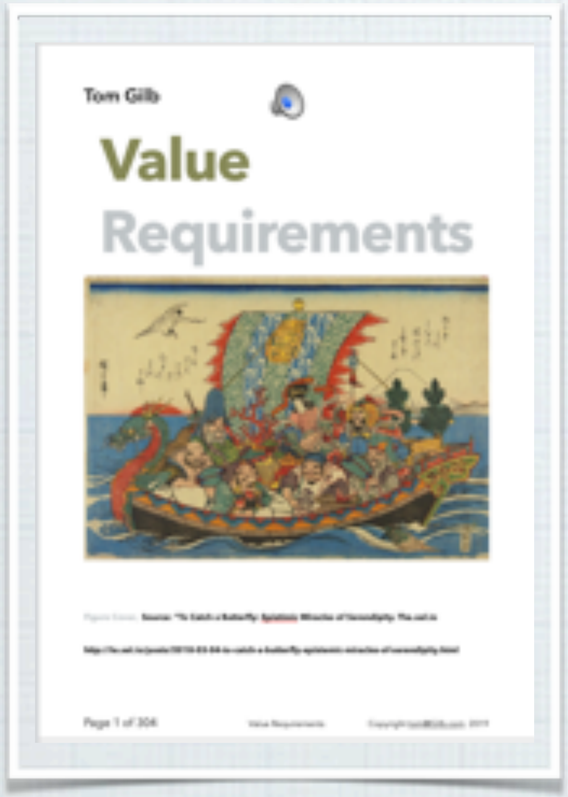
Value Requirements

(a 3 Hours BCS SPA SG ** Online Course)



For initial presentation Wednesday 22nd April 2020
Video URL Afterwards: URL= tbd
Slide Location: http://concepts.gilb.com/dl969. (check it)

https://www.dropbox.com/s/cl9h84tzelemzxd/Value%20Requirements%20PDF%20Good%20V%3D%20220420.0204.pdf?dl=0

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

** British Computer Society, Specialist Group SPA, Software Practice Advancement, http://www.bcs-spa.org/index.php

The VR Book

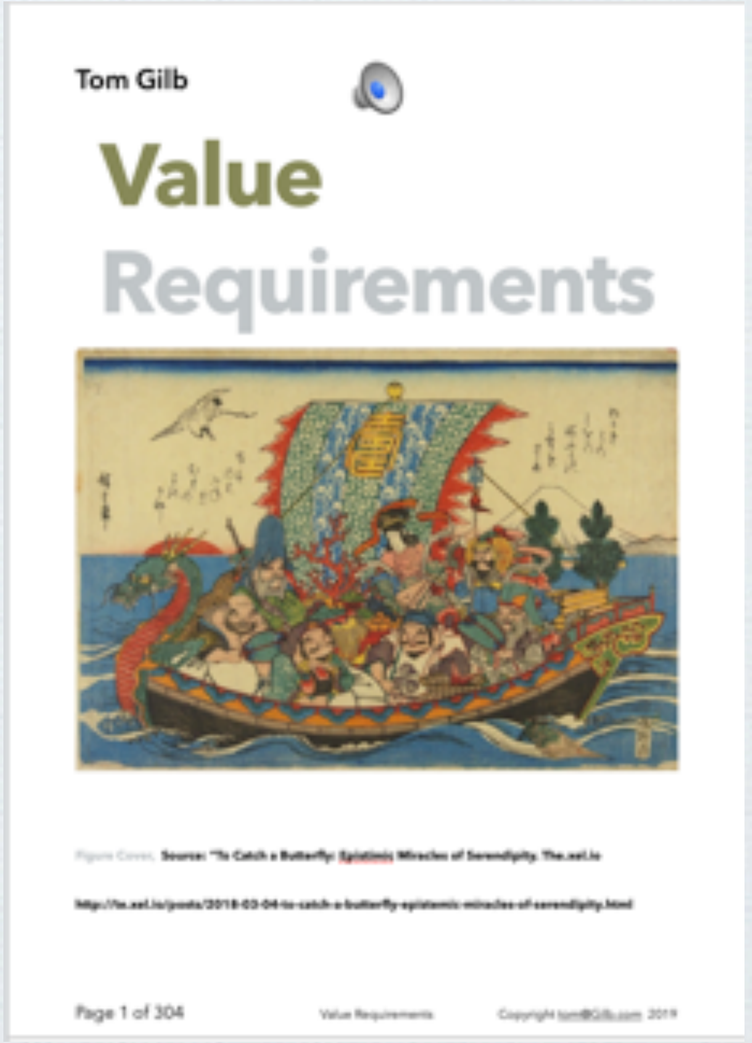
Main Subjects:

(these correspond to the book text)

1. What are value objectives?
2. Defining a Scale of Measure
3. Scale Parameters, for prioritisation and decomposition of complex values and environments.
4. Benchmark Levels of Performance
5. Requirement Level Constraints: Worst Case, and defining failure clearly in advance.
6. Requirement Level Targets: defining success in advance, knowing when to stop.
7. The Meter: practical measurement tools to control projects.
8. Background Specification parameters; for risk management, prioritisation, responsibility.
9. Making use of value specifications in practice.
10. Presentation of Value Specifications
11. Levels of Value Specifications
12. Defining Resource Levels, short term and long term.
13. Change Control of Value Specifications
14. A briefing on the use of value specifications for design and architecture

15 A Review of requirements methods compared to Planguage (lecture)

16. Formal standards for Value Specifications (lecture)
17. 'Valplan' and other apps for Value specification. Make your own app in Excel!
18. Stakeholders and the planning environment



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.

Part 1.
What are 'value' Requirements?

Value Requirements are the most important requirements for any project.

They are the <u>main purpose</u>, and main justification, for a project.

They are the *stakeholder's* values.

Value requirements start life as <u>value 'attributes' *needed*</u> by 'stakeholders'.

No project can deliver all 'needed' values, by a deadline.

No project will find all stakeholder values to be *worth* delivering.

So all value requirements start life by being <u>*acknowledged*</u> as *possible* delivery candidates. We call them 'Wish Level' statements.

But VRs need to go through an *evaluation process* to determine that we can prioritize them for real delivery.

Then we reclassify them as committed project requirements, which we call a <u>'Goal' level value requirement.</u>

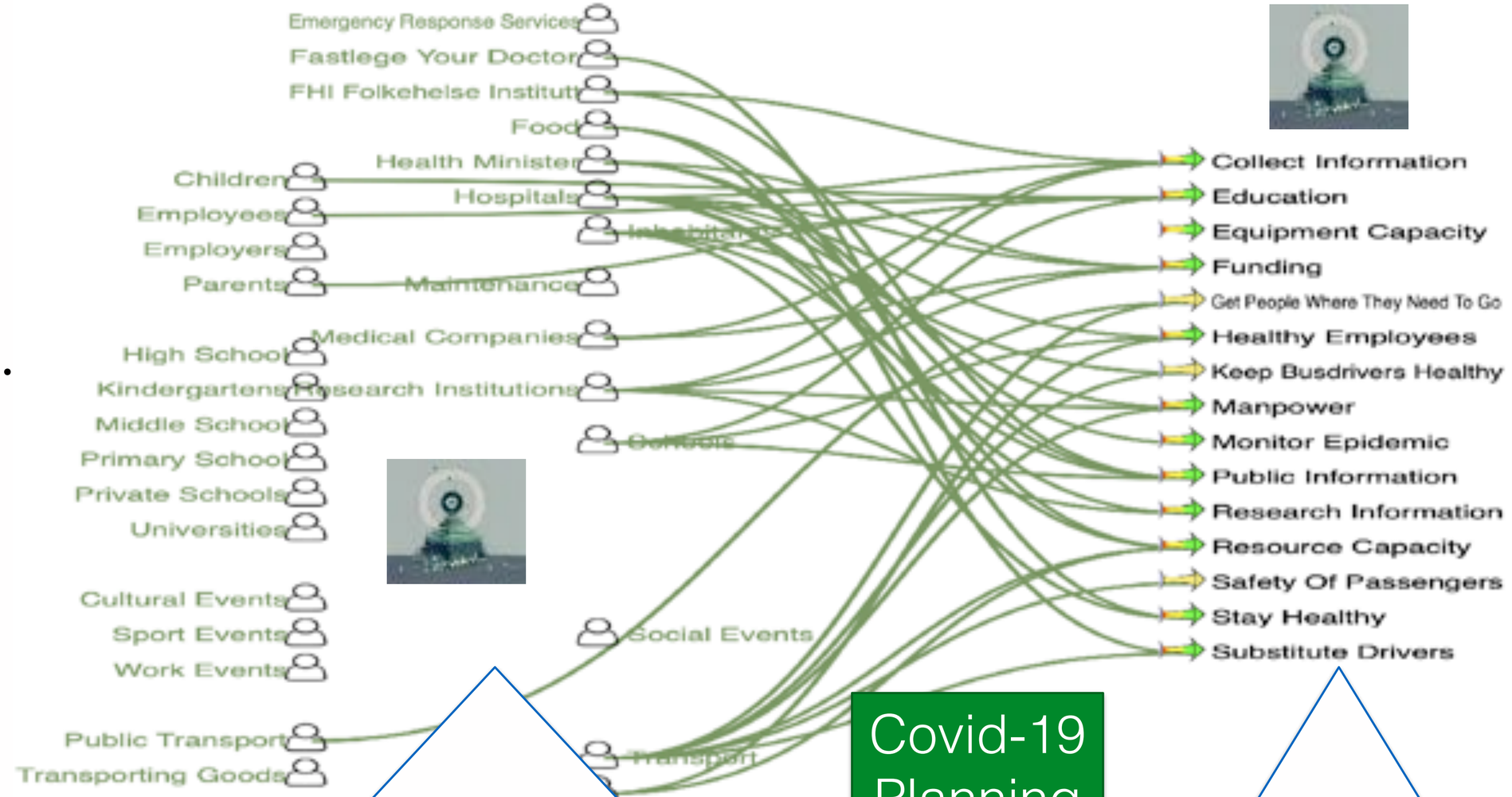
All 'functions' (what it does)
of any system,
Have a set of 'performance attributes'
Some of which are valued by some stakeholders.

Values are the primary, core, critical, key
Requirements for any system and any project.

So we should focus much better on the critical
stakeholder values

Critical few Values
Are valued by critical stakeholders.

Stakeholders decide what they value.



Stakeholders

Covid-19 Planning Exercise

Values

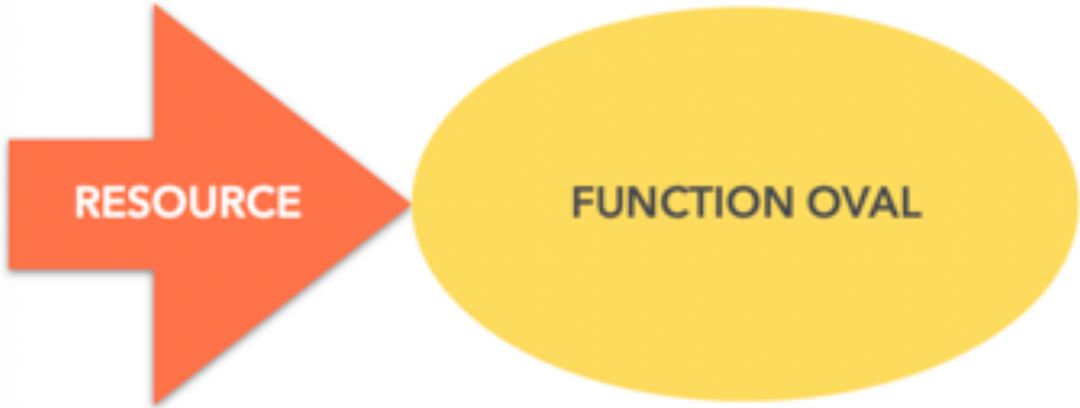
Other Requirement Types

Function Requirements: WHAT the system must DO.

The Function 'keyed icon' is: (any oval keyboard symbol)

Figure 1.2 : a system function, represented as an oval shape in Planguage icons.

Resource Requirements: limitations on any kind of resources (people, time, money).



Binary Constraints: legal constraints, design constraints, anything that must either be **done**, or **not done**.

Figure 1.4 : A constraint (the rectangular shape) which the system must be 'within'.

Planguage Requirements Definitions

Oct 2014 Full Planguage Concept Glossary
In Downloads/Gilb Essentials
http://www.gilb.com/dl830

Planguage Concept Glossary as edited in Competitive Engineering book 2005 (10% of Full Glossary)
http://www.gilb.com/DL387

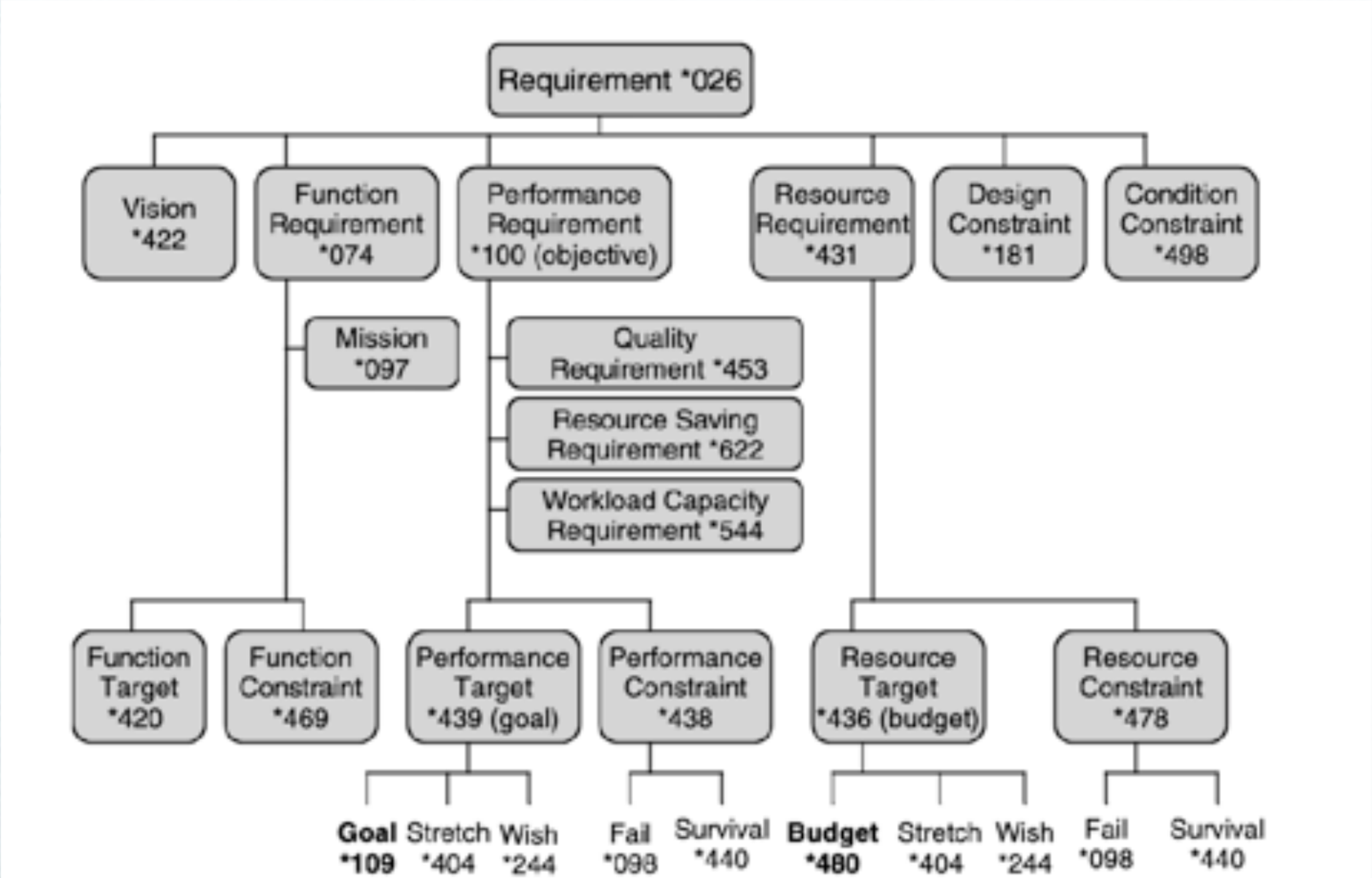


Figure G20
Requirement Concepts.

User Stories ?

Can 'User Stories' be used?

User stories do not contain enough information to serve as value requirements. But you can add on the sort of information we present in this book, to make them serve as value requirements

User Stories have a useful structure:

1. **A stakeholder (narrowly called 'user')**
2. **A 'story', which functions as a sort of requirement. But is not detailed enough for serious purposes.**
3. **A justification for the story, which is good practice. But can be improved upon.**

My initial advice for people who have to deal with User Stories is to write them up as a Planguage statement **, and then proceed to derive more-useful detail from it.

** I am (Spring 2020) working this out in more detail. Please email me for a copy of the ideas.

Usability:

Type: Value Requirement.

User Story: As an expert user I want shortcuts to save me time. <- US030719.

Scale: Average cycle time in minutes for a [Task] by a [User].

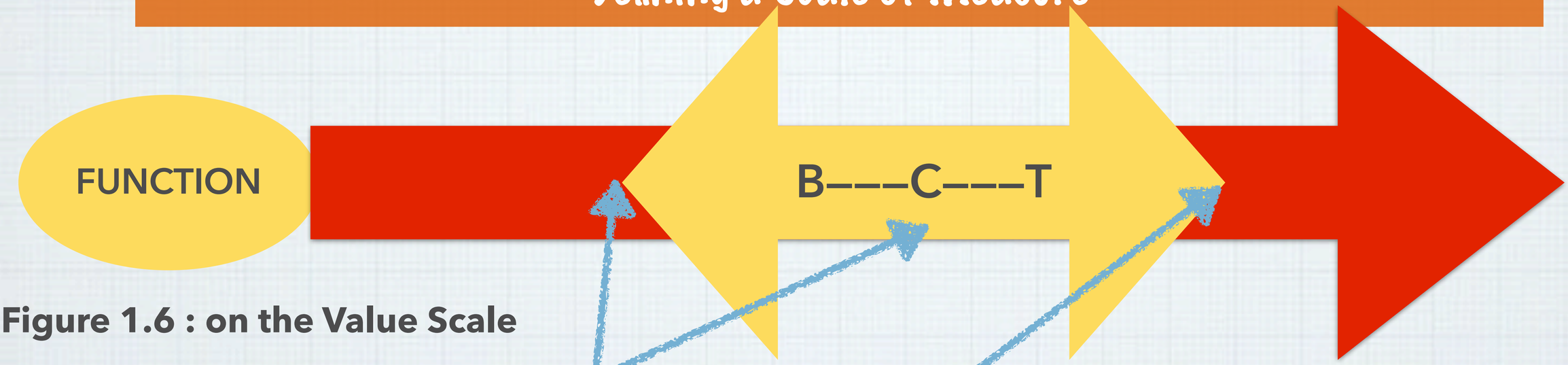
Wish: 6 minutes,

Deadline = End Next Year,

Task = Expert Complex Tasks,

User = Expert.

Part 2.
Defining a Scale of Measure



- Figure 1.6 : on the Value Scale
  - (Red Arrow symbol) a value *requirement* might be expressed by
  - numeric ***constraints*** (C, Worst case) and/or by
  - numeric ***targets*** (T, Success level defined).

  - In addition, a Benchmark level (B, or comparison) might be added to a requirement specification (not as a 'requirement', but as 'Background information')
    - to inform us of *past and current* levels of performance, (us, competition, state of art)
    - for comparison with the *required* levels. (C as minimum, T as success level)

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)

Scales (Part 2), and ‘Scale [Parameters]’ (Part 3),
for *prioritisation*, and *decomposition*
of complex values, and complex environments.

- ‘Scale’ is the core technique
- We *alway quantify* all value requirements
- Scale can be used to formally *define* success, completion
- Scale can be used to *define failure* levels of a value
- Scales give *clarity*, and *teamwork basis* (shared idea)
- Legal specification basis (contracts, bids)
- Scale useful for *geographically spread* projects.
- Scale is a *quantification* idea
  - (not ‘*measurement* proess’ that is a ‘Meter' spec)
- Scale is reused, and used in common for many purposes.

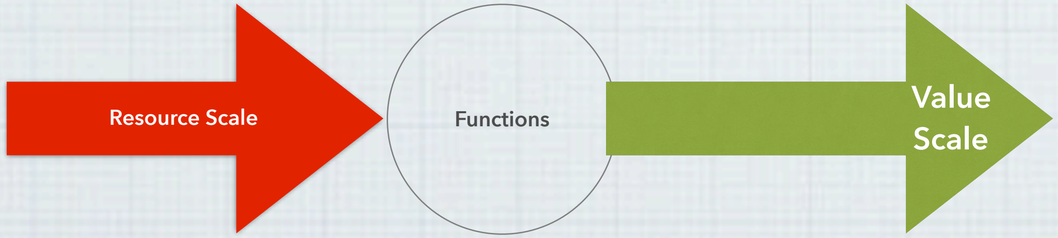


Figure 1.8 : Good value Requirements help us avoid project failure. All 5 of these failure causes are actually related to good ‘value requirements’.

Source: http://mobile.baselinemag.com/project-management/slideshows/why-some-companies-have-more-successful-projects.html

Simple Scales

Here are *the minimum* attributes of a Scale parameter specification:

- **Must allow scale numbers to have a 'useful meaning', when associated with the scale,**
- **Should not be so short a Scale-specification, as to leave critical concepts undefined, or to be ambiguous to any reader.**

Here are some desired attributes of any Scale specification:

- **It should be *intelligible* to domain specialists,**
- **Should be a good *reflection of the value*, as perceived by the domain specialists, and other relevant stakeholders.**

Usability:
Scale: % of users who can master the basics within first day of use.

Impressiveness:
Scale: % of people who took a test ride, who then joined a waiting list within a week.

Developing a Scale from an Ambition Level.

- An **Ambition Level** is an informal statement of a requirement, about one sentence long.
- It often comes from an ‘official’, attributed source.
- The problem is that it is filled with ambiguous terms, and does not lend itself to quantification.
- So it becomes our job to clarify and quantify ‘His Masters Voice’.

