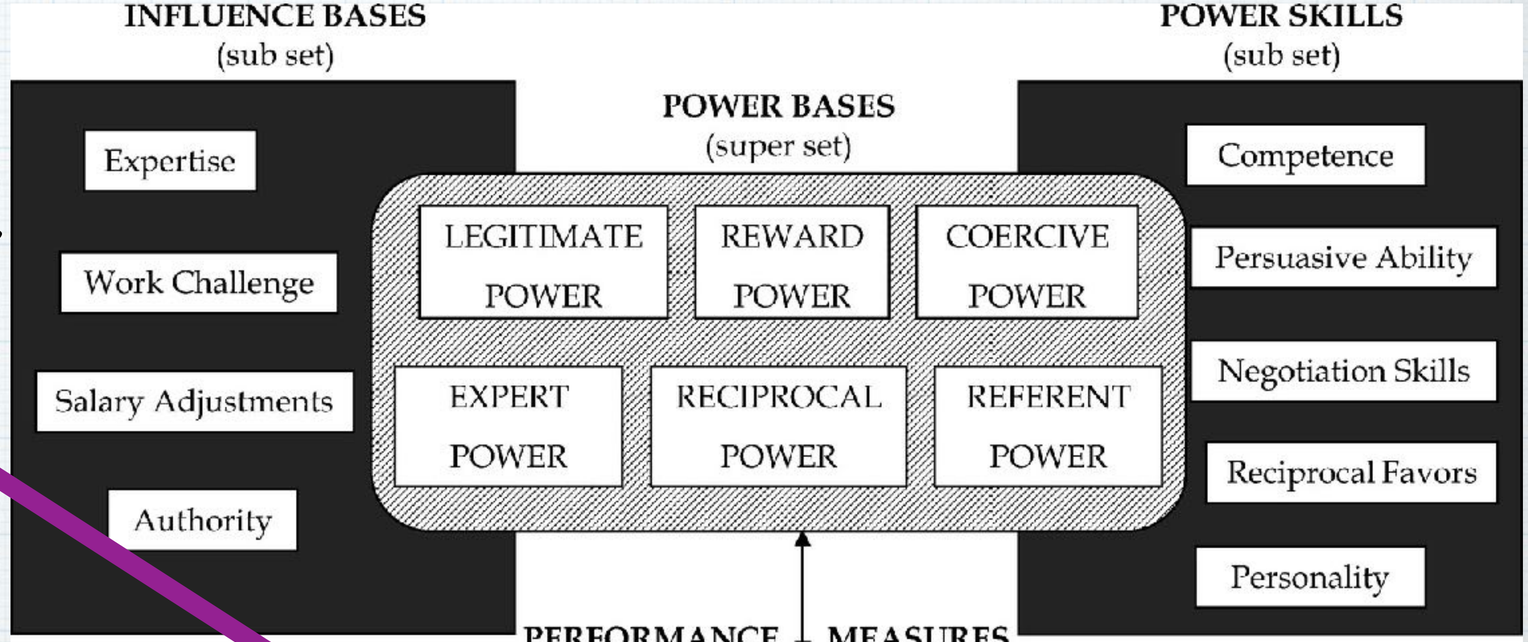


(Source: The Focused Organization (A NIETO-RODRIGUEZ))

The Competent Have Nothing to Fear, and much to Win

- Competent managers have nothing to fear.
  - Their ability gets documented.
- Incompetent managers intuitively understand that
  - these clear value objectives methods will unmask them,
  - and they will lose their job.
  - Very threatening. Not even nice, really.
- But the option is that
  - the whole company will go out of business
  - and *nobody* will have a job,
  - and the shares will be *worthless*.
- At the top management level,
  - you can't be nice,
  - or you will lose your own job
  - for bad Value delivery.

- Don't get me started on safe civil servants.
  - But I do like the way politicians get pressured by the media and electorate!
  - Politicians (ministers, mayors) need to learn to lead their civil servants better.
  - We are a very long way from being competent, I am afraid.