Here is an example of an ambition level:

Ambition: “before performance, Tesla prefers to focus on safety first’ <- Elon Musk. 140319

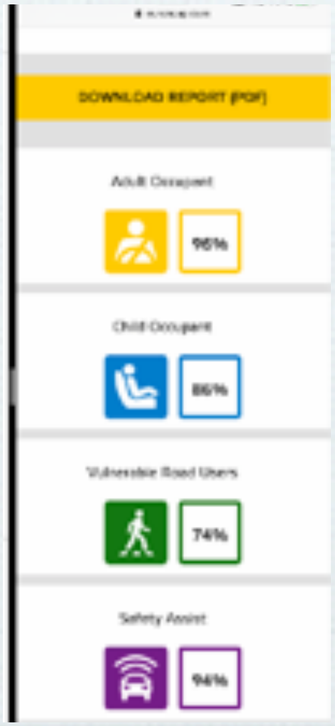
And here is a Scale we can derive from that:

Scale: % average passenger safety rating by Euro NCAP

Developing a Scale from an Ambition Level.
(Scale development is an art, requiring knowledge and practice)

To derive that Scale we had to think:

- **What kind of safety ratings give useful objective proof of safety?**
- **What units of measure are used in them (stars, % survival) ?**
- **Which units of measure, if there are several, serves our purposes in this project?**
- **Some searching on the the internet (Tesla, Safety Ratings) might give specific options of ideas.?**
- **Would the power-that-be (Musk) think this is a good scale of measure for *his* purposes?**
- **Is our suggestion broad enough for purpose, or is it unnecessarily narrow?**
- **Does it cover all market areas?**
- **Does it cover all types of the value (safety)?**

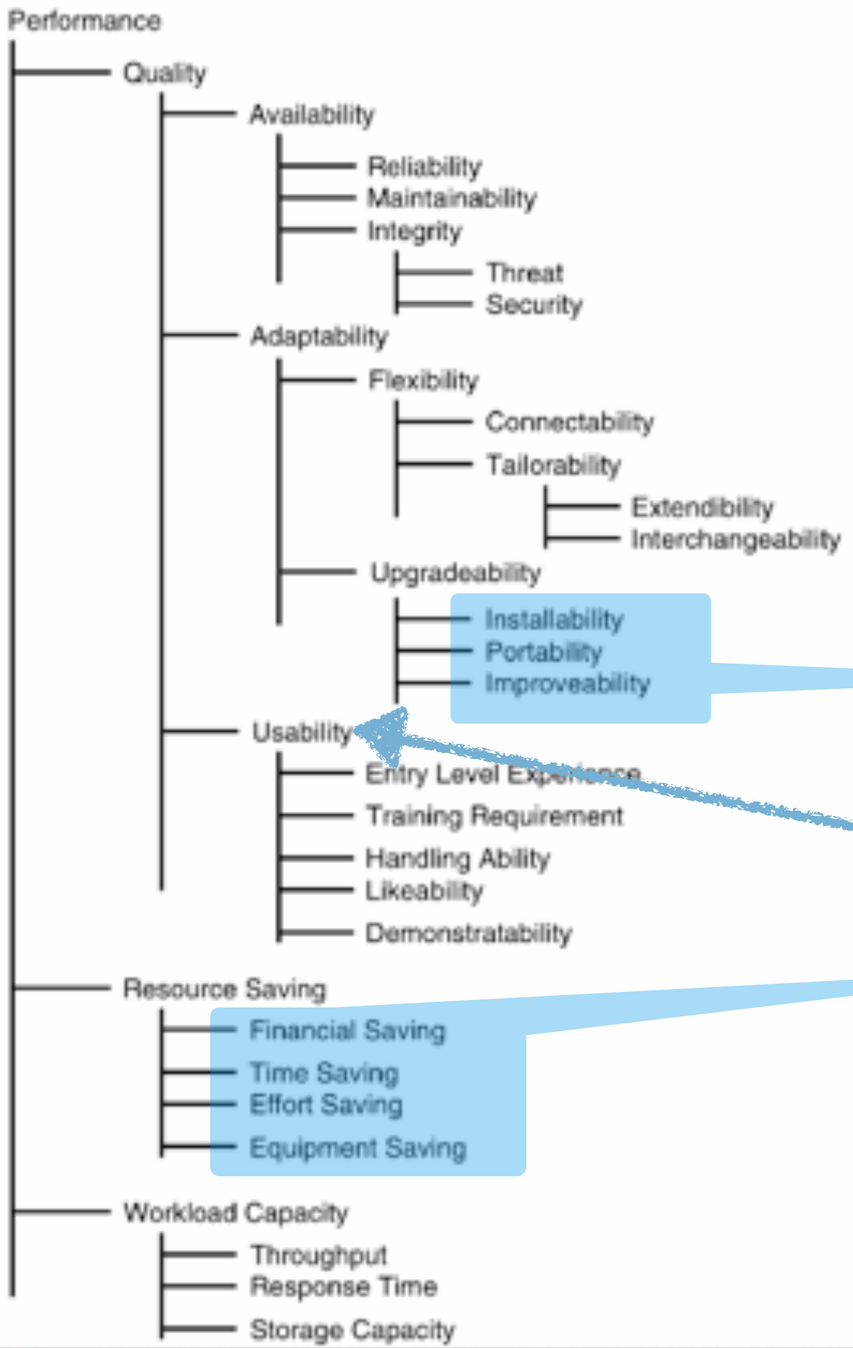


Here is an example of an ambition level:

Ambition: “before performance, Tesla prefers to focus on safety first’ <- Elon Musk. 140319

And here is a Scale we can derive from that:

Scale: % average passenger safety rating by Euro NCAP



Knowing when to decompose a value into sub-values, using sub-scales.

There is an engineering heuristic that says 'decompose the value you want to quantify until 'quantification is obvious'.

This works well.

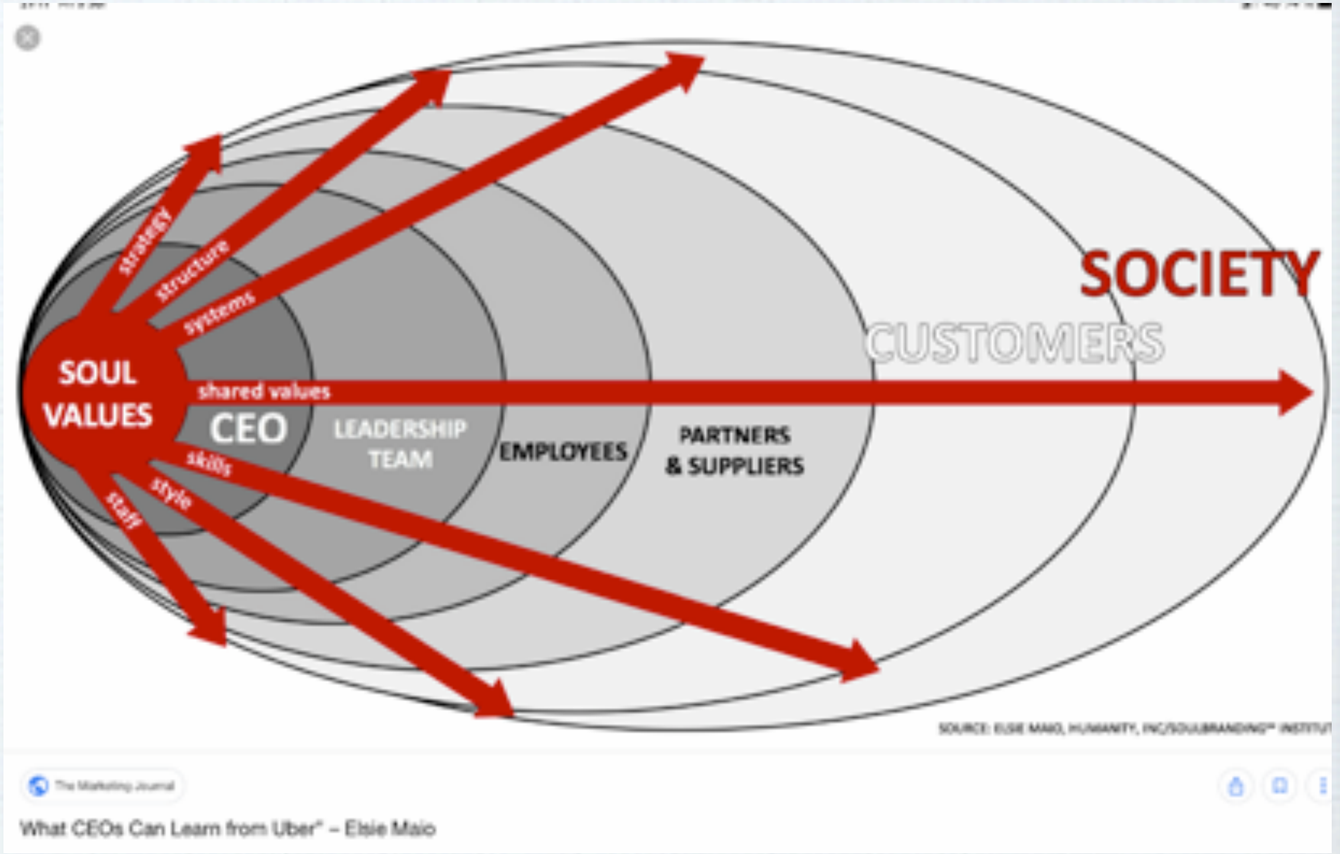
Each detailed level is defined by its own tailored Scale of measure.

The root level (or, 'set name') is defined by the 'set' of quantified detail elements.

CE book, Chapter 5: Scales of Measure:
http://www.gilb.com/DL26
Gives detailed examples of template Scales for everything in this diagram.

Good and 'useful' Scales of measure:

- **Some values might be more difficult to measure in practice,**
  - **but you should never choose a <u>value Scale</u> *because* it is 'easier to <u>measure</u>'.**
  - **You must first choose a <u>*relevant*</u> Scale,**
  - ***then* try to optimise its costs of measurement (to be 'worth it'). Its 'Meter'.**
  - **Cheap measurement tools (Meters), of a badly correlated Scale, are a waste of time.**
- **Scales will be highly-correlated to the real environment, (cause and effect)**
  - **with plenty of necessary [Scale Parameters]**
  - **to specify realistic and critical dimensions of the value's application or environment.**



Alignment levels and related concerns.

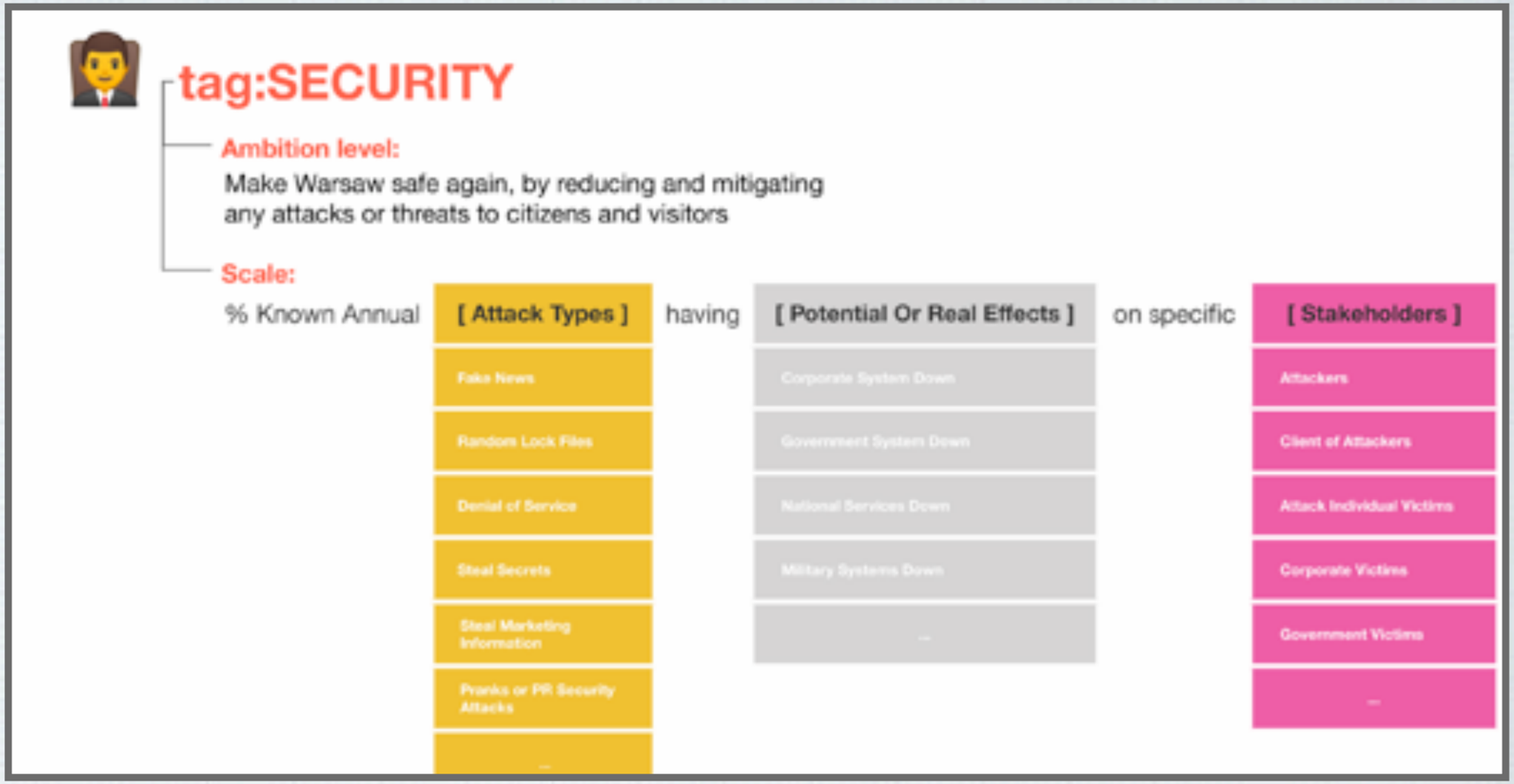
Part 3.
'Scale [Parameters]'

An advanced Scale definition technology.

If we want

- **Accurate modeling of large and complex systems**
- **The possibility of separating critical value-deliveries from less-critical ones,**
  - **… which permits us to 'do critical stuff early'.**
- **The possibility of much better requirement definitions,**
  - **…. so nobody can possibly misunderstand (like contractors and suppliers, and even managers).**

Then you will want to use this to improve the 'resolution' of the Scale tool.



Source: anna@Karlowska.PL, 2019

Vehicle Safety:

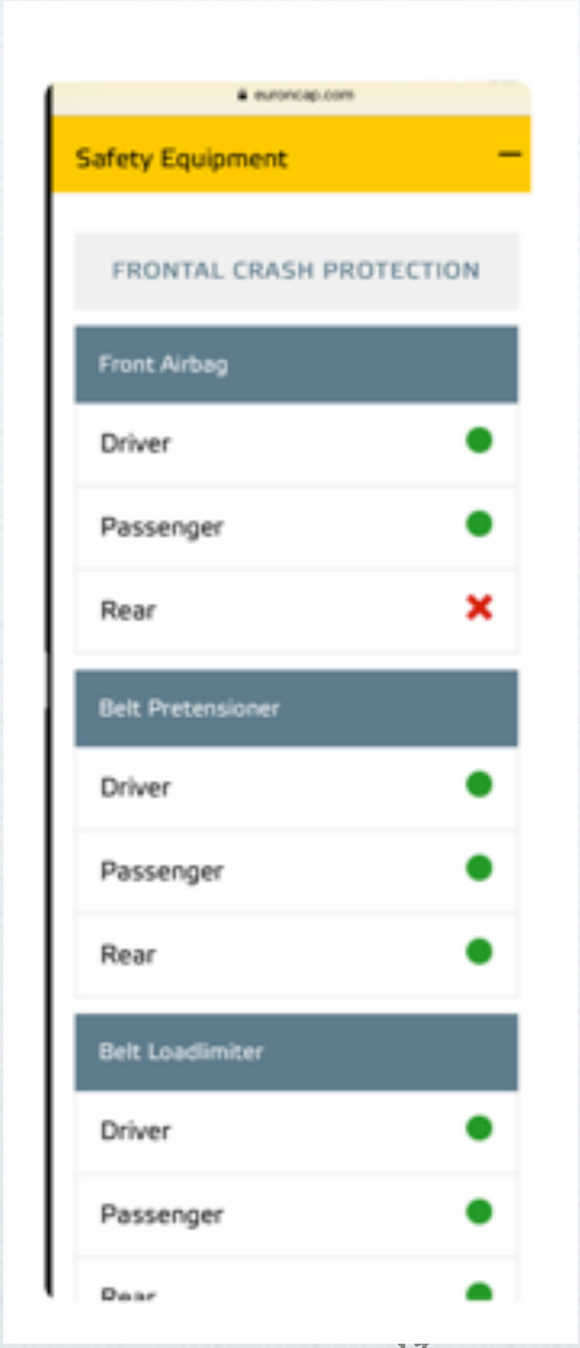
Scale: Star Rating number for [**Person Type**] and [**Car Specs]** for [**Safety Equipment]** with [**Alternative Model Validity**] for a [**Publication Date**] by a [**Rating Agency**].

Scale [Parameters]

The terms in [Square Brackets]

(e.g. '[Car Specs]') are:

- Formally defined terms: The **C**apital **L**etters signal that they are defined, somewhere
- [General Concepts]: defined with a specific set of elements, which as a set, define the general concept.
- For example:
  - **People** = {Babies, Children, Adults, Aged}.
- We are going to sometimes use the *set* '{...}' parenthesis, to list a {***set}*** of things (as above example).



Vehicle Safety:

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 Pre-tensioner, Belt Load Limiter, Knee Airbag, Side Head Airbag, ...},

Alternative Model Validity=Dual Motor AWD Model 3,

Publication Date =2019,

Rating Agency= All.

Scale [Parameters]

- **[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'.**

<u>Vehicle Safety</u>:

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,

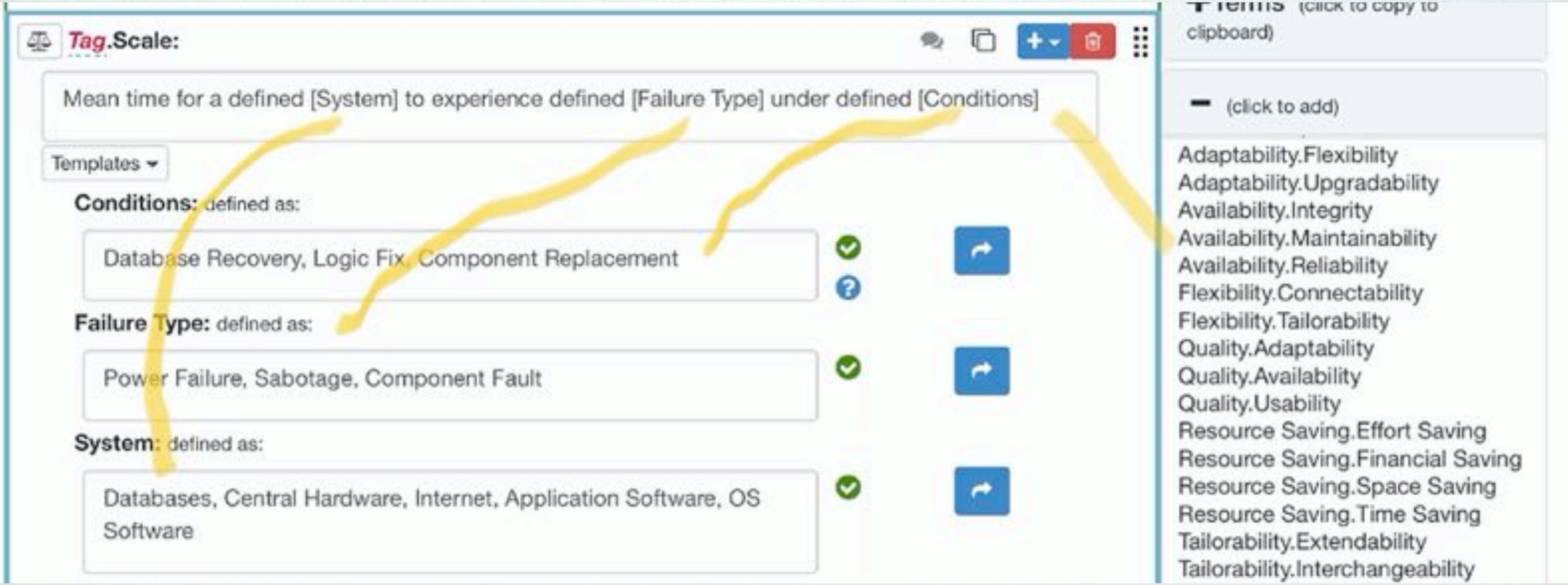
Publication Date =2019,

Rating Agency= All.

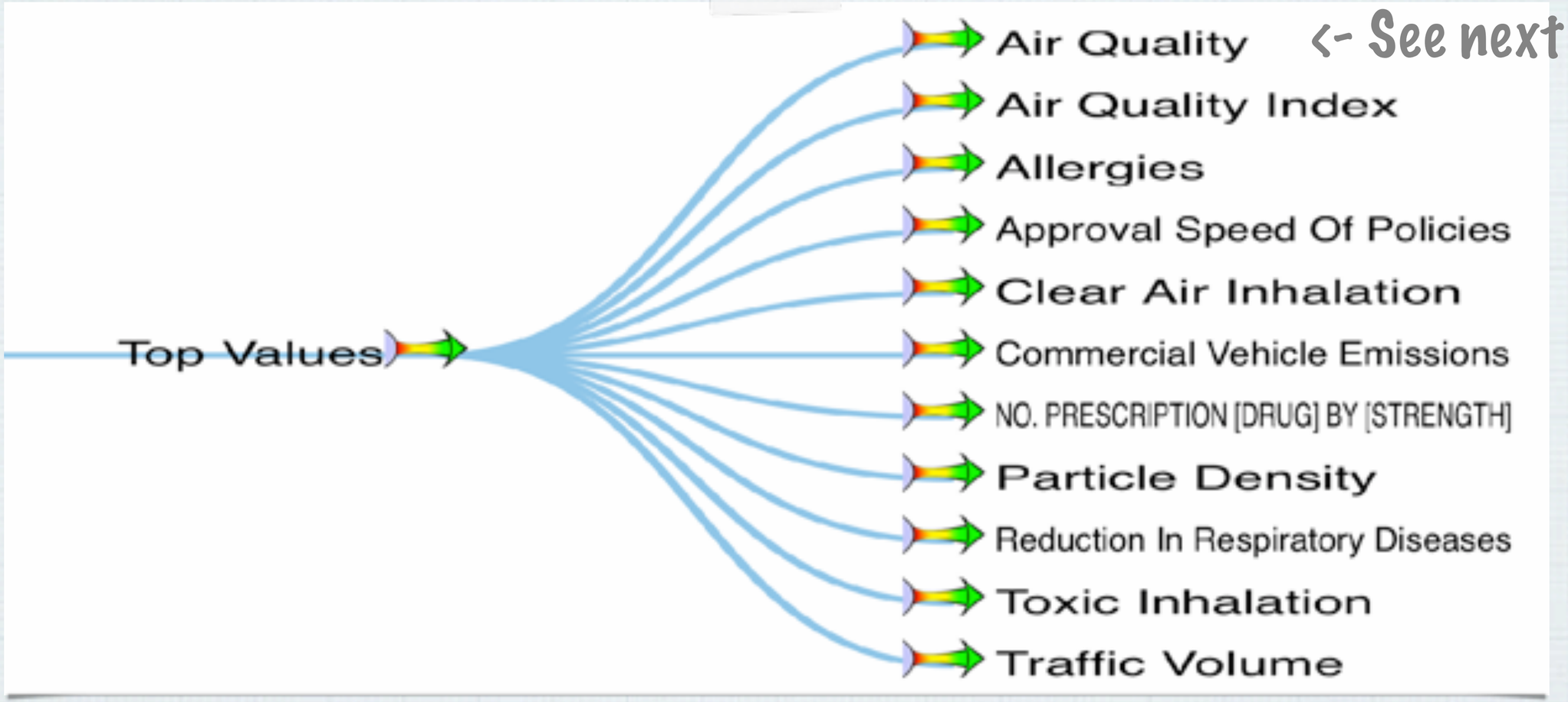
Scales of Measure From a library (in ValPlan.net)

CE Book Chapter 5, Scales of measure. Gilb: Competitive Engineering, 2005. ——————>
http://www.gilb.com/DL26

Defining Scale [Parameters]
We can define the [Scale Parameters] locally



Using tool valplan.net



<- See next 2 slides

Example of focussing on a few critical Value Requirements, at a given level of concern

Source BCS Exercise Sept 2017, 'London Congestion for Air Quality'. Notice that this is also a definition of 'Project Value' using the 11 decomposed different values, as the definition-by-subset. We could call it the Value Set. It is a prioritisation method, to choose some and not others to focus on, for the time b being.

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

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

<- next slide for detail

Detailed 'Value Requirement' specification

(1-line/statement view)

One of the critical value requirements in previous slide

"Air Quality'

Source BCS Exercise Sept 2017, 'London Congestion for Air Quality'. Notice that this is also a definition of 'Project Value' using the 11 decomposed different values, as the definition-by-subset.. Figure 1.19 in Value Requirements book

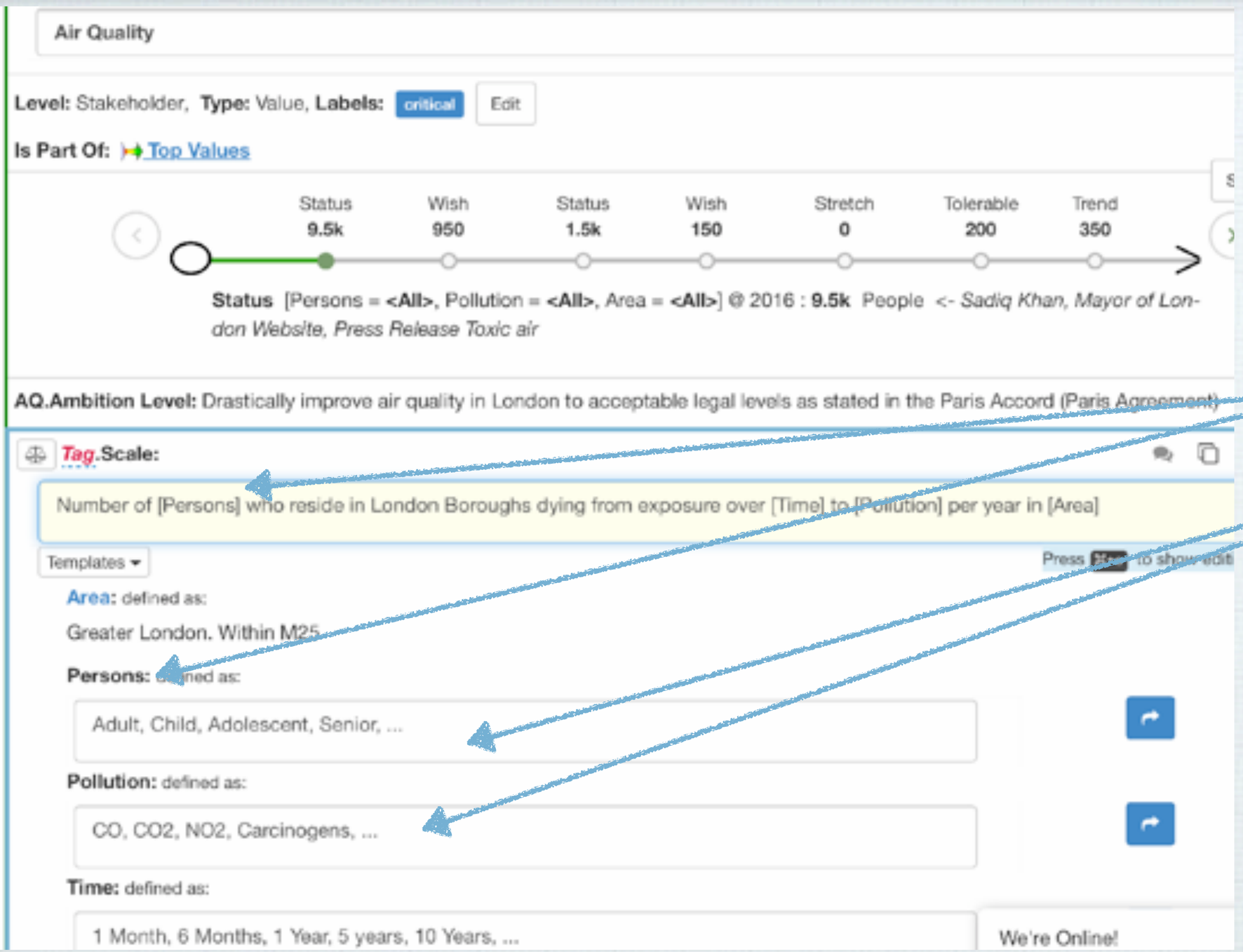


Figure 1.20 :

Here is a view of the 'Air Quality' value spec

with detail of the Scale.

You can see that the [Scale Parameters] are defined as

a *set* of attributes.

'Scale' spec is displayed in a 'window' <———For more detail

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

Figure 1.20 in Value Requirements book

Part 4.
'Benchmarks'
Levels of Performance, Systems analysis, Background to requirements. Potential 'points on our Scale'

* The purpose of 'Benchmarks' In a Value Requirement Specification.
* **A 'Benchmark' level of value, on a Scale of measure, is *background* information about a requirement level.**
* **It helps us decide if we have set the future requirement levels, appropriately.**
* **This is traditionally something a Business Analyst should look at, as a prelude to setting requirements.**
* **But in Planguage, I decided that it was better to *integrate* Benchmark data with the requirements data.**
    * **Both as a logical basis for setting requirements, and a basis for adjusting and improving requirement levels, to keep up with change.**
* **in order to make it possible for all reviewers and creators of a requirement object, to decide for themselves, if the future requirement levels are in reasonable proportion to the benchmarks**
* **To make it even clearer if the Benchmarks data is *missing* from your analysis process, or not particularly credible, or not up to date. In your face, not long forgotten and invisible.**
* **To support incremental value delivery,**
    * **where Benchmarks need to be updated, at each value increment, ('Status')**
    * **not just in an initial Waterfall analysis phase ('Past' levels).**

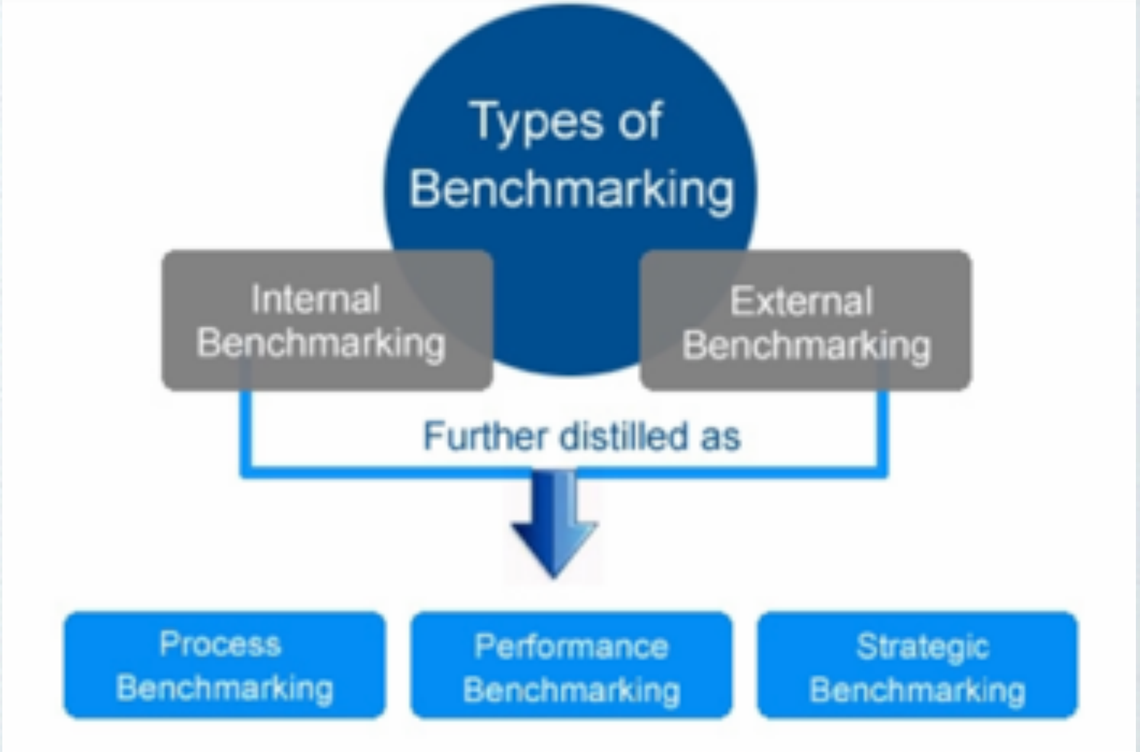
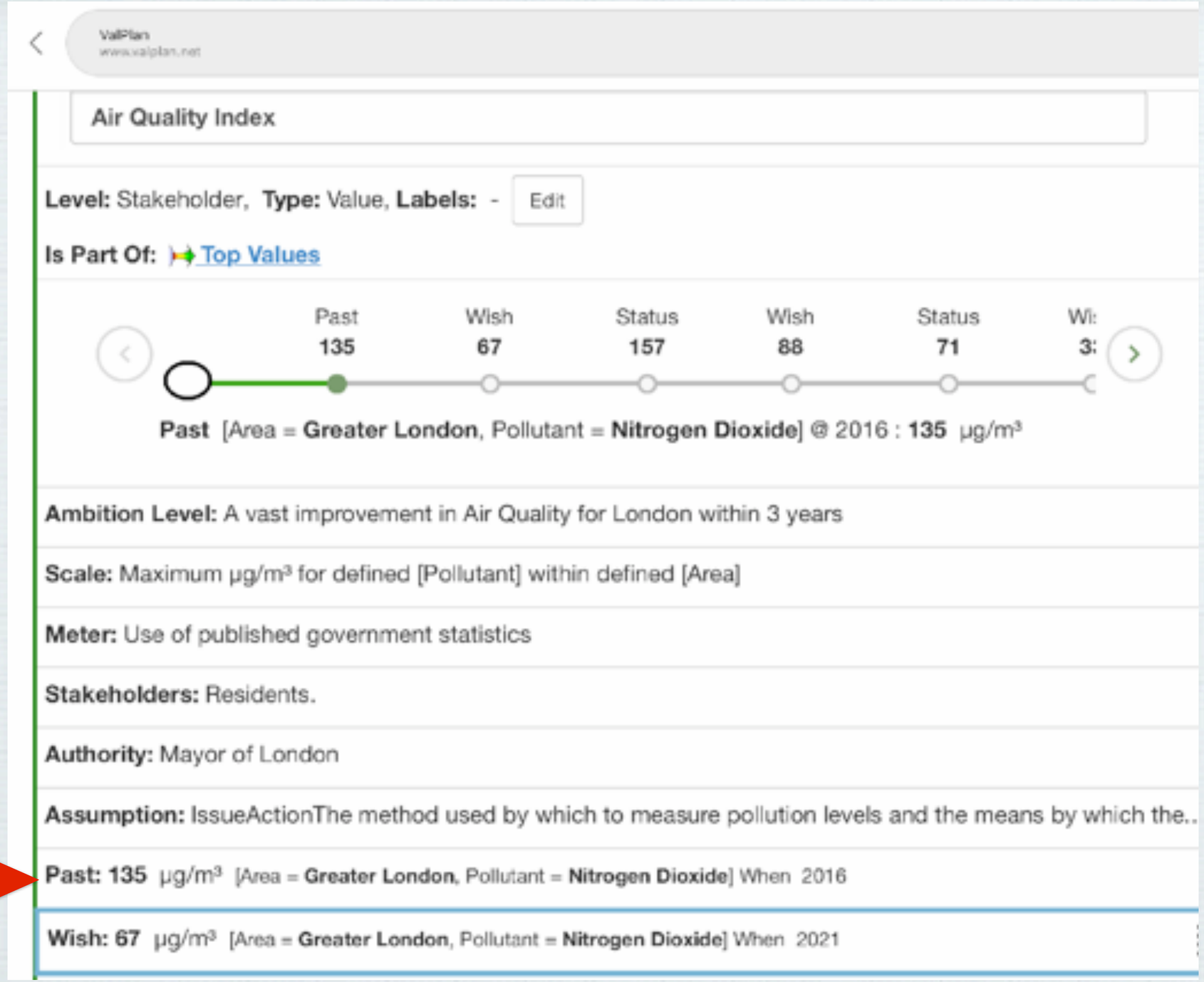


Figure 1.24 :



'Past' Level as a benchmark

- 'Past' data here, for Greater London and Nitrogen Dioxide, are directly comparable in the 2 Scale parameters, and with the units of measure on the Scale.
- The *requirement (Wish)* is for a 2x reduction over a 5 year period.
  - Compared with the 'Past' level
- As it is when planning, 2019,
  - we need to ask if the Past data is up to date
  - (at least last year, or hopefully this year to date)
  - and if any progress has been made as a result of our project deliveries, if any.
- Maybe is time now (say 2020) to introduce a Status specification to track progress. Maybe it is Status = 157!
- We need to ask why the best and only benchmark is 2016, which could be seriously out of date.
- Figure 1.30 : in VR book.

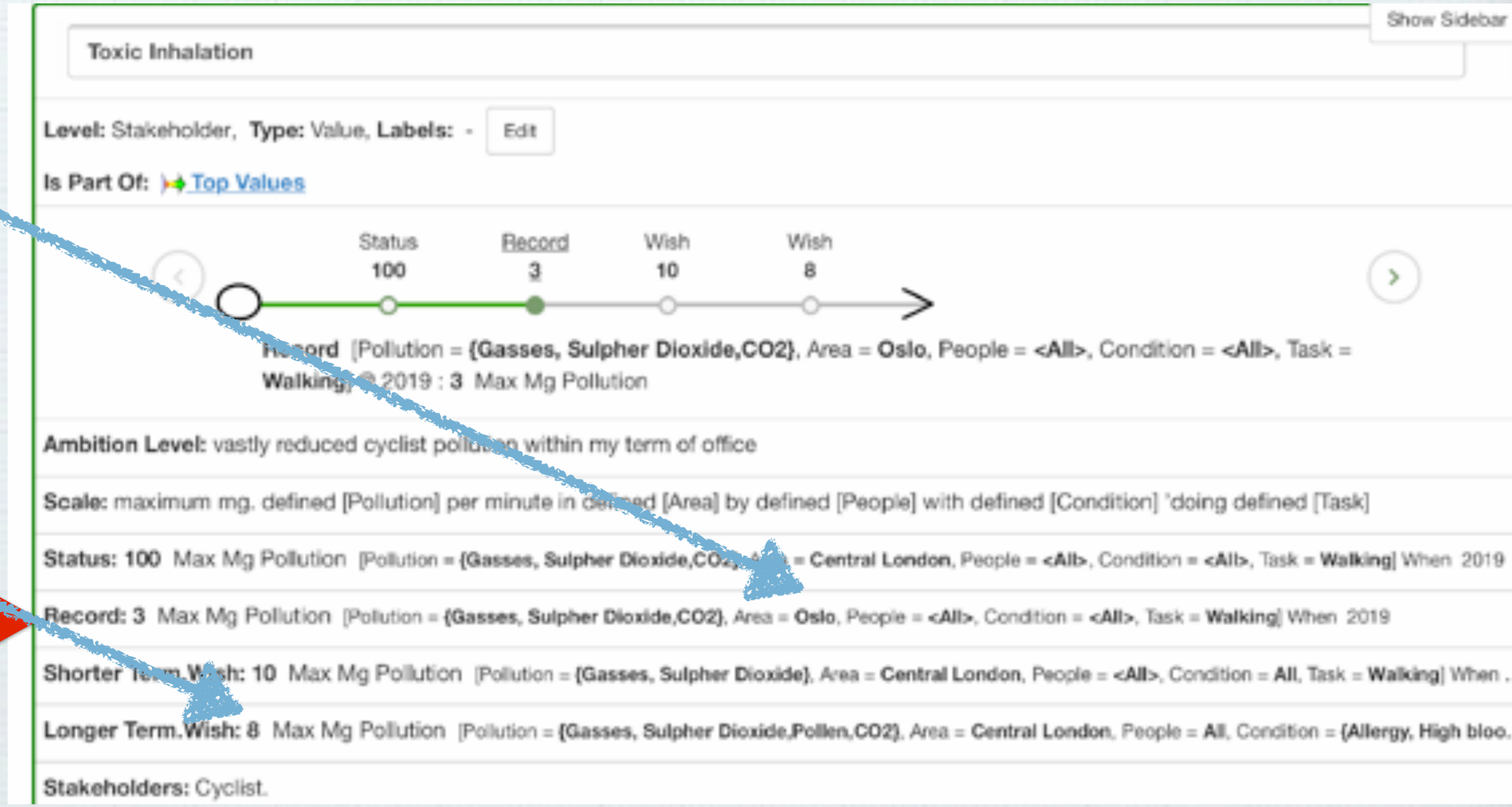
'Status' level: real-time step feedback benchmark (Agile stuff!)

* Here is Status used as an initial planning data,
  * 'where our system is, before we start delivering value increments'.
* It is followed by 2 *different* Wish levels,
  * which have slightly different Scale parameter attributes, and different delivery dates.
* So the Wish levels are not completely comparable to the *Status* information.
* This fact is a signal that Status information might possibly be updated,
  * to be *comparable*, for those Wish conditions, if possible.
* Figure 1.31 : in VR Book.

‘Record’ level (Benchmark type)
The possible state-of-the-art extreme

- We added a Record, level statement
- comparable, but for ‘Area = Oslo’,
  - and it shows that it is possible for a waterside city (Oslo,London) to get to a good low value level of 3.
  - Well Oslo is not London, but what are *they* doing, that *we* might learn from?
- and are our desired Wish value-levels ambitious enough?
  - Are the Wish requirements good enough considering the Record levels?
  - Can we and should we aim to do better?
- Imagine how useful and necessary it is to have comparative benchmarks like this available during presentations and discussions.

- Figure 1.32 : in VR book.



‘Trend-level’ Benchmark
(a Background statement)

- The ‘Trend-level’ benchmark is
  - an attempt to stop looking in the rear-view mirror,
  - and look out at the road traffic, coming up, *ahead* of us.
- This is especially important in environments which experience high rates of unpredictable change,
  - from competitors, enemies, nature, economics, technology, and politics.
  - That is just about all of us
- ‘Trend’ is quite useful to challenge people to think about how things might be
  - with competitors,
  - by the time your plans are delivered,
  - or if you fail to deliver them. (Motivation)

Part 5.
Requirement Level Constraints:
Worst Case, and defining failure clearly in advance.
More potential 'points on our Scale'

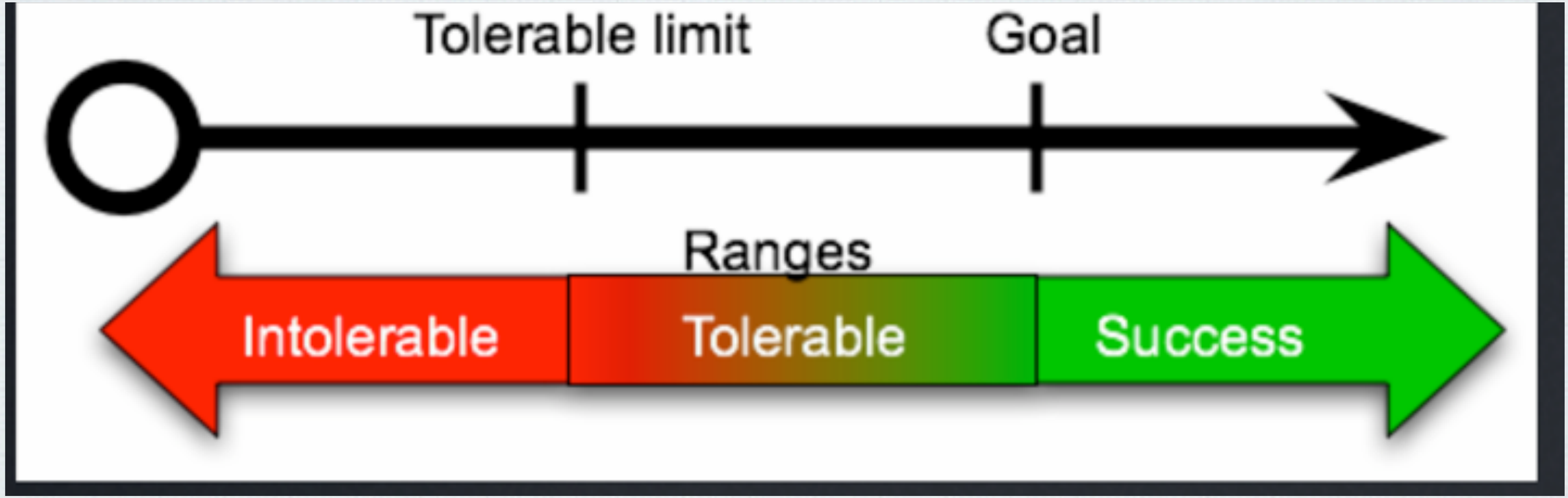


Figure 1.34 : the Tolerable Level, or Tolerable Range is just above the 'Intolerable level, and is a range extending until a 'success level' is defined. It is possible to have Intolerable levels and ranges at both extremes of a value scale, as in too hot and too cold. Constraints are core requirement statements with potentially dramatic effects on priorities and architecture.

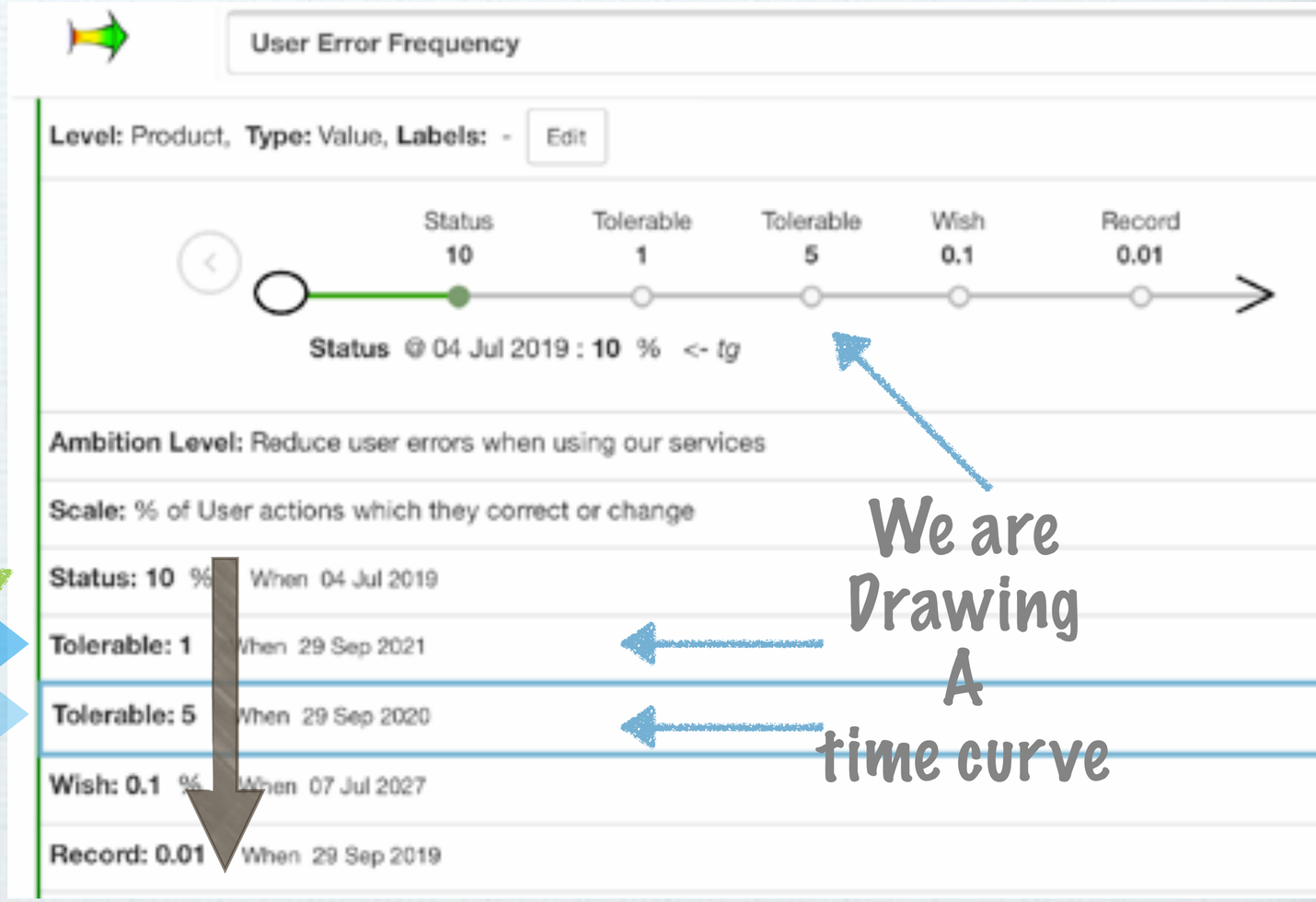
Figure 1.35 : Getting out of Intolerable levels of a critical value, is our first step.