Principles for Value Motivation
Clear Goals are Key to Making it happen

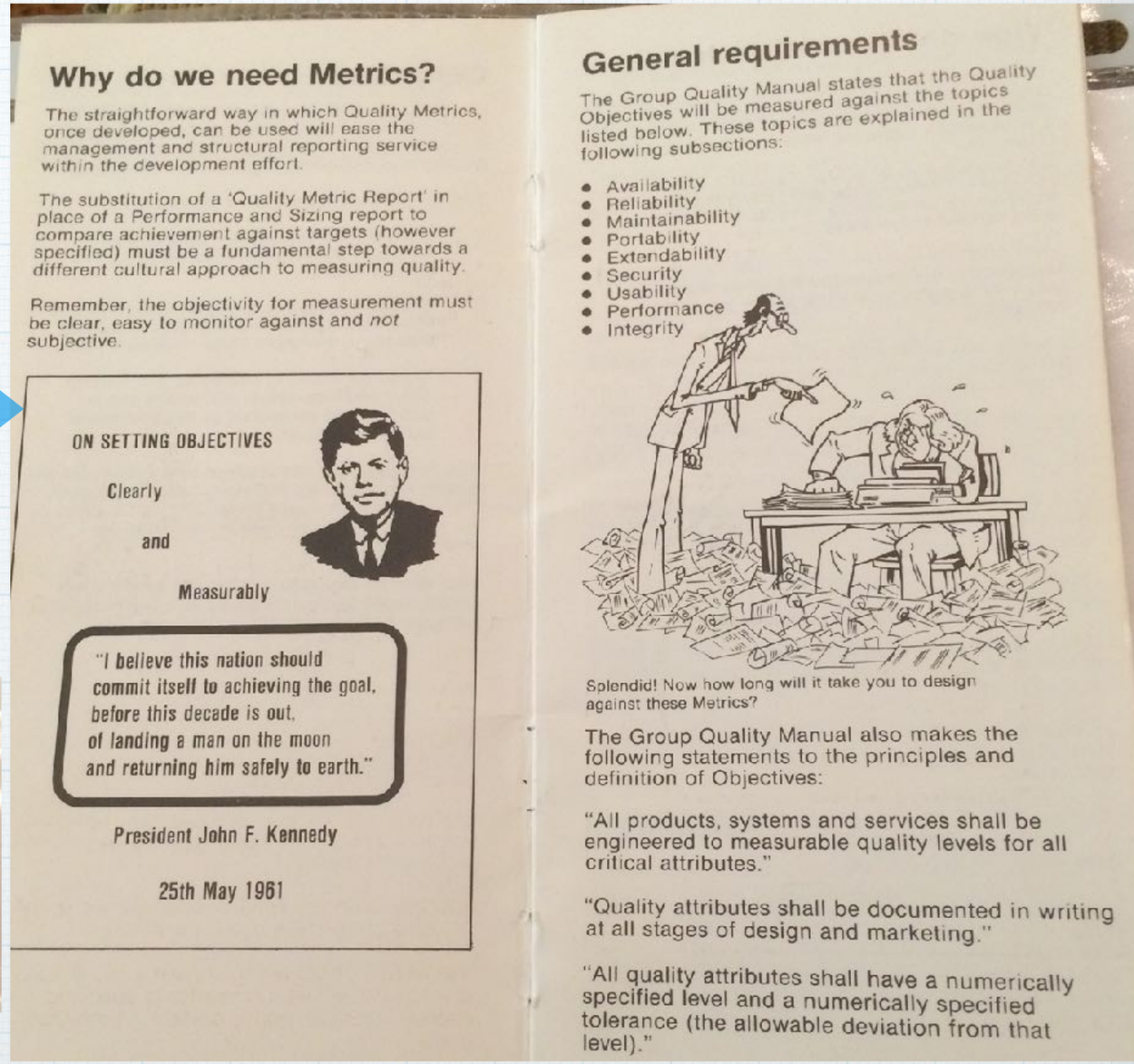
1. If the value is not *clear*, you cannot *motivate in the right direction*
2. *Incompetent* managers and professionals are quite happy to hide behind fuzzy value statements.
3. *Quantified Values* motivate suppliers, often through *contracts* and *payments*.