Then incrementing the value until we reach all success target levels (like a Goal).

If any resources remain, we can choose to use them to increment values to more than 'just barely' success. Perhaps towards Stretch levels, or to longer term, and [Special Conditions], success levels.

Multiple Levels enriches planning and control and prioritisation

- Several different statements of different 'Tolerable' levels
  - or any other types of Scale levels
    - like Tolerable, Wish, Past, Trend
  - might be appropriate for different circumstances.
- One single value type, might well need to concurrently specify a variety of different Tolerable levels, for different circumstances.
- This avoids over-generalization of requirements,
  - with consequent unnecessary costs for some circumstances.
  - If you had to generalise, all ends, by using the higher level (worst case) for all sets of conditions
  - This opens for smart *decomposition* of requirements,
    - so that we can support smart value agile delivery sprints
- Because 'multiple level specification' supports our need to focus on particularly critical circumstances early, delivering value to those circumstances early.

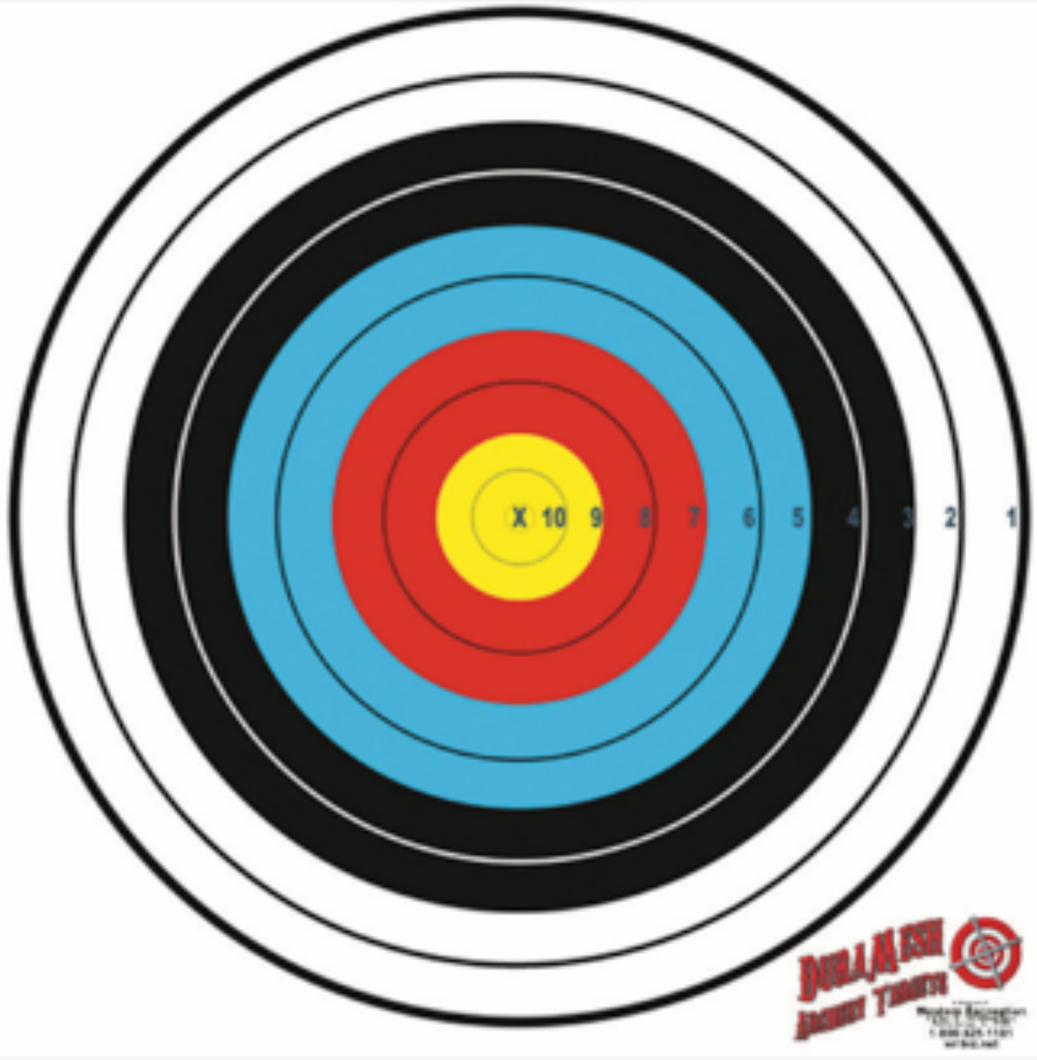


Part 6.

Requirement Level 'Targets':

Defining 'success' in advance, clearly, measurably, contractually
Knowing when to stop using resources.

- A 'target level' is a value level we plan to achieve.
  - 'Wish', 'Goal', 'Stretch'
  - = different priorities of Targets
- There are varying levels of targets.
  - For different deadlines
  - And different [Conditions]
- And there is such a thing as *not hitting the target at all*.



Clear Targets
Begin with clear Scales —->

For example: if the Stakeholder says:

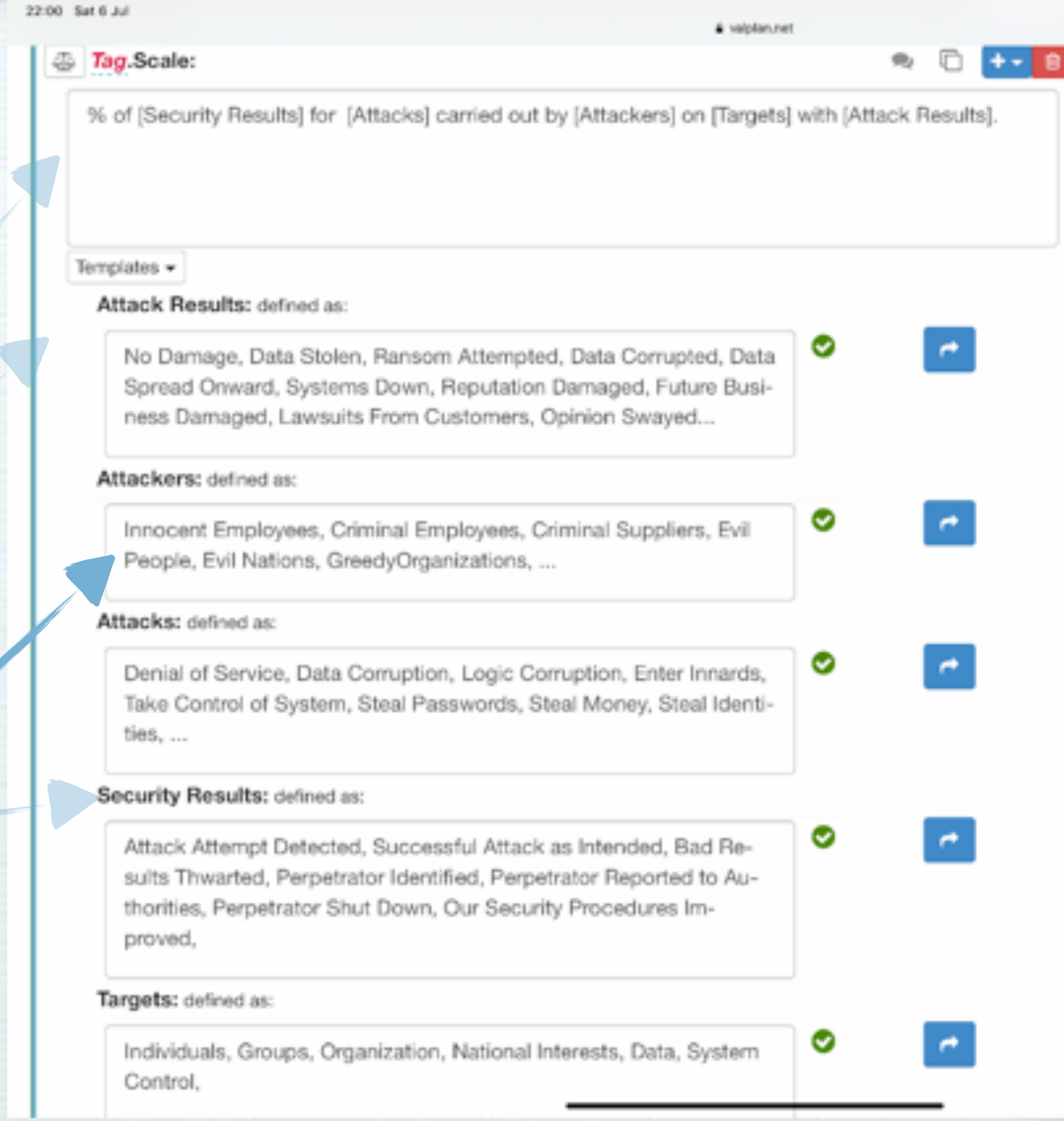
Ambition: ***I want the best security, to fight hackers, and protect my customers and company.***

Or

User Story: ***As a User I want good security, to fight bad guys.***

These are simply *unacceptable* statements

- **No defined scales**
- **No definitions**
- **No conditions [Scale Parameters]**
- **No levels (benchmarks, constraints, targets)**



Wish-level Target

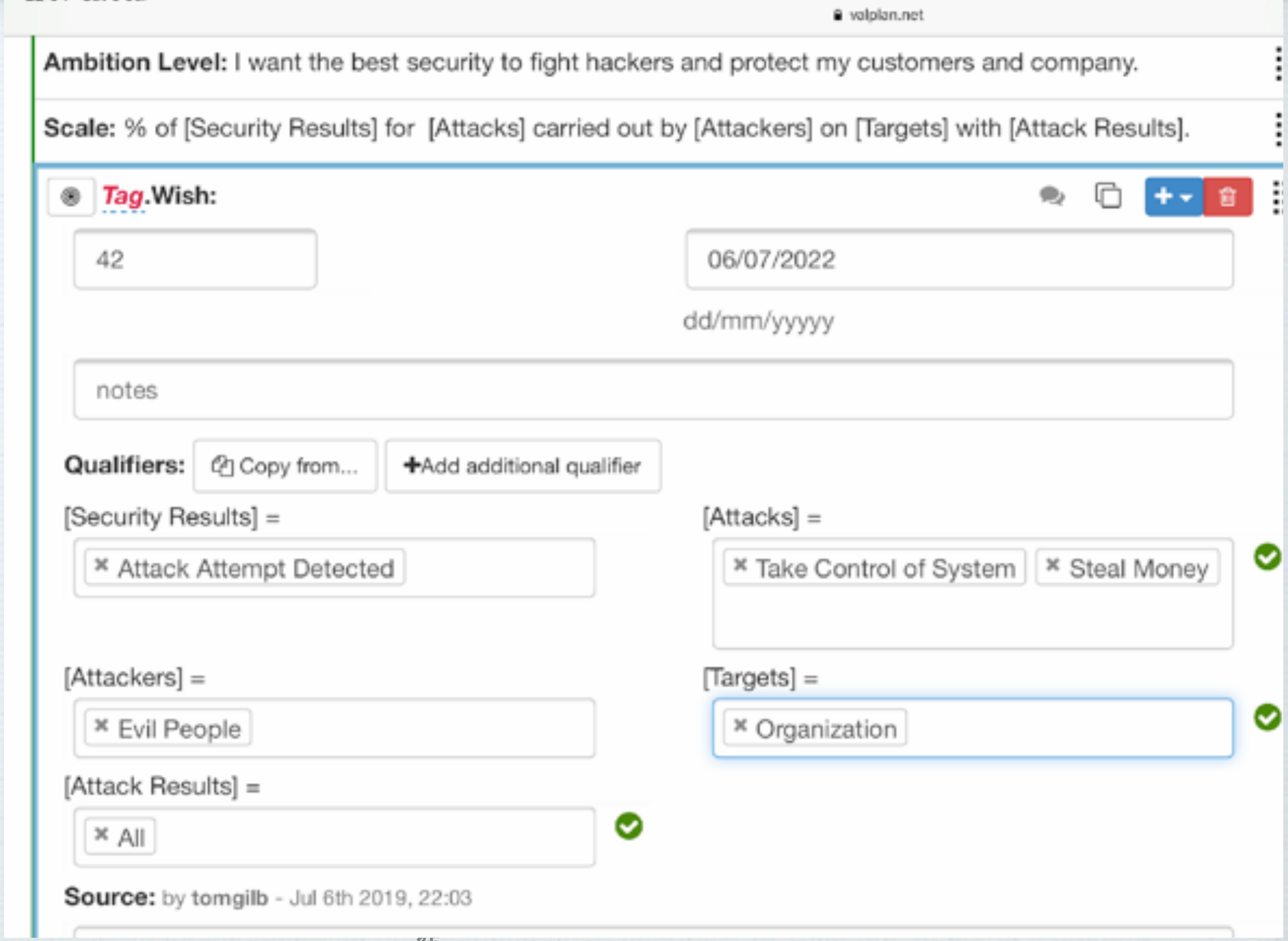
'Wish' level target

A 'Wish' specification is an expression of a *stakeholder desire*, based on *stakeholder's* needs and values.

- **It is a '*stakeholder* target', but not yet qualified as a '*project* target'. (a Goal)**
- **'Wish' belongs to 'systems analysis' (not contracts):**
  - **what should this project *consider* delivering?**
  - **What would be *valued most* by the important stakeholders?**
  - **But, is Wish do-able and cost-effective?**
- **There can be serious problems with Wish statements,**
  - **which means we cannot simply accept them as serious *project* requirements, initially.**
- **'The customer is always right',**
  - **but they might not know state-of-the-art limitations, or have infinite time and money.**
  - **So, before the Wish can be granted we need to pass it through a number of filters (feasibility, costs, priorities)**

Drafting a 'Wish' level

- Note that this Wish statement example window
  - is a very small subset of all the possible Security Scale possibilities.
  - See previous slide, for multiple conditions and possible combinations
- That is good
  - I can focus on this slice of the action,
  - if it is high priority and critical.
- I have a fair chance to understand it,
  - and find security solutions
    - for this level, date and conditions
  - and cost them.
- I do not need to consider all possible conditions, now.
- I can use the Scale Parameters as a practical tool to decompose my 'Security' problem.
- To simplify the real-world complexity, in an agile way
- For prioritised incremental Security value delivery to critical stakeholders



Too early to state a serious 'Goal', a committed requirement

- So now I have my Wish specification.
- But I cannot possibly commit to a Goal
  - (Goal = a 'firm committed promised value delivery from a funded project')
- because I have not identified and costed the necessary technology for delivering the Wish Level
  - (42%) on time, 6 July 2022.
- But at least the 'Wish' problem is much clearer.

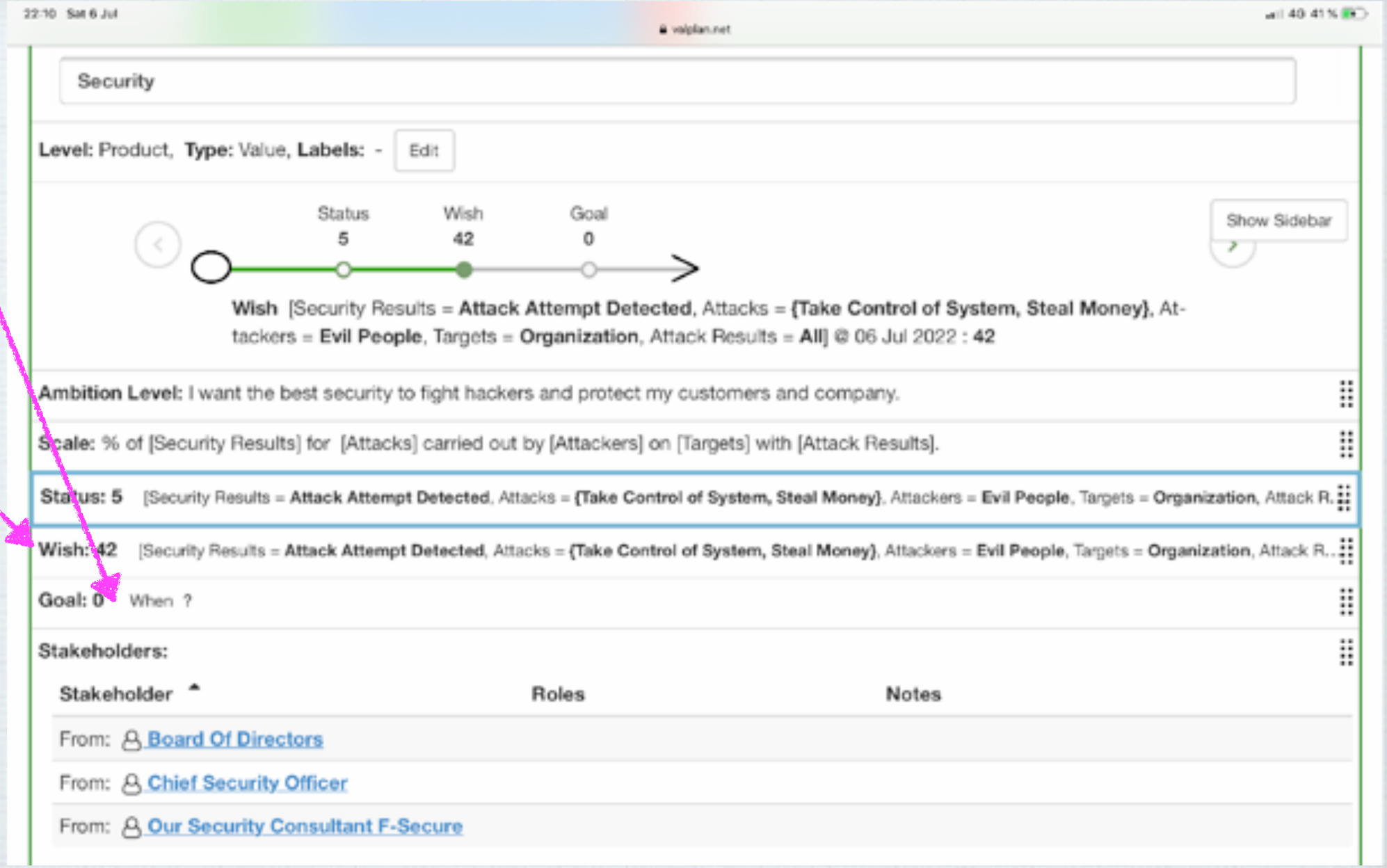


Figure 1.42 : V R Book

Many Goal statements

* 7.3 Comment on the Multiple Goal Example (Fig. 1.43).

* I tagged both the Wish, and a corresponding Goal, with a tag 'A', to make it more visible that they were related.

* And by 'Tagging' any Planguage statement, I can more easily refer to it when presenting, or writing an analysis of the plans.

* Bullet points don't 'make the cut' (work well). Nor do 'yellow stickies' (kid stuff) in my world. Not even for early informal drafting of plans. We live in a complex integrated world and digital plans are the only reasonable tool for modeling it. Even just a Text editor!

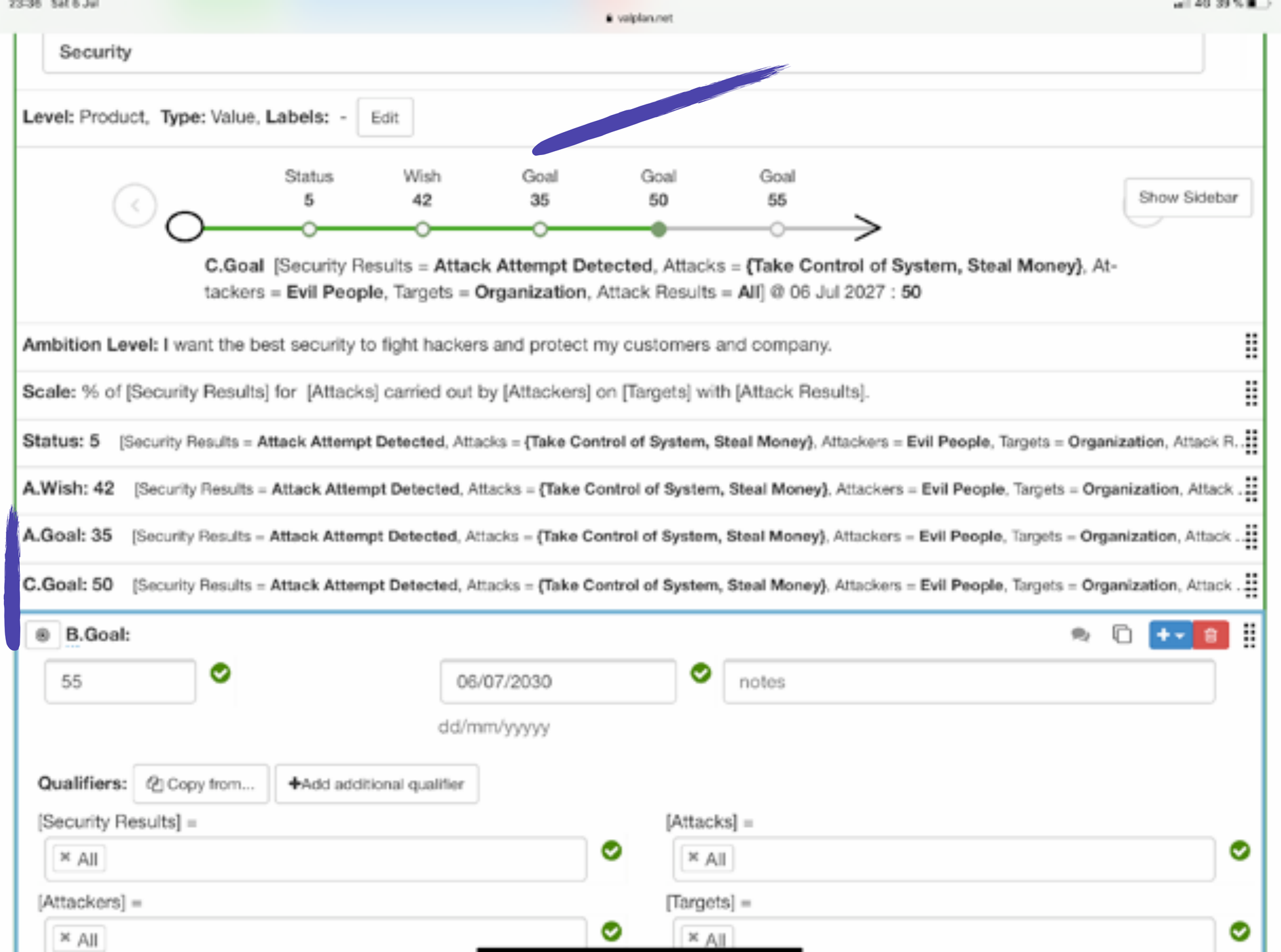
* It looks like we cannot achieve the A.Wish on time, but only a level of 35%. (A.Goal)

* The good news is Goal 'C' (with exactly same Scale parameter attributes as A) we *can* achieve a higher than Wished for level of 55%, later, in 2027

* And Goal B, the ultimate 'All, All, All, All, All attributes' Goal, all security problems which are included in the Scale parameter definitions (see above Scale detail), can be achieved in the year 2030. With current funding! If you want *earlier* results, you can cross my palm with silver.

* The implication for all Goal statements is that we have done the necessary 'homework'. That means we have found a suitable architecture or design, that will give the stated Goal level, by the stated deadline for the particular Goal level.

    * We have in addition found that the 'technology solution costs' are within our 'resource budgets' (see later in the book, Chapter 12), when regard is paid to all *other* Goal commitments in the same time frame, or project. We know that, because otherwise we cannot commit a Goal.



Adding a Stretch level for the 'A' value conditions

7.4 Stretch-level Target

- **A 'Stretch' parameter specification is an *even greater* level-of-success than we dared to commit (in a Goal = commit) to initially.**
- **We are *not sure* if we have the resources, or even technology, to actually get to Stretch levels.**
- **But, Stretch means that some stakeholder, sees *some* specific value, in getting to that level, sooner rather than later.**
- **So 'Stretch' is the 3rd Priority (1 = Survive (Tolerable), 2 = Succeed (Goal), 3 = Surpass (Stretch level).**
- **When all top-level critical factors have reached initial Goals, for this point in time, you have formal permission to choose a Stretch level of that value (or another value), and see if you can reach it, before money, or time, run out.**
- **Stretch can be treated as a sort of 'Stakeholder Wish', that we *cannot* commit to (a 'Goal') in the current time frame. But we might just get lucky! Exceed formal planned expectations.**

Figure 1.45 : VR book

ValPlan | Gilb International AS | Terms Of Use | Privacy Policy | Cookie Policy
Powered by Needs & Means © RSBA Technology Ltd 2015-2019

Part 7
Meters
practical measurement tools to control projects.

Meters are sort of like this Weather Forecasting Stone. — — — —->

They ‘tell it like it is’.

Meters give you the benchmark numbers.

The defined Meter process gives you insight into the *credibility* of the Benchmark

Figure 1.24 :

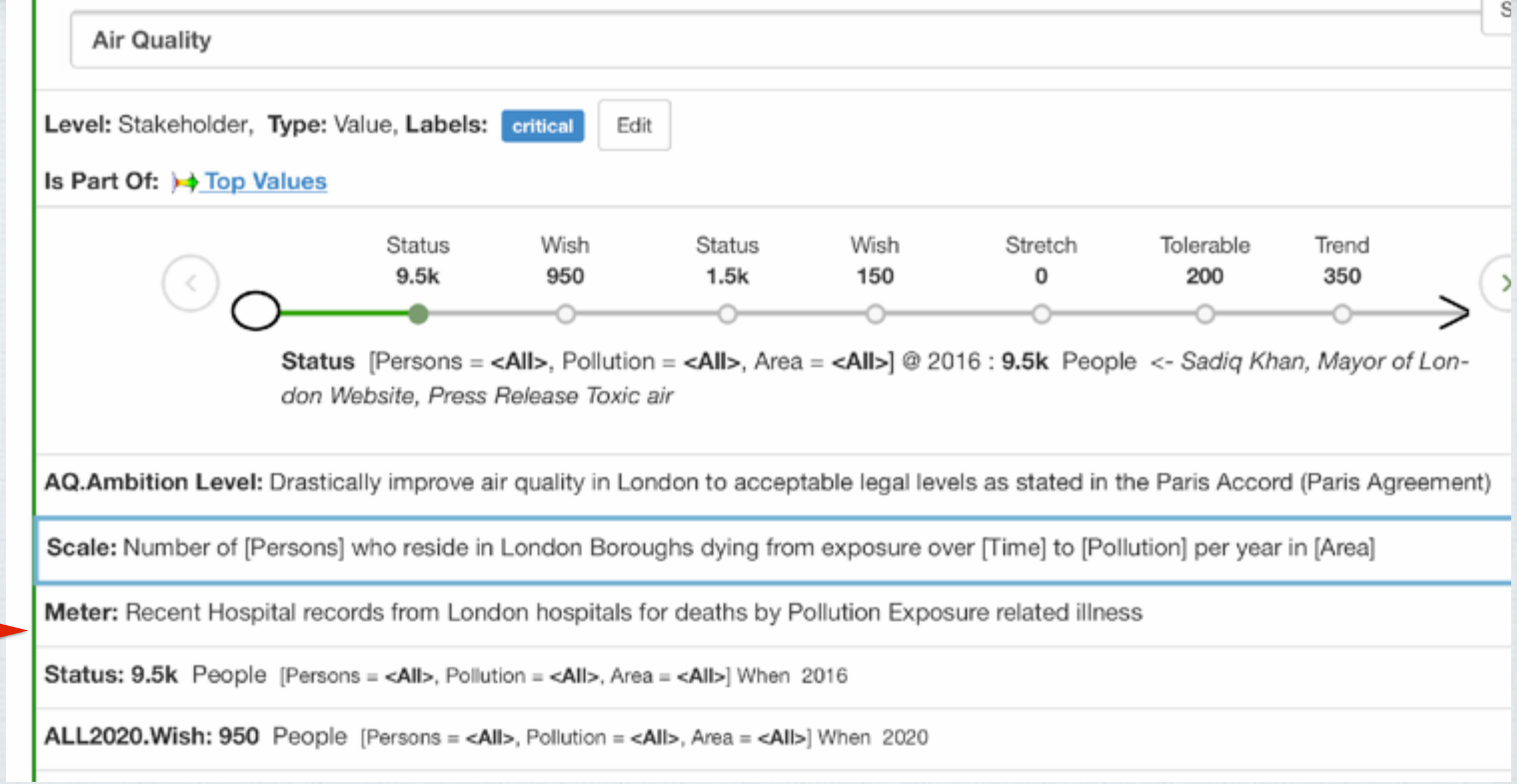
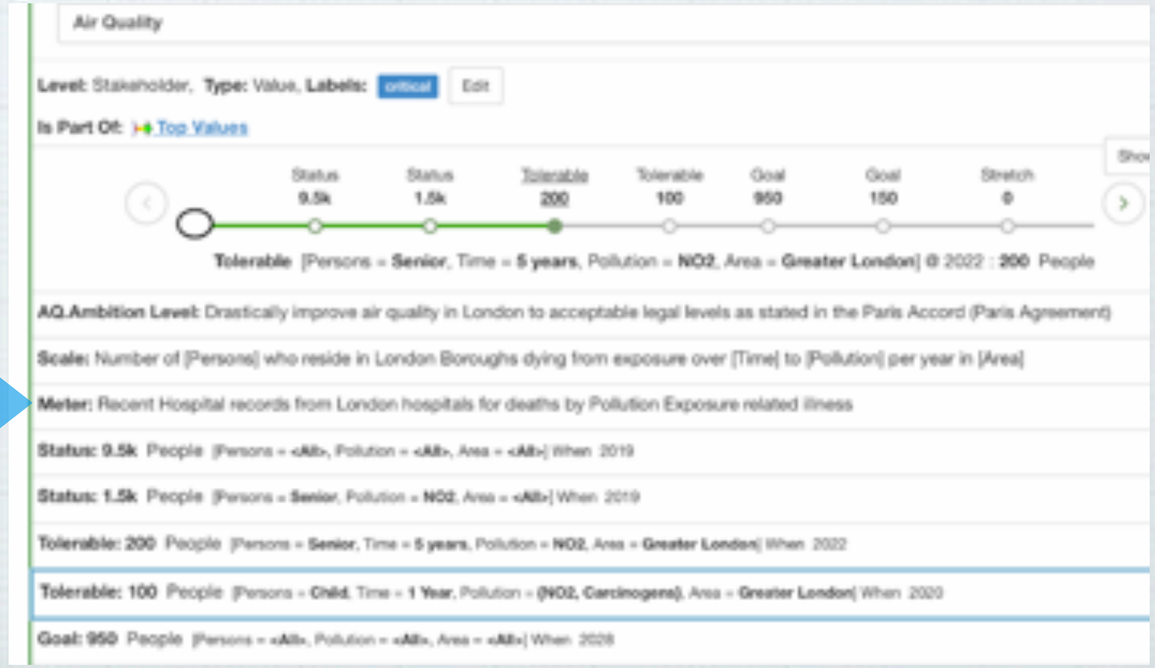


Figure 1.25 : the Meter is a direct reflection of the Scale. It measures along that exact scale.

‘Meter’ spec attributes

* **Here are some of the quality Aspects of a Meter:**
* **Accuracy (is it close enough to the truth?)**
* **Relevance of its Scale**
* **Repeatability (same results each time)**
* **Sensitivity (to disturbing factors)**
* **Credibility (will people believe it and buy in)**
* **Legality (will it break laws, customs, standards, contracts?)**
* **Automate-ability (an we measure automatically?)**
* **Privacy (permission to snoop?). Are we allowed to measure?**

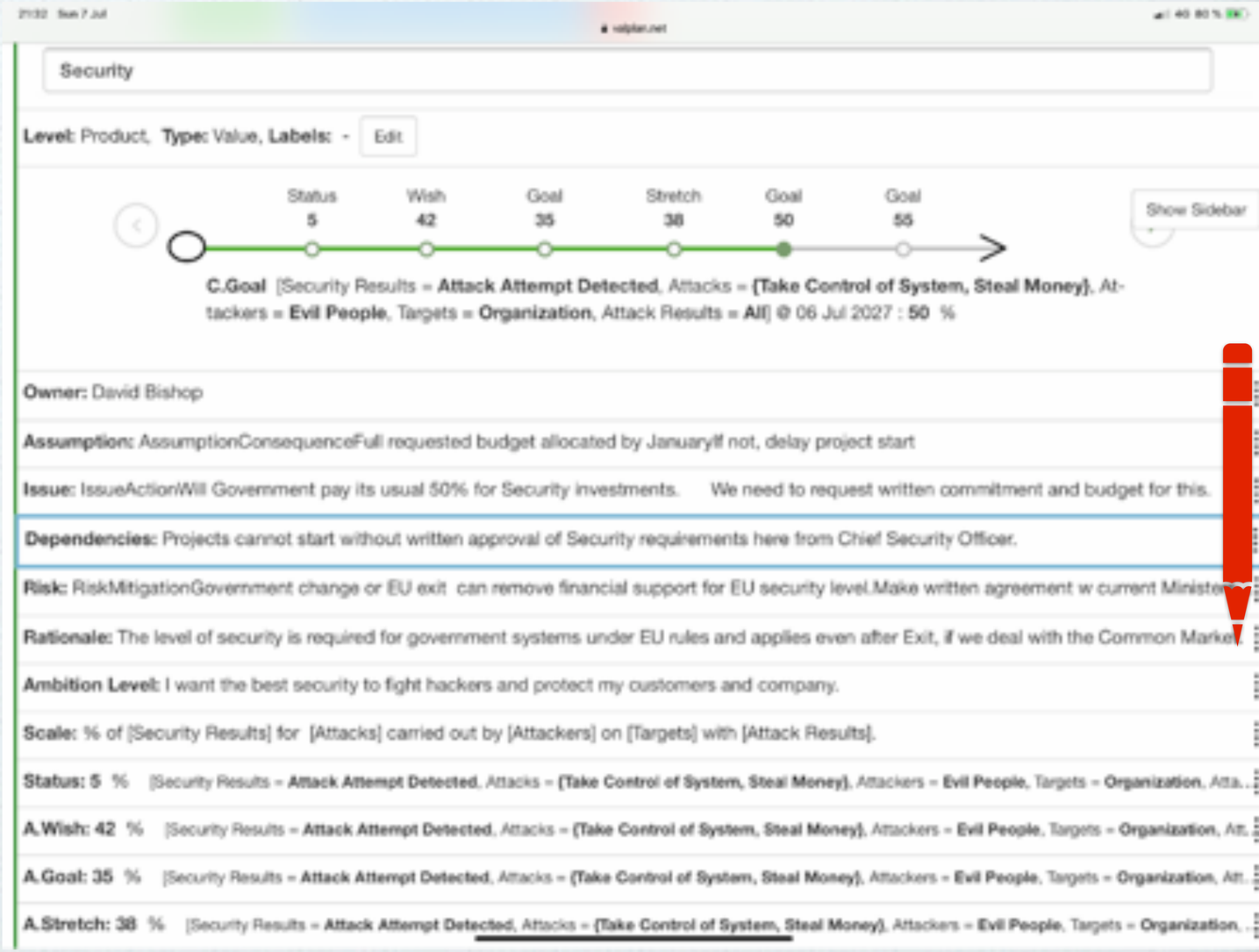
* **Here are some of the cost aspects of a Meter:**
* **Detailed Planning costs**
* **Execution Costs**
* **Result analysis costs**
* **Presentation Costs**
* **Permissions costs**
* **Travel costs**



Part 8.
Background Specification parameters;
for risk management, prioritisation, responsibility.

8.1 The General Purposes of Background Specifications.
The 'Background Spec' purpose is to enrich the requirement spec with information, that:
...

- Might *never* otherwise get specified
- Might be '*lost*' in earlier or later documents, like slide presentations and Business Analysis
- Might *not get used seriously* unless they are 'in your face' in the spec.
- Might be *difficult to retrieve* from other documents, or from human memory
- Might be well-known to some, but *unknown* to others
- Might be correct and updated for some people, but *incorrectly remembered and not updated for others.*
- Is needed for ongoing, real time, incremental steps, of value delivery *decision-making (Agile as it should be)*
- Is needed for *risk management, prioritization, taking responsibility, motivation, reviewing efficiently:* and any other purposes on the path to successful value delivery, without being delayed by poor decisions, based on lack of correct information.
- Is needed to enable *automation* of certain aspects of requirements, such as Quality Control, prioritization, presentation, and risk detection. Yes AI. A complete Specification 'Object'.
- Help to manage the *updating and changes* to the spec.
- Help us to *follow our adopted defined Rules* (specification standards for requirements)



Prioritization and Responsibility: Background statements

* Here are 4 examples of background Parameters to the ‘User Error Frequency’ Value spec. ——————————————->
  * Hopefully you can see that each one might contribute to a _responsibility decision.
* See also these 4 background statements:, ——————————->
  * Stakeholders, Rationale, Cost Impact, and Value Impact.

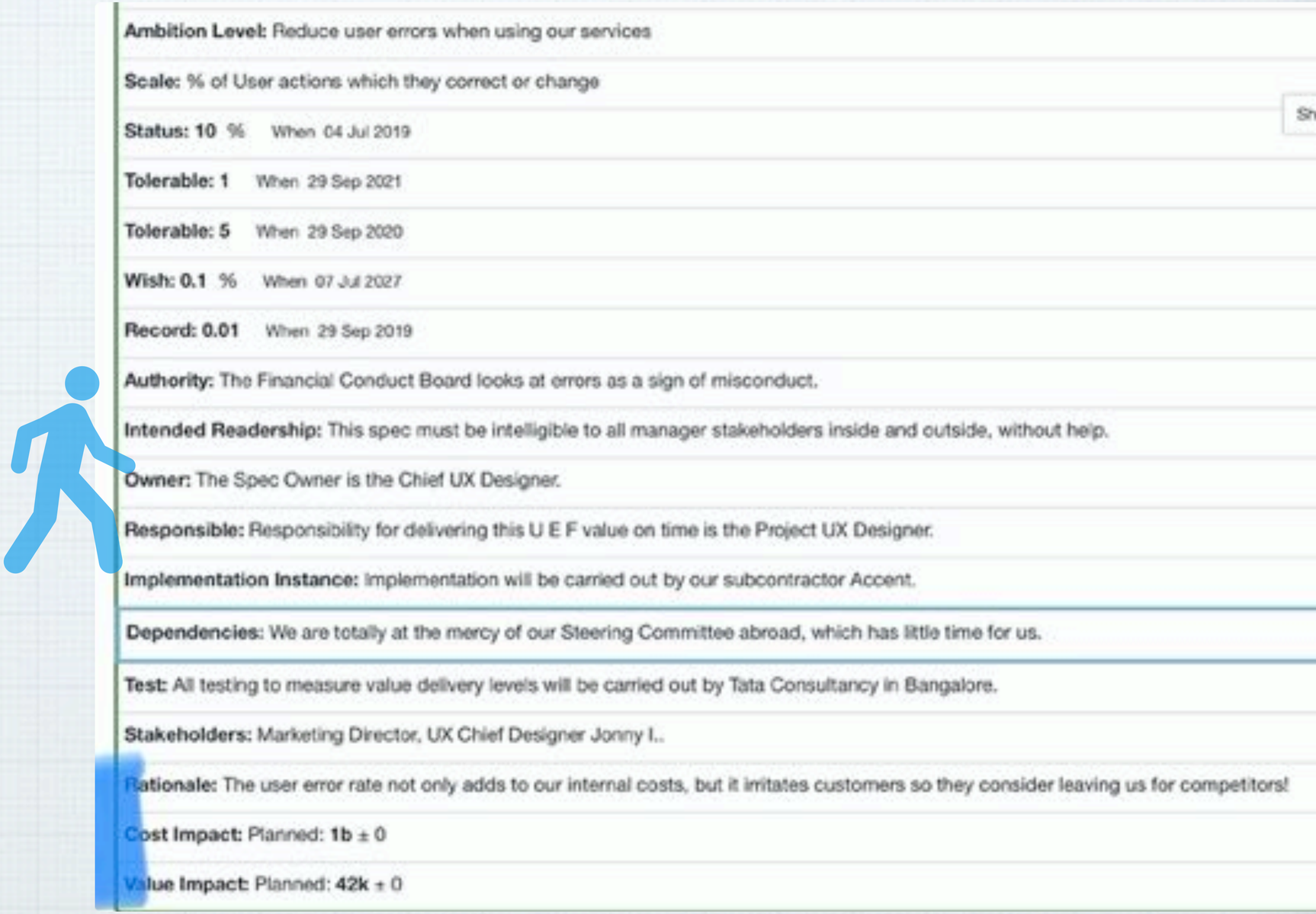


Figure 1.50 in V R Book

Responsibility and Motivation with Background specs

Detailing the Stakeholder Spec.

This is a *detailed window* ——————————->

for a 1-liner Stakeholders spec,

for the ‘User Error Frequency’ value spec,

Above is a ‘diagram presentation’
of the same relationship
Of 2 stakeholders to the ‘User Error Frequency’ value.

Note that the Security Value is related to 3 other stakeholders.

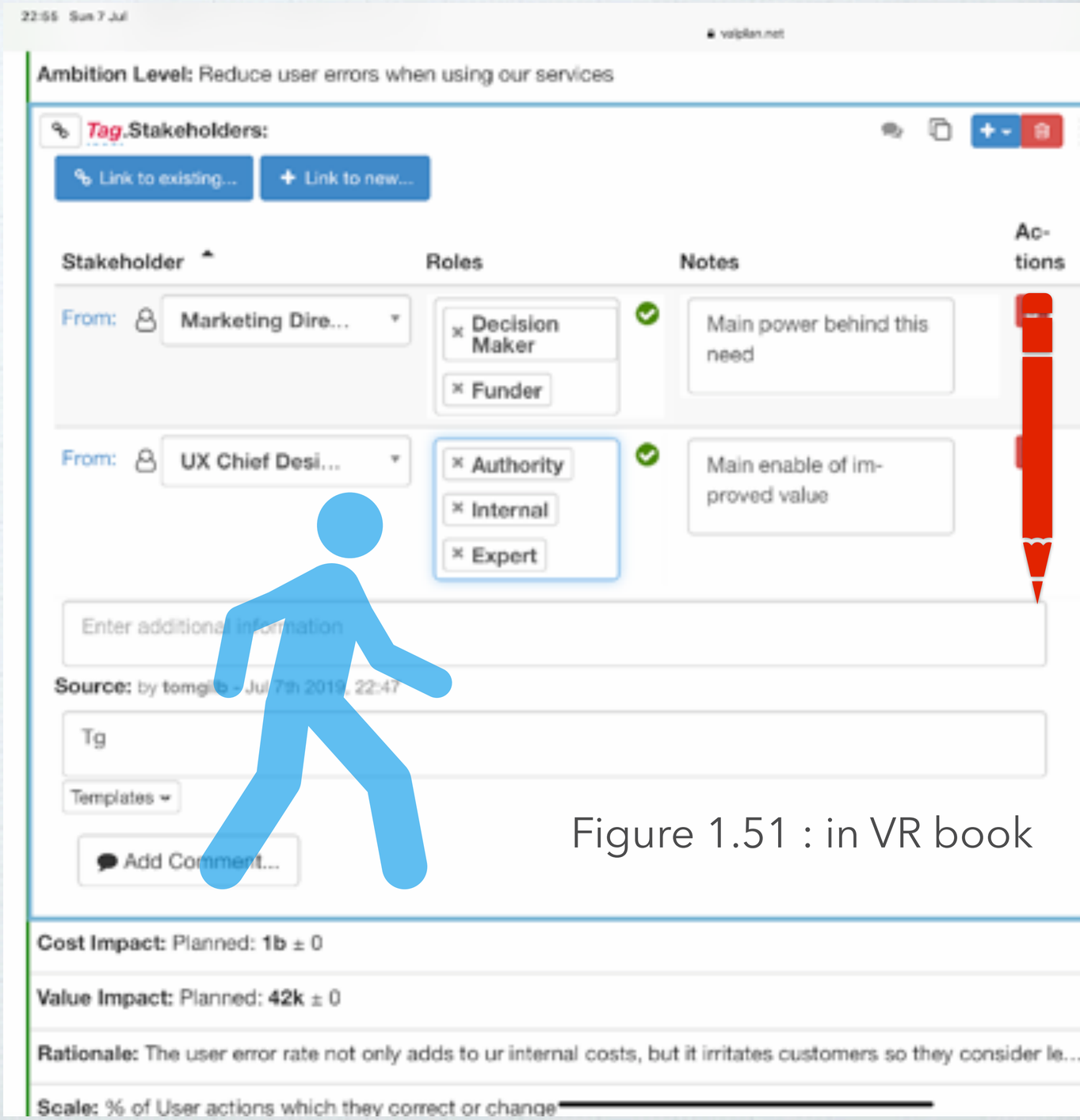


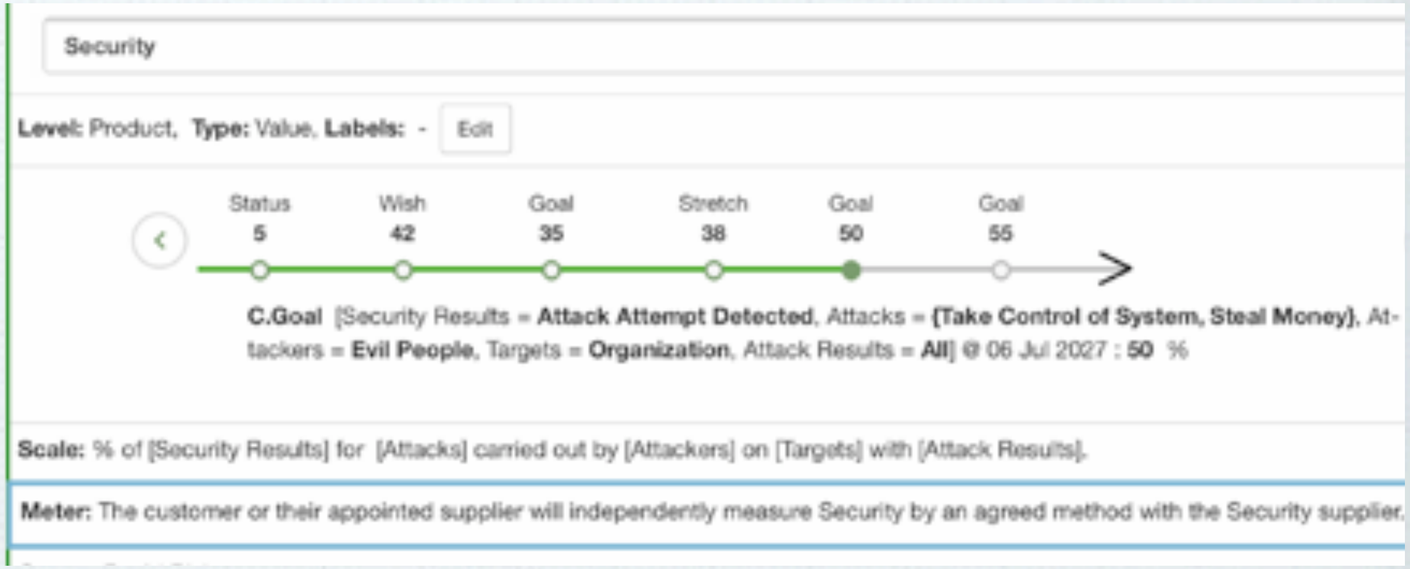
Figure 1.51 : in VR book

Part 9.
Making use of value specifications in practice.