Policies For Value Motivation
Make Value Quantification and Clarity FORMAL and SUPPORTED

1. All critical values will be *clear, quantified, structured for detail, and measurable* in practice.

2. People will be *trained and equipped to quantify and measure* critical values

3. Real value delivery will be *rewarded, acknowledged, credited and honored.*



Why do we need Metrics?

The straightforward way in which Quality Metrics, once developed, can be used will ease the management and structural reporting service within the development effort.

The substitution of a 'Quality Metric Report' in place of a Performance and Sizing report to compare achievement against targets (however specified) must be a fundamental step towards a different cultural approach to measuring quality.

Remember, the objectivity for measurement must be clear, easy to monitor against and *not* subjective.

ON SETTING OBJECTIVES

Clearly

and

Measurably

"I believe this nation should commit itself to achieving the goal, before this decade is out, of landing a man on the moon and returning him safely to earth."

President John F. Kennedy

25th May 1961

General requirements

The Group Quality Manual states that the Quality Objectives will be measured against the topics listed below. These topics are explained in the following subsections:

- Availability
- Reliability
- Maintainability
- Portability
- Extendability
- Security
- Usability
- Performance
- Integrity

Splendid! Now how long will it take you to design against these Metrics?

The Group Quality Manual also makes the following statements to the principles and definition of Objectives:

"All products, systems and services shall be engineered to measurable quality levels for all critical attributes."

"Quality attributes shall be documented in writing at all stages of design and marketing."

"All quality attributes shall have a numerically specified level and a numerically specified tolerance (the allowable deviation from that level)."

Value Motivation Checklist
Would this Bill of Rights be a cultural disruption in your organisation?

- How clearly do you *credit* individuals and especially teams with value delivery?
  - Publish, talk about it, document it, award it.

- Have you made it clear that managers and teams
  - do not deserve,
  - and will not be given, *budget and responsibility;*
  - if they are not working towards,
  - and *delivering*,
  - *their* critical Value set?

- What would happen if you were to make use of the
  - Bill of Rights for Company Communication?

Bill of Rights

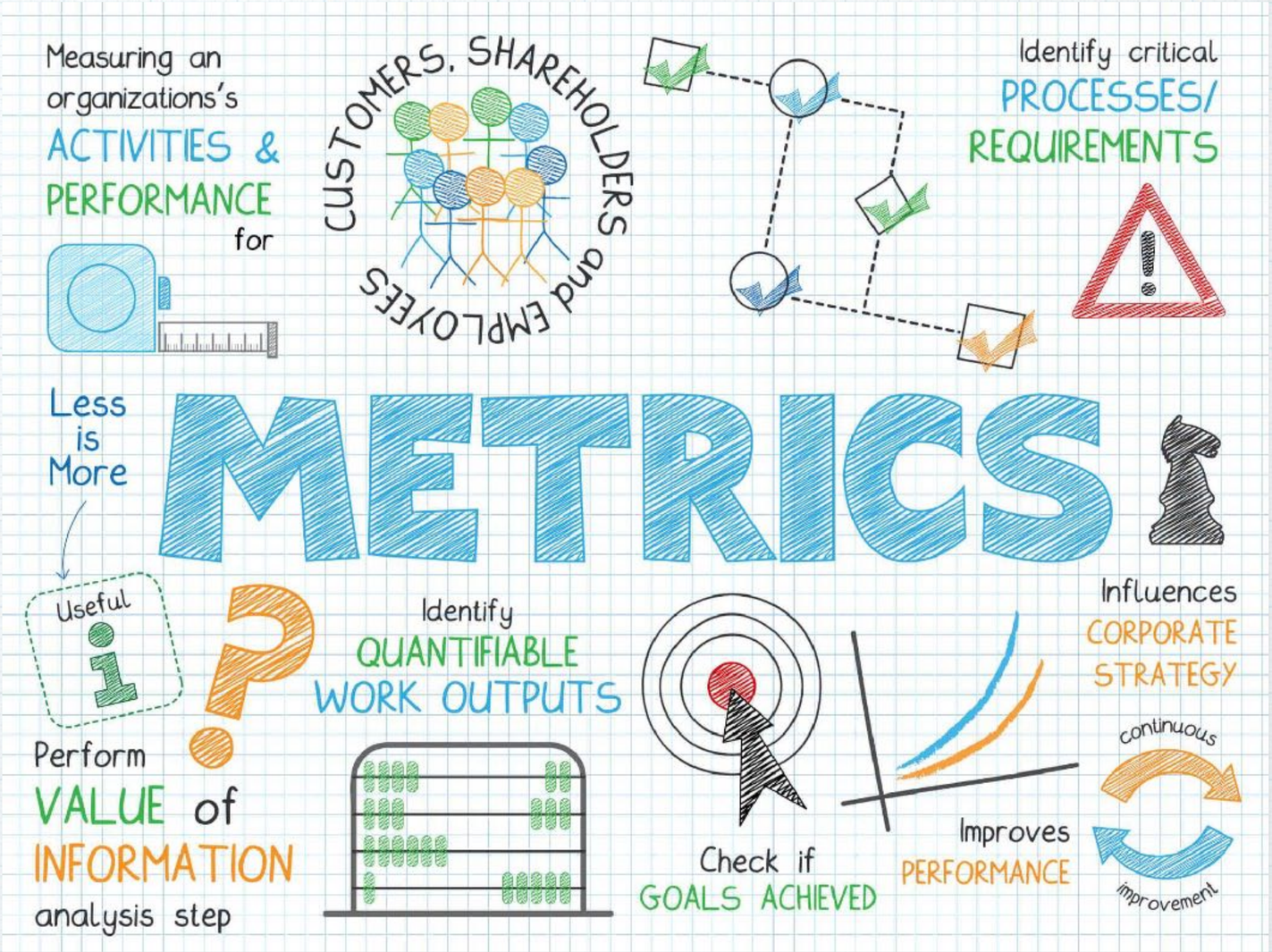
- You have a right to *know* precisely what is expected of you.
- You have a right to *clarify* things with colleagues, anywhere in the organization.
- You have a right to *initiate* clearer definitions of objectives and strategies.
- You have a right to get objectives presented in *measurable, quantified* formats.
- You have a right to *change* your objectives and strategies, for better performance.
- You have a right to *try out* new ideas for improving communication.
- You have a right to *fail* when trying, but must kill your failures quickly.
- You have a right to *challenge* constructively higher-level objectives and strategies.
- You have a right to be *judged* objectively on your performance against measurable objectives.
- You have a right to *offer* constructive help to colleagues to improve communication.

Original version in (Gilb 1988 Page 23)

Course Summary

Your guiding slogan can be

"QUANTIFIED STAKEHOLDER VALUES FOR MONEY"



Last Slide

* tom@Gilb.com
* www.Gilb.com
* @ImTomGilb
* www.linkedin.com/in/tomgilb
* +47 920 66 705
* Honorary Fellow of BCS

* Slides = https://tinyurl.com/ValueMgtBCS
  * (Dropbox Folder)
* Book = https://tinyurl.com/ValueMgtBook
* Video= https://www.youtube.com/playlist?list=PLKBhokJ0qd3_wlvr0j85YhmNfNj8ZJ8M-

* Permission to reuse all or some of my slides is freely granted, but
* Credit source ©: tom@Gilb.com 2020
* And nice if you email me a copy for my information.
*

* Kai Gilb Training Courses

https://www.gilb.com/valuefirst-requirements-online-course

https://www.gilb.com/lean-specqc

Pictures from Spring edition of the Masterclass

7 / 41

GO BACK TO LAST SLIDE