Samples
of a few of the many possibilities

9.4 Contracting and Proposals

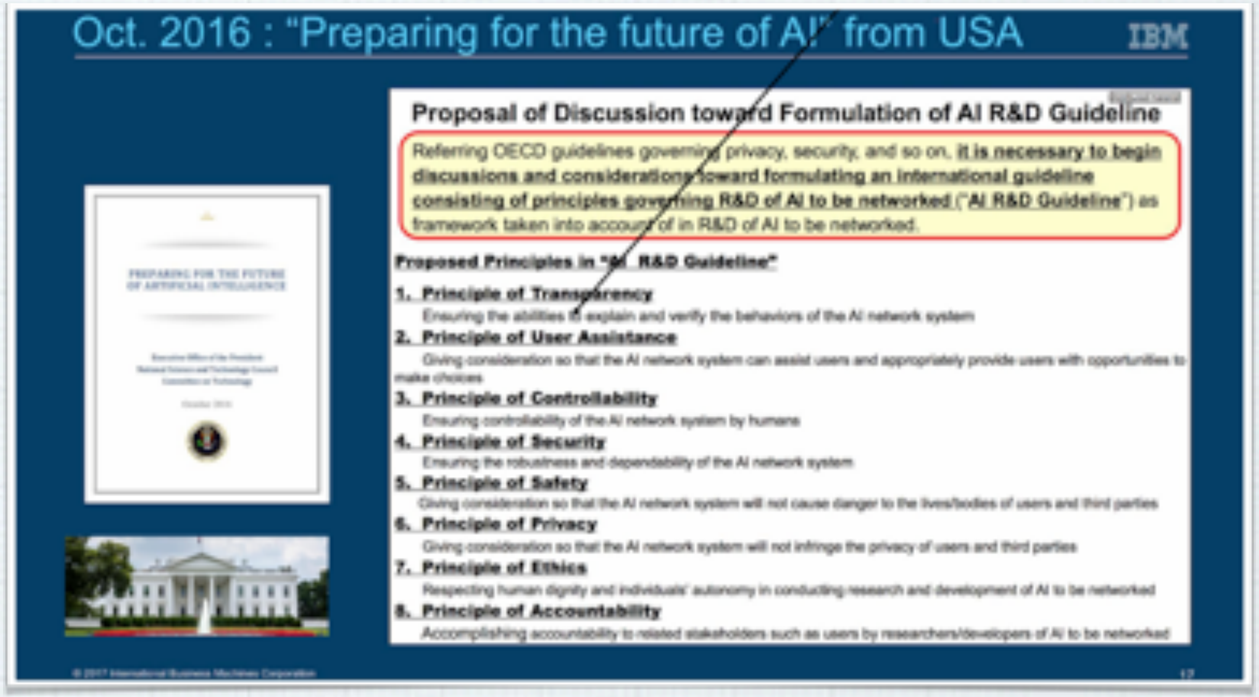
- **A well-specified Value Requirement is absolutely mandatory when soliciting proposals and estimates.**
- **No contractor can possibly estimate the cost of 'high Security'. They need more detail to begin to, even roughly (3X, 10X), estimate costs.**
- **In fact if they make an estimate, and fail to ask you for more information about your exact value specifications, then you are in danger of severe failure.**
- **They are either incompetent, they do not understand the relation between value levels and costs, or**
- **They are *intentionally avoiding this*, to get your business, but plan to charge you much more when you reveal, after their first botched delivery, what your hidden quantified value requirements actually were.**
- **So put that clearly-specified requirement in your proposal. In their face, up front. And make sure they understand that this is a legal condition for their estimates, for acceptable delivery, and for payment.**



9.5 Make AI and 'Sustainability' Clearer

9.5 Handover to architecture and design engineering.

- **Requirements, as you know well, are the basic input, the problem formulation, for any process of design, or architecture, or design engineering.**
- **Unfortunately there are several 'disciplines' that do not have 'clear quantified value requirements' as an input.**
- **These informal design disciplines I encounter in management planning (only time and money thinking), IT (we code, we found bugs, and have deadlines, not value). And in IT Enterprise Architecture.**
- **More worryingly, I found a complete absence of quantification for the most important Artificial Intelligence International Standards, led by top professionals and academics (c, XAI).**
- **And in the UN Sustainability Development Goals**
  - https://tinyurl.com/UNGoalsGilb
  - **BCS course 27 May 2020, SG SPA**
- **The good news is some of my friends there decided to listen!**



SERIOUS VALUES CANNOT BE BS****

Quantifying AI Transparency, and UN Sustainability:

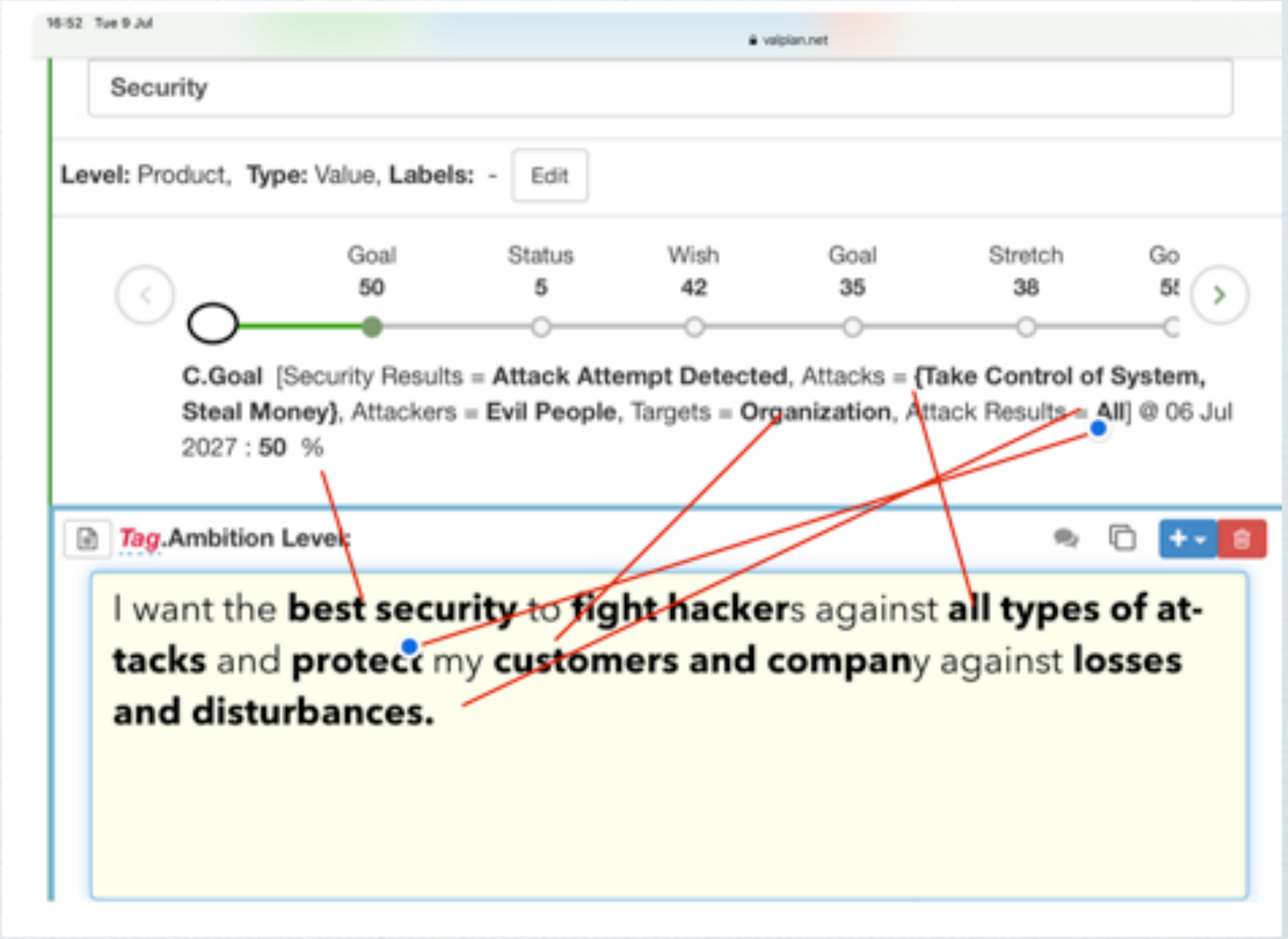
http://concepts.gilb.com/dl962
Aim2North Conf. 7Nov2019, Slides
Podcast video 24 minutes before the lecture
https://www.youtube.com/watch?v=J70zf1gF2b8
Doomsday version
http://concepts.gilb.com/dl964

10. Presentation of Value Specifications

Samples
of a few of the many possibilities

10.1 Presentation to Stakeholders

- Stakeholders need to feel they are listened to, and understood correctly.
- They need to feel that we are really correctly analyzing the most critical sets of Scale Parameter attributes (who, where, when, what):
  - the value sub-sets they need action on, in the short term.
- One presentation method is
  - to start with the Ambition Level, and its source or authority.
  - Then show that the Scale is helping to clarify



10.5 Value Presentation to Architecture

- Let us define the Systems Architect as whoever makes the technology selection,
  - all the stuff needed to deliver the Values, within Resource Budgets. And other constraints.
- Here is the brief to them: (See 'Value Design', 2019, gilb.com)
  - BCS SPA Course 29 April 2020
- find a set of designs (the 'architecture') which arguably will deliver the planned Value Levels, within the resources budgeted.
- document your ideas with estimates and evidence, on an Impact Estimation Table (CE, VP) also called Value Decision Table.

Part 11.
Levels of Value Specifications

11.1 Vision Levels, Vision Engineering

Visions are the territory of top politicians and business , or society leaders. Think Musk, Kennedy, Martin Luther King.

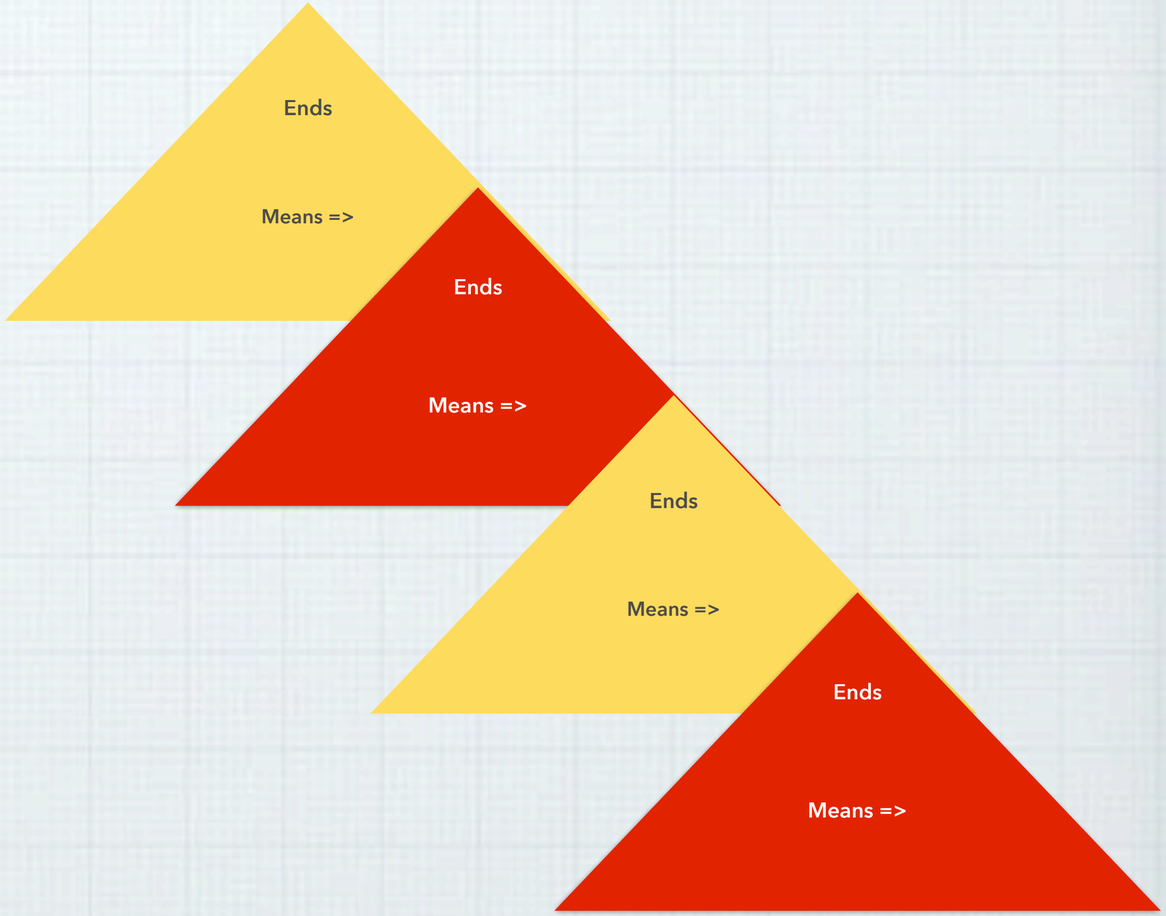
We can certainly categorize *visions* as the top level of Value requirements. Everything which contributes to vision achievement is *related* to them.

You could, at one extreme, say that everything *below* that vision level is some kind of 'design' (or architecture, or solution) for achieving that vision.

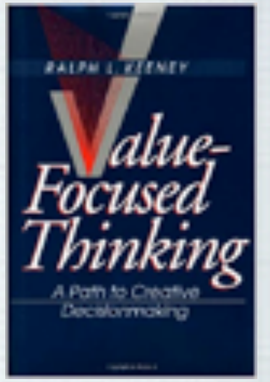
At least that is true for the Visionary, they have value requirements, and everything that helps to achieve their vision, is clearly a *solution to meeting those requirements.*

However, just below the visionary is a hierarchy of vision-production stakeholders. Each one of them has a set of their own level of requirements, their own *value* requirements.

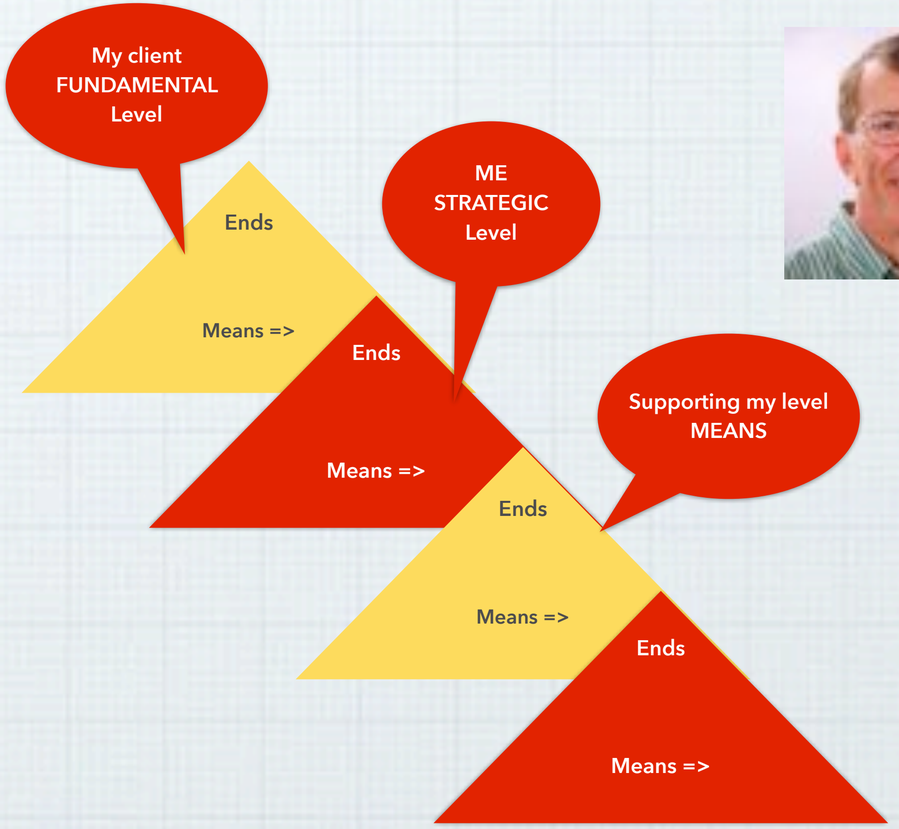
Each one of them has their own set of solutions to reach their requirements. And some of those solutions, are in fact perceived as 'requirements' by the next level below them, in the value chain.



11.2 Fundamental Value Requirements



* Figure 1.74 : Ralph Keeney's Levels Hierarchy.
* Upstream from you, are your Fundamental-level demands.
  * They become your Strategic requirements,
  * your own level of requirements.
* There is the real danger that the Fundamental Means, your requirements, are badly thought out, badly formulated.
  * What should you do?
  * Blind obedience, or constructive change, with your help?
* Downstream of your responsibility is the 'Means Objective' of someone.
  * Your helper's task.
  * If they reach it, that means your solution (their means objective) is 'in the box'.
  * Your value requirement levels will be met.



Keeney, Ralph L. 1992. Value-focused Thinking

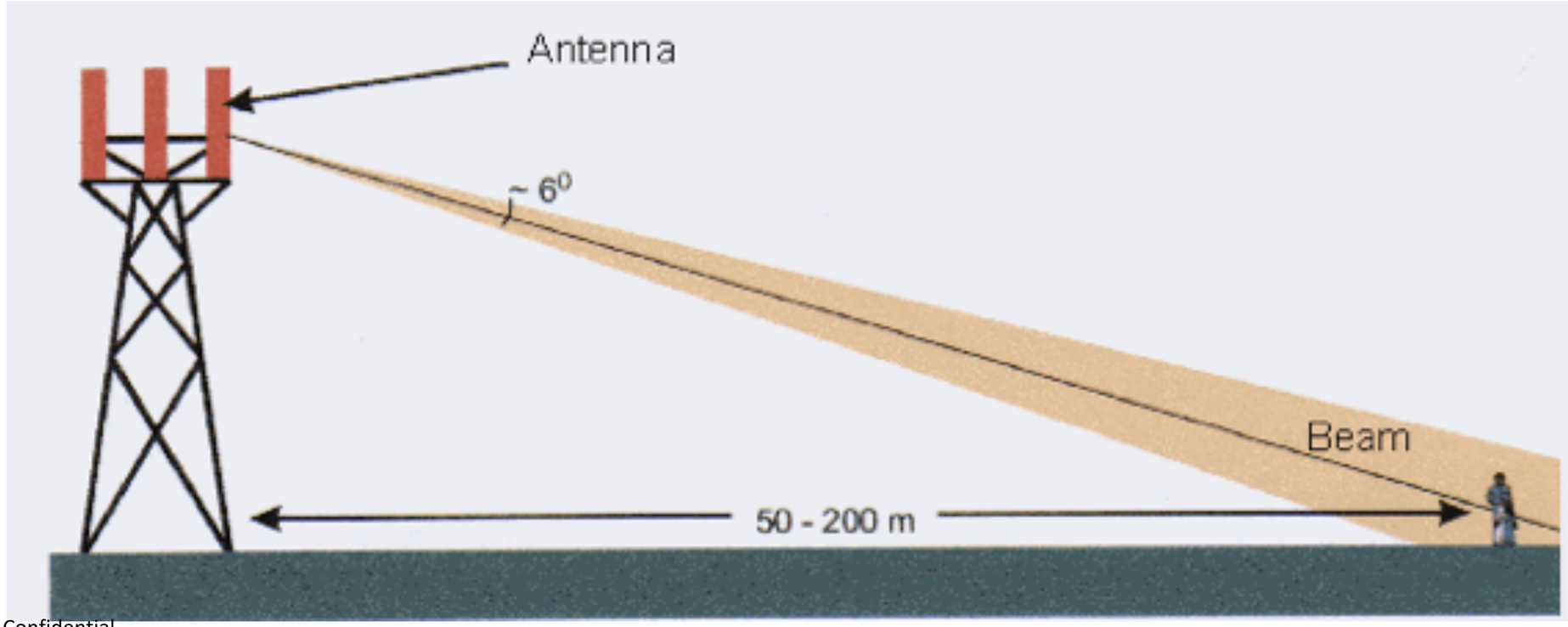
Case Study Ericsson Productivity

Software Engineering Productivity Study

ERICSSON

An example of setting objectives for process improvement

For 1997 with 70% software labor development content in products



Main beam from a macrocell base station antenna

Non-Confidential

The problem

- Great Market Growth Opportunities
- Too Few Software Engineers

- Solution:
  - Increase productivity of existing engineers

The One Page Top Management Summary

(after 2 weeks planning)

The Dominant Goal

Improve Software Productivity in R PROJECT by 2X by year 2000

Dominant (META) Strategies

Continual Improvement (PDSA Cycles)

.DPP: Defect Prevention Process

.EVO: Evolutionary Project Management

Long Term Goal [1997-2000+]

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 Ericsson Quality Policy:

ERICSSON

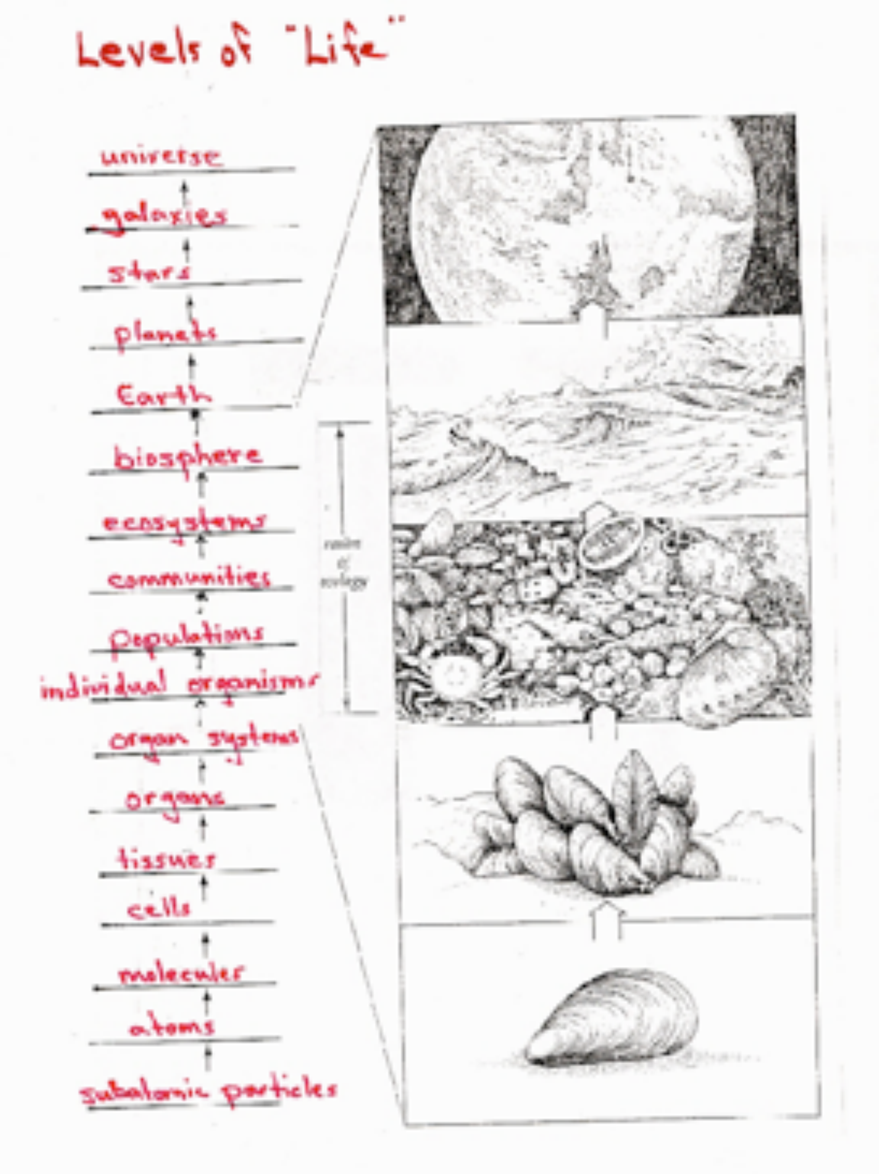
"every company shall *define* **performance** indicators (which) ..

- –reflect **customer satisfaction**,
- – internal **efficiency**
- –and business **results**.

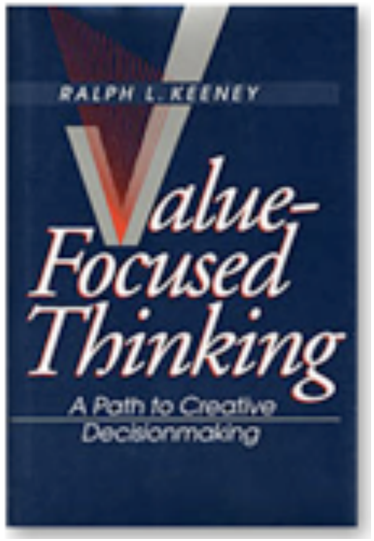
- The performance indicators are used in **controlling** the operation."
- Quality Policy [4.1.3]

Levels of Objectives.

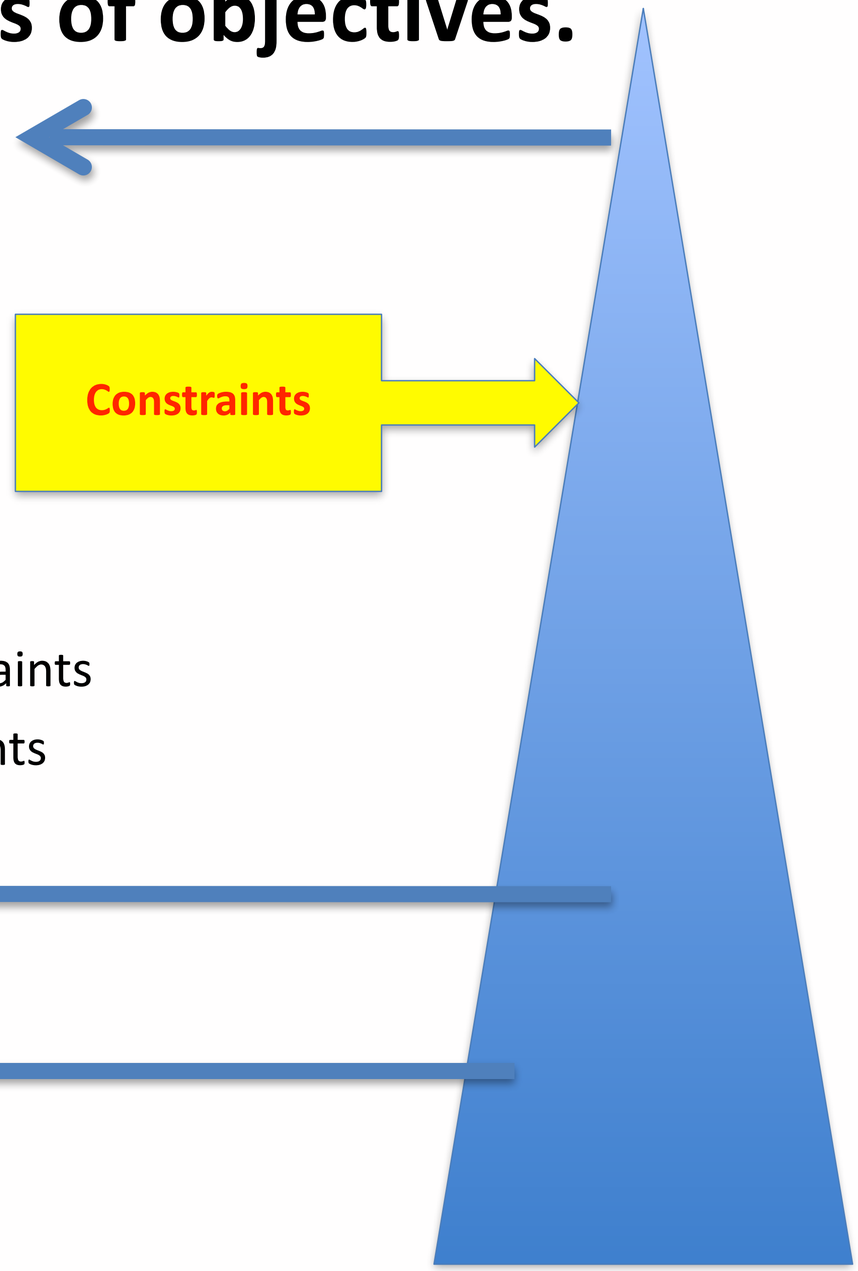
- **Fundamental Objectives**
- **Strategic Objectives**
- **Means Objectives:**
-
- **Organizational Activity Areas.**
  - Pre-study.
  - Feasibility Study.
  - Execution.
  - Conclusion.
- **Generic Constraints**
  - Political Practical
  - Design Strategy Formulation Constraints
  - Quality of Organization Constraints
  - Cost/Time/Resource Constraints



Keeney's: Levels of objectives.



- **1. Fundamental Objectives**
  - **(above us)**
- **2. Generic Constraints**
  - **(our given framework)**
  - Political Practical
  - Design Strategy Formulation Constraints
  - Quality of Organization Constraints
  - Cost/Time/Resource Constraints
- **3. Strategic Objectives**
  - **(objectives at our level)**
- **4. Means Objectives:**
  - (*supporting* our objectives)



The Strategic Objectives (CTO level)

- 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**:

‘Means’ Objectives:

- Support the **Strategic** Objectives
  - ***Complaints:***
  - ***Feature Production:***
  - ***Rework Costs:***
  - ***Installation Ability:***
  - ***Service Costs:***
  - ***Training Costs:***
  - ***Specification Defectiveness:***
  - ***Specification Quality:***
  - ***Improvement ROI:***

"Let no man turn aside,
ever so slightly,
from the broad path of honour,
on the plausible pretence
that he is justified by the goodness
of his end.
All good ends can be worked out
by good means."
Charles Dickens

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?

Principles for Prioritizing Strategies

- They are well-defined
  - Not vague

- The have some relevant predictable numeric experience
  - On main effects
  - Side effects
  - Costs
  - Risks - Uncertainty
- Not huge spread of experience

"Software Productivity" =

Lines of Code Generation Ability

–*"Software Engineering net production in relation to corresponding costs."*

–*Ambition: Net lines of code successfully produced per total working hours needed to produce them. A measure of the*

– *efficiency ('effective production/cost of production') of the organization in using its software staff.*

Scale: [Defined Volume, kNCSS or kPlex] per Software Development Work-Hour.

•**Scale: [Defined Volume, kNCSS or kPlex] per**

•**Software Development**: Defined:

•Productivity calculations include Work-Hours

• **Meter : <PQT Database and EPOS, CPAC>**

–***Comment: we know that real software producti… as it is available in our current culture. AB, PK, TG.***

–**P1: Past [1997, ERA/AR] < to be calculated wh…**

• **Past-R PROJECT: Past [1997, R PROJECT] < to be calculate… available, available Volume/Work Hours >**

• **Past-EEI: Past [1997, Ireland, Plex] ___??__ kPLEX / Work-Hour.**

•**<add more like LuleÂ>**

•**Fail [end 1998, R PROJECT, Same Reliability] 1.5 x Past-R PROJECT <- R PROJECT AS 3 c " by 50%".**

–***"50% better useful code productivity in 1.5 years overall"***

•**Same Reliability: State: The Software Fault Density is not worse than with comparable productivity. Use official The Company Software Fault Density measures <- 1997 R PROJECT Balanced Scorecard (PA3).**

•**Goal [Year=2000, R PROJECT, Same Reliability] 2 x Past-R PROJECT,**

– **[Year=2005, RPL, Same Reliability] 10?? x Past-R PROJECT**

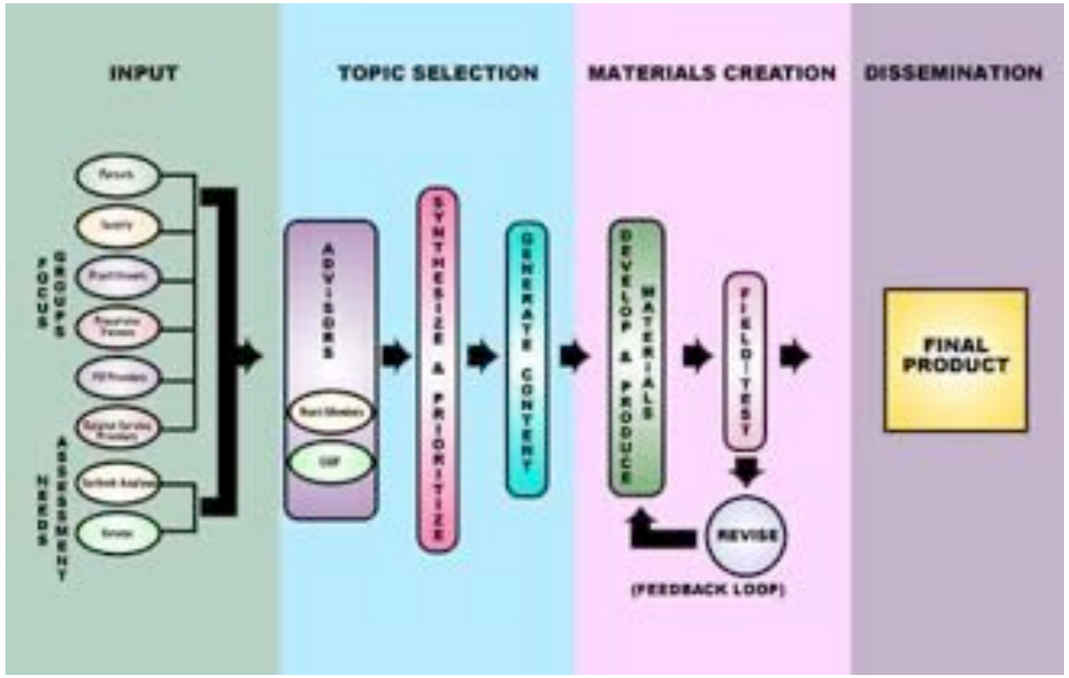
•**Wish [Long term, vs. D pack.] 10 x Past-R PROJECT "times higher productivity" <- R PROJECT 96 1.1 c**

•**Wish [undefined time frame] 1.5 x Past-R PROJECT <- R PROJECT AS 3 c " by 50%"**

–***Comment: May 13 1997 1600, We have worked a lot on the Software Productivity objectives (all day) and are happy that it is in pretty good shape. But we recognize that it needs more exposure to other people.***

Lead-Time:

- **Lead-Time:**
  - *"Months for major Packages"*
- ***Ambition**: decrease months duration between major Base Station package release.*
- **Scale: Months from TG0, to successful first use for**
  - **major work station package.**
  - ***Note: let us make a better definition. TG***
- **Past [C Package, 1996?] 20? Months?? <- guess tg**
- **Goal [D-package] 18 months <- guess tg**
- **Goal [E-package and later] 10.8 Months <- R PROJECT 96 1.1 a "40% > D"**
- **Goal [Generally] ??? <- R PROJECT AS 3a**
  - ***"10% Lead-Time reduction compared to any benchmark".***

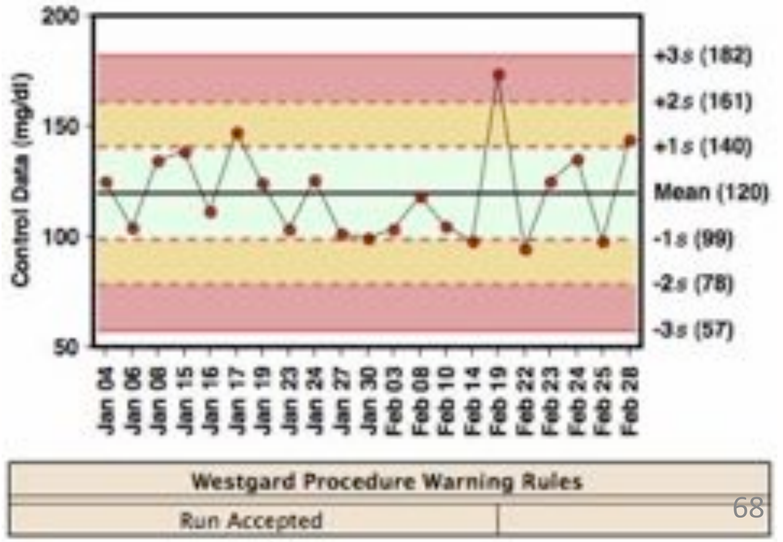


Predictability of Time To Market:

- **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**

Product Attributes:

- **Product Attributes**:
  - *"Keeping Product Promises."*
  - *Ambition: Ability to meet or beat agreed targets, both cost, time and quality. (except TTMP itself, see above)*
- ***Scale*: % +/- deviation from [defined agreed attributes with projects].**
- ***Past* [1990 to 1997, OUR DIVISION] at least 100% ???**
  - **<- Guess. Not all clearly defined and differences not**
    - **tracked. TSG**
- *Goal* [Year=2000, R PROJECT] **near 0%** negative deviation <- TsG for discussion.



Customer Satisfaction

Customer Satisfaction:

"Customer Opinion of Us"

Scale: average survey result on scale of 1 to 6 (best)

Meter: The Company Customer Satisfaction Survey

Past [1997] **4**

Goal [1998-9?] **5** <- R

PROJECT 96 1.1 b

Profitability

- **Profitability:**
  - ***"Return on Investment."***
  - ***<u>Ambition</u>: Degree of saleable product ready for installation.***
  - **<u>Scale</u>: Money Value of Gross Income derived by**
    - **[All R PROJECT Production OR**
    - **defined products] for**
    - **[Product Lifetime OR**
    - **a defined time period]**
  - **Goal: <we did not complete this>**

‘Means Objectives’ Samples
Same *definition* process as higher level objectives

Means Objectives

- *“support Strategic Objectives”*

- Summary:
  - **'Means Objectives'** are
    - not our major Strategic Objectives (above),
    - but each one represents areas which if improved
      - will normally help us achieve our Strategic Objectives.
  - Means Objectives have a lower priority than Strategic Objectives.
  - They must never be ‘worked towards’
    - to the point where they reduce our ability to meet Strategic Objectives.

Complaints

Complaints:

"Customer complaint rate to us"

***Ambition**:*

Means Goal: for Customer Satisfaction (Strategic).

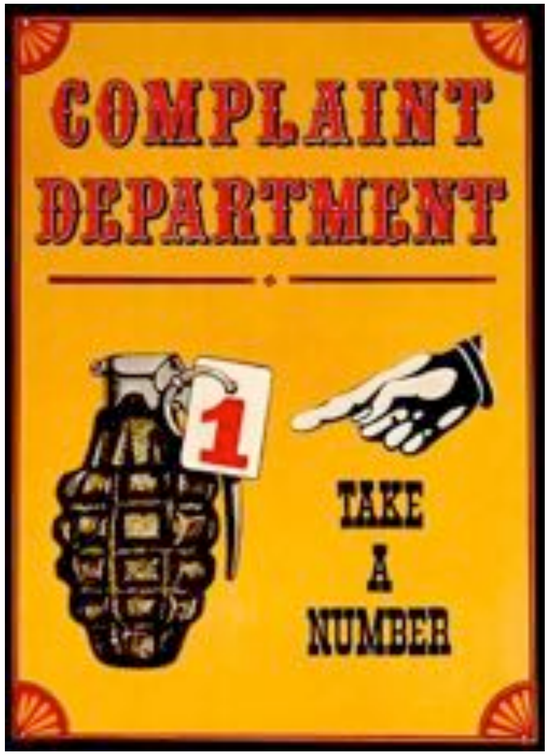
Scale: number of complaints per customer in [defined time into <operation>]

Past [Syracuse Project , 1997] ?? <bad> <- ML

Goal [Long term, software component, in first 6 months in Operation] **zero complaints** <- R PROJECT 96 1.1 b

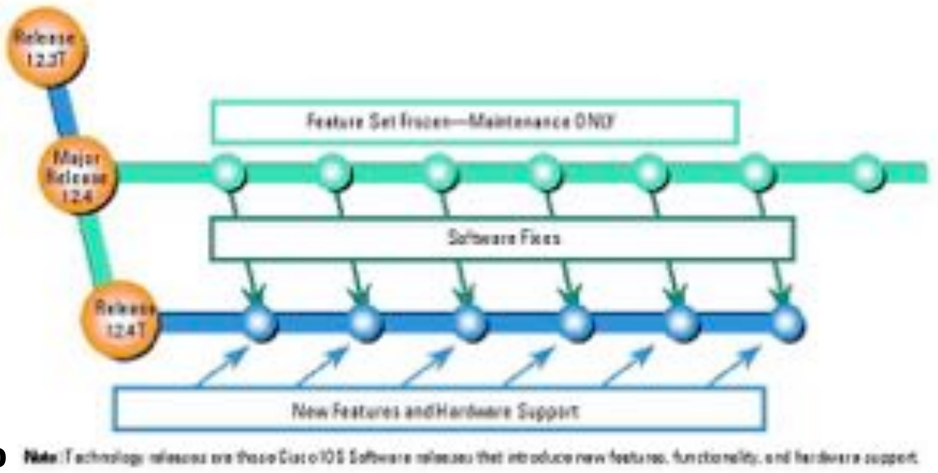
"zero complaints on software features"

Impacts: <one or more strategic objectives>



Feature Production:

- **Feature Production:**
  - *"ability to deliver new features to customers"*
  - ***Ambition**: reverse our decreasing ability to deliver new features <- R PROJECT AS 1.1*
  - **Scale**: Number of new prioritized <Features> delivered successfully to customer per year per software development engineer.
  - Too Little: **Past** [1997] ?? "estimate needed, maybe even definition of feature"
  - **Goal** [1998-onwards] **Too Little + 30% annually**?? <-For discussion purposes TsG.
  - "we need to drastically change our ability to effectively develop SW" <- R PROJECT AS 1.1



Improvement ROI:

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

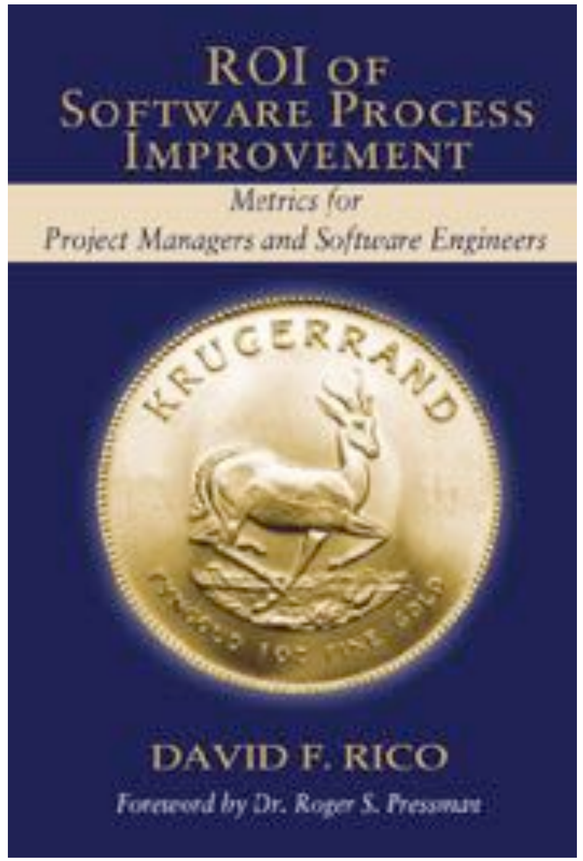
_____[Shell NL, Texas Instruments , Inspections] 30:1 <- Independently published papers TsG

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.



2004

Part 12.
Defining Resource Levels, short term and long term.

Resource Budgets

Here are some interesting resource categories:

Up Front Costs: *before* producing Value

- Capital Costs Money
- Calendar Time
- Work Hours

Lifecycle Costs

- Money for recurrent lifetime operational costs
  - System Maintenance, bug fix, port to new platf
  - Software Licenses
  - Premises and Hardware rental
  - Training Costs
  - Internet Costs

12.6 The possibility of getting control of the value-to-cost ratio by decomposition; and then by prioritization of high-efficiency designs.

- By decomposing values, functions, and design components into a more-detailed set-of-things,
  - we can select a small fraction of the total product or system, to implement incrementally.
- This will give us real 'value delivery' to *some* stakeholders, for *some* value improvements.
  - But, at least as important, it will give us *measurable feedback* about a large number of factors,
  - so we can better see what we need to improve,
  - before we scale up, when implementing the rest of our agenda.

Value Decomposition Methods	Details		
Design Decomposition	Independently Implementable	Measurable Value Delivery ability	Urgent subset
Value Decomposition	Multiple Values	Scale Dimensions	Scale Dimension Sets
Deadline Decomposition	Date Due Start Date	Event Synchronization like 'Law in Force', 'Official Holiday'	Condition Synchronization like 'Interest Rates over 5%','Heat Wave'
Priority Type Decomposition	Wish (Desired Level)	Tolerable (Minimum Level)	Goal (Committed Level)

23. DIFFERENT TYPES OF DECOMPOSITION INTO COMPONENTS OR SUB-SETS

Source: Value Planning book, Chapter 5 Decomposition

Figure 1.82 : various decomposition methods,

so that we can learn reality early, and credibly,

before scaling-up size or volume.

https://tinyurl.com/VPDecomposition
(for Value Planning detailed chapter)

Real Example of Decomposition of Technical Solutions to Control Cost-Efectivenss

Figure 1.83 : real planning example of decomposition and values/ costs estimation, of the decomposed design ideas.

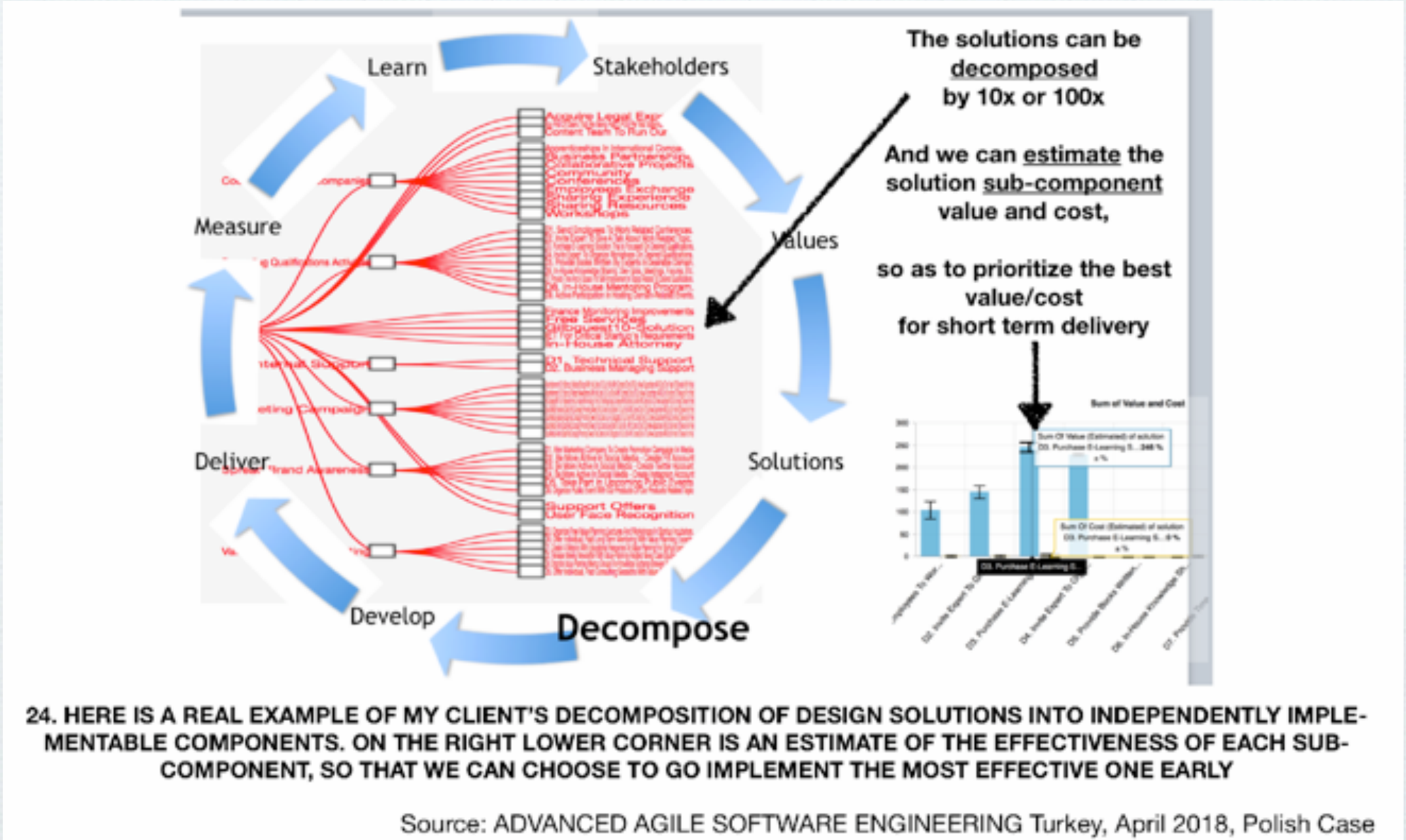


Figure 1.83 : real planning example of decomposition and values/costs estimation, of the decomposed design ideas.

Part 13.
Change Control of Value Specifications

VR 13.1 The concept of a specification Owner. For Specification Change Control

A ‘Specification Owner’, or more precisely a Specification Object Owner is a person or group given sole power to change a specification object, such as a single Value Requirement.

The spec Owner should:

- have accepted the Owner role voluntarily
- Be more than usually knowledgeable in the specific requirement
- Be interested in making the spec the best possible, over time; motivated.

The spec Owner is responsible for:

- receiving any hints from any sources, like stakeholders, of the need for corrections, updates, and changes
- Being password-enabled to actually do, and publish, any change
- Informing all instances, documented in the specification object, all relevant stakeholders, of the pending change, and the actual change (according to corporate guidelines for changes)
- Quality controlling, and reviewing changes, personally, or using others, and using Rules for specification best practices.

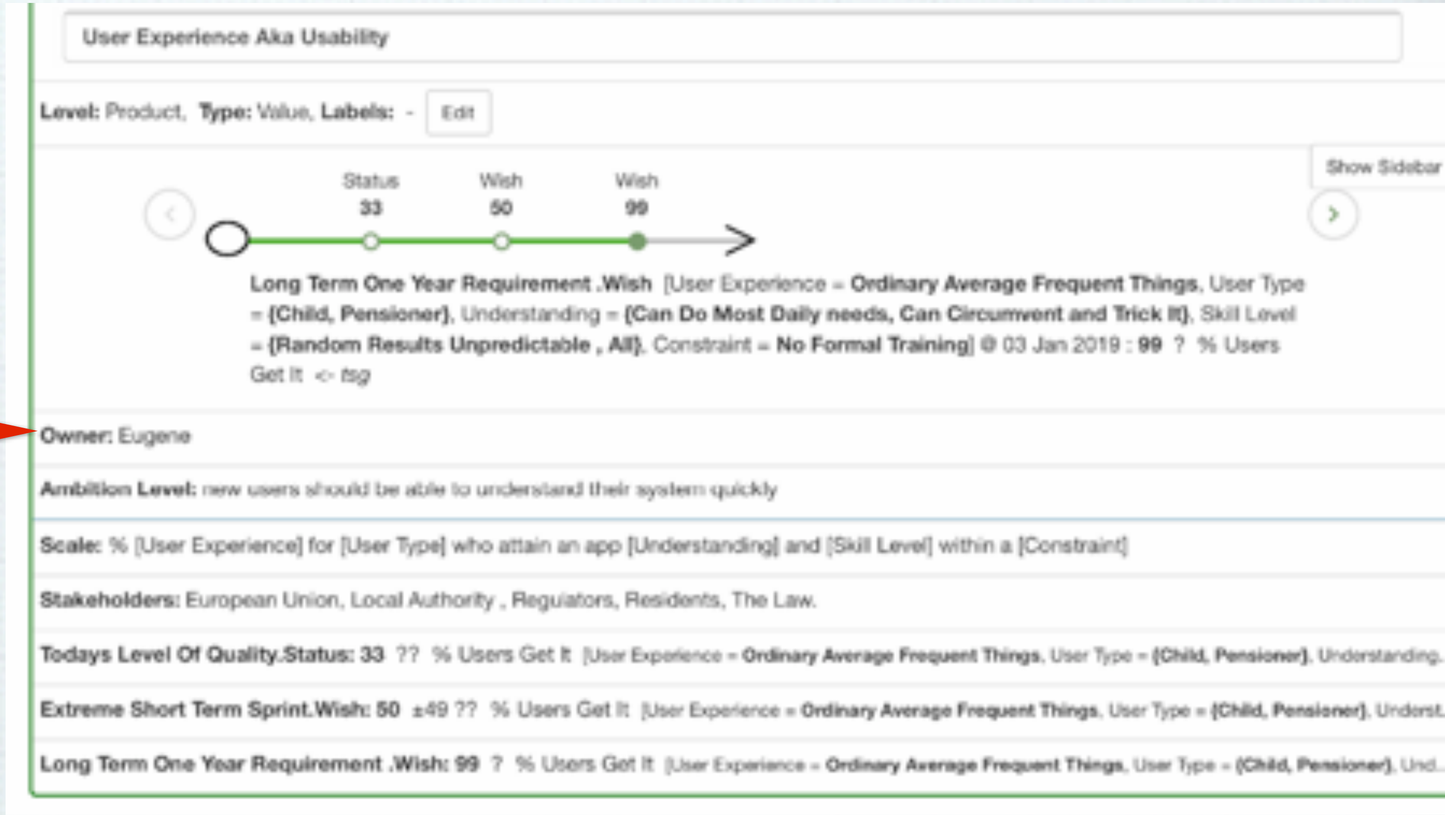


Figure 1.87 : ‘Eugene’ is the designated spec object (the User Experience Aka Usability Value requirement) Owner. ‘Source BCS April 2018, Waste Management’

VR 13.2 Annotation of change source, and time stamps

- I am pretty clear that we need to annotate the 'Source' of each individual element of a plan.
- At the same time it is a good idea to get a time stamp for exactly when changes are made.
- There are two change sources:
  - the Spec Owner, or whoever actually keys in the change
  - The information Source: 'who exactly said 64%?', or 'London?'
- For example:
- Wish: 99% <- Tom
- And then there is the question of exactly WHY a change was made,
  - its justification, or background.
  - This justification is important because:
  - We need to make sure the change is really justified.
  - We need to explain to other stakeholders why the change is being made.
  - Other stakeholders need to be able to argue about that justification.

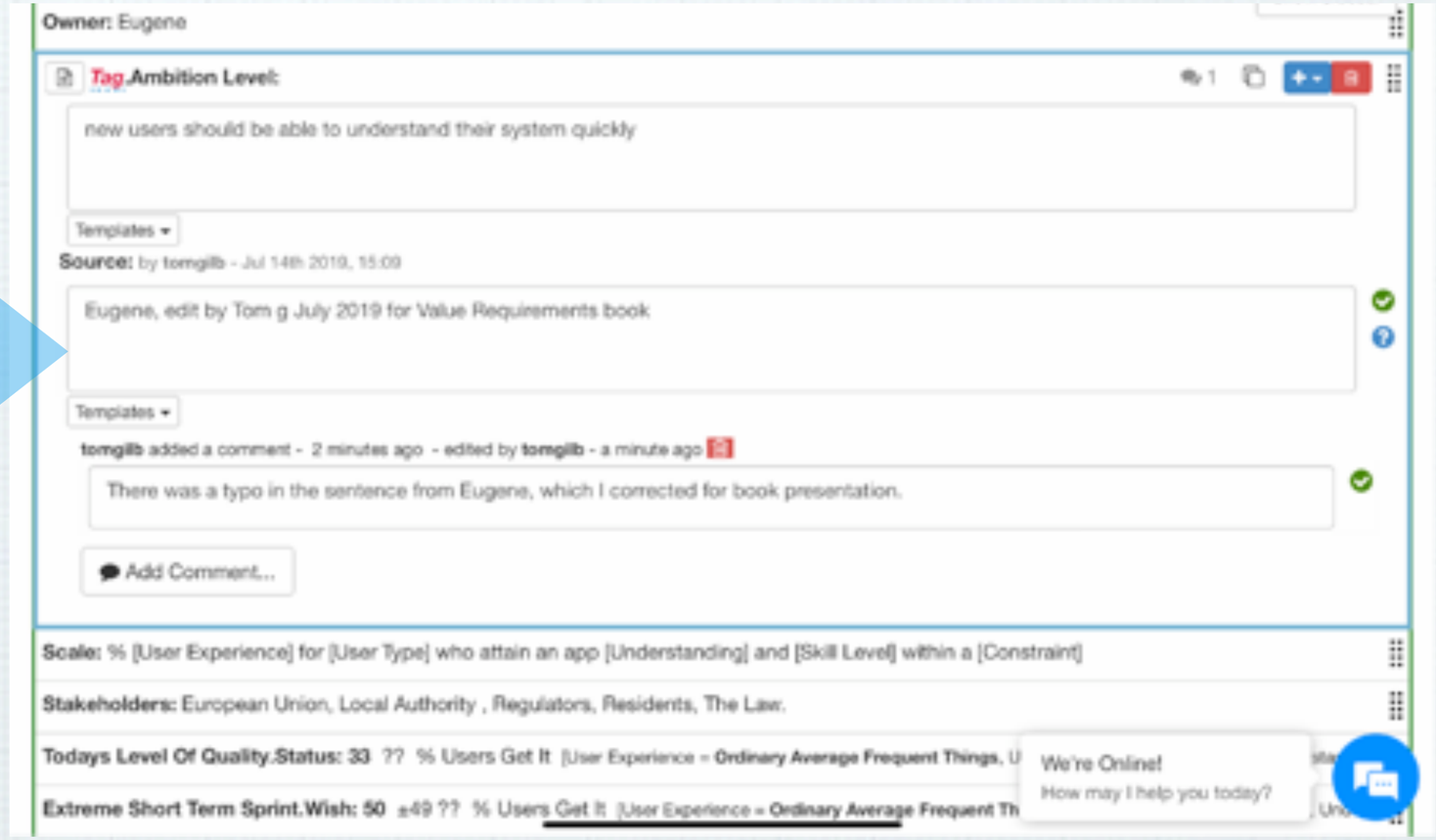
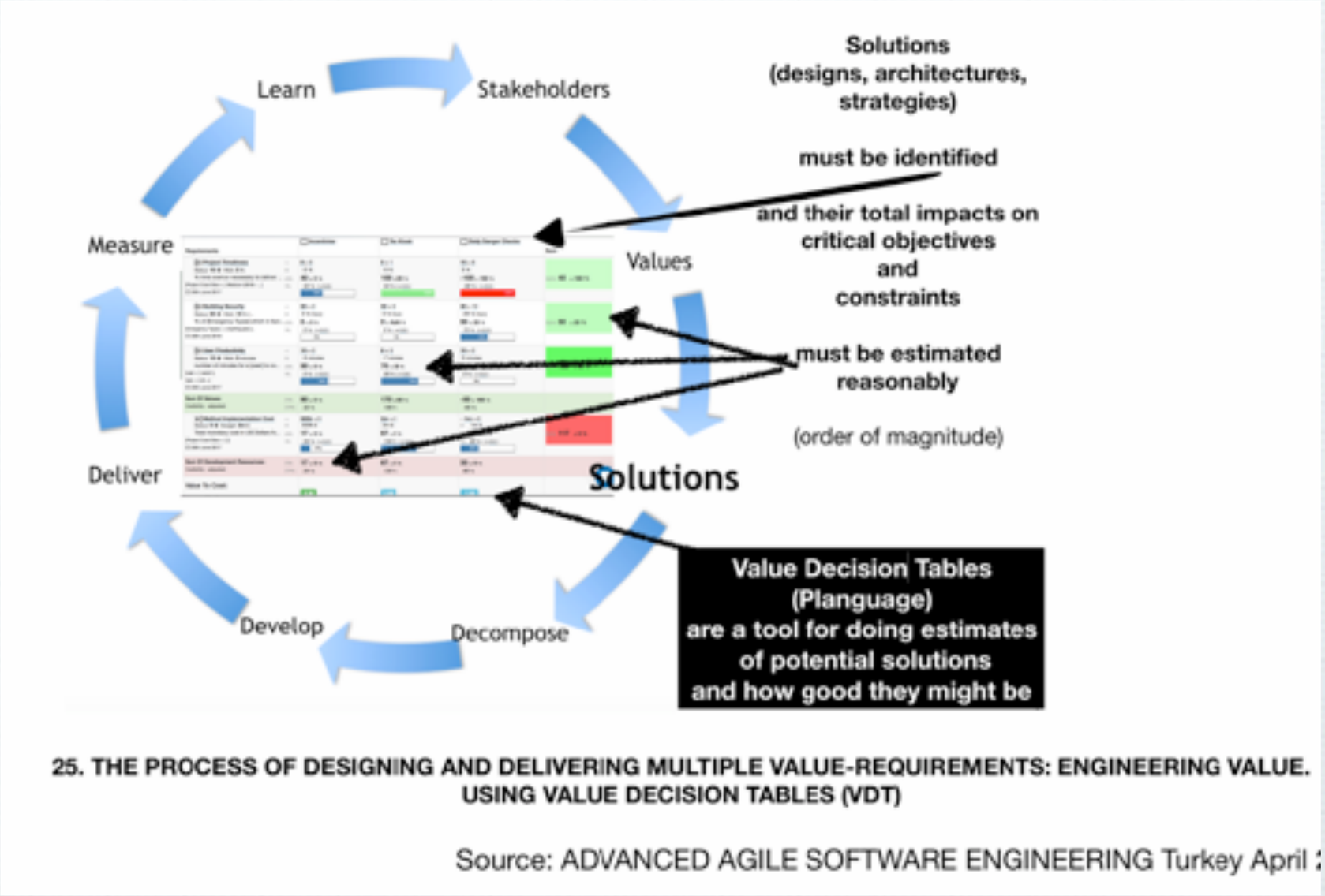


Figure 1.88: A detail window of the 'Ambition' parameter specification.

Sources and change details are there.

A simple way of noting the source of any statement, is to use the keyed Icon '<-'

Part 14.
A briefing on the use of value specifications for design and architecture



25. THE PROCESS OF DESIGNING AND DELIVERING MULTIPLE VALUE-REQUIREMENTS: ENGINEERING VALUE. USING VALUE DECISION TABLES (VDT)

Source: ADVANCED AGILE SOFTWARE ENGINEERING Turkey April

See Value Design book and BCS Course, 3 hours, April 29 2020 https://www.bcs.org/events/2020/april/value-design-how-to-get-the-qualities-you-need-to-win-and-succeed-using-advanced-design-thinking-workshop-based-spa-sg/

Part 15.
A Review of requirements methods compared to Planguage (lecture)

14.1 General observations of methods for specifying Value Requirements

I am quite disappointed in the prevailing culture of dealing with Values, and Value requirements.

That is why I have had to invent my own way.

The current unhealthy requirements culture is very widespread, and new bad methods seem to spring up quickly and spread widely.

But our projects continue to fail, and part of that is bad requirements.

My central criticism is that most methods do not **quantify** the Value requirements at all. And the few that do so, do not do it *well*.

The following material, is for people who would like more-specific background.

They might have to attack some Holy Cows in their 'Temple', in order to deal with these problems.

VR 14.2 A Checklist for understanding capabilities of value requirement Specifications

Here is a basic checklist, I do mentally, to **compare any requirements method with Planguage.**

1. Is the Value Quantified (or is it just nice words?). (“Highly efficient”)
2. Is a re-usable **Scale of measure defined well,** or is an oversimplified badly-defined scale only hinted at, together with the numeric level (“35% agree”)
3. Is the requirement **tagged** in some way, or is it just a bullet point, a sentence, or sub-clause?
4. Is there any systematic way used to **define terms** used in the spec, or are we left guessing at clarity and ambiguity?
5. Is there any **structure** in the Scale similar to our Scale Parameters? How is this variation and definitions of (whom when, why, where) dealt with?
6. Is there any way to annotate of capture the **justification** for a requirement?
7. How do they capture **sources** of requirements ideas?
8. Is there any set of **Rules** for requirement specification which could be the basic for Spec Quality Control: the defect level?

A. Is there any concept of **measured Defect Density**, which could give a basis for Exit from the requirements process?

B. Does the process simply capture a raw ambition level requirement, and leave it at that, or is there an attempt to analyze it and come up with a **better clearer requirement**.

C. Does the requirements process actually **permit ‘designs’ to sneak in as requirements**, when the real requirement is unstated, implied, or badly formulated? (‘We want a password for Security’)

D. Is there any concept of **stakeholders** for the requirements?

E. How good it the capture of **background information**, to help understand quality, risks, relations, priorities?

F. Are **Benchmark** levels systematically captured (Past, Status, Record, Trend)

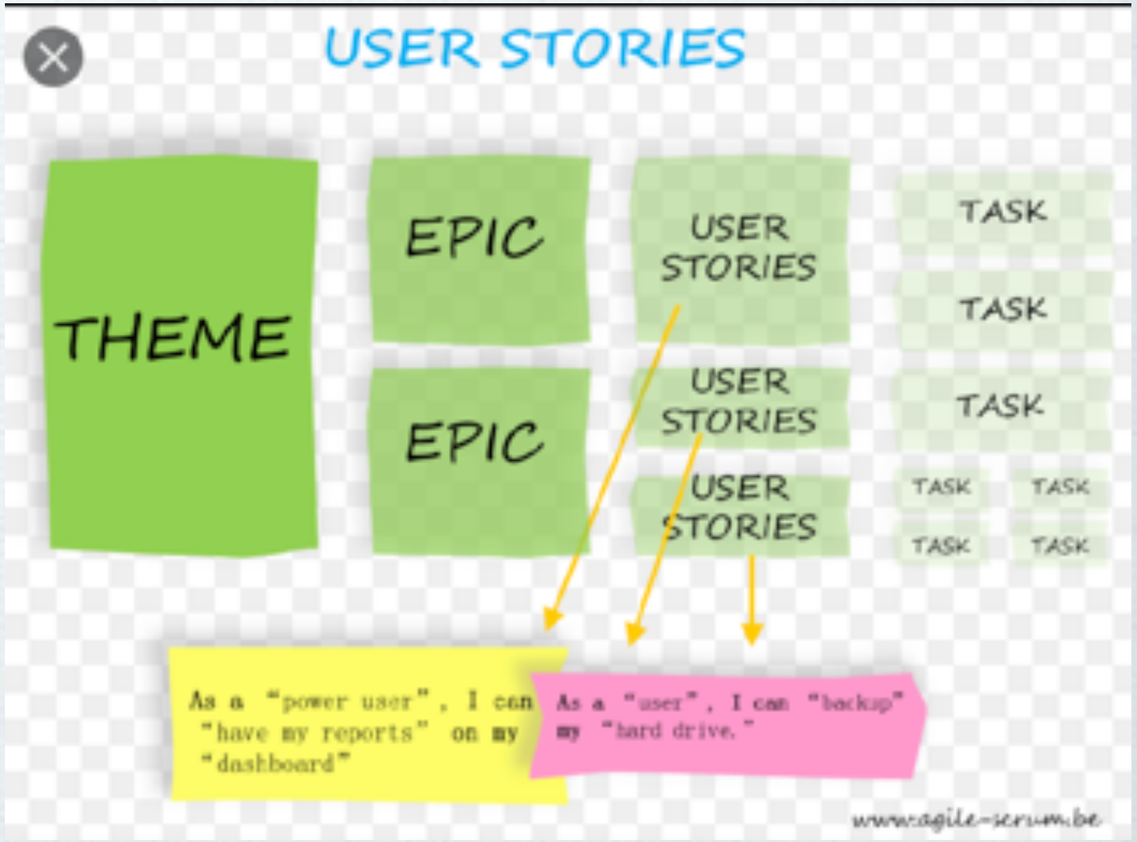
G. Are the requirements suitable for **digital automation**? Can you program visual presentations, and analyze the specs?

H. Is there a **well defined classification and definition of requirements types**? (Function, Resource, Value, Mandatory Design, Constraint, Scalar Constraint, Scalar Target).

I. There is more, but this list should separate *strong* Value spec methods from weaker methods.

15.1 User Stories

- I commented early in this book about User Stories. They are at the level of an Ambition Level, and we can use User Stories to start the process of deeper understanding of the implied Value requirement. But User Stories do not pretend to go into depth themselves.
- My good friend Mike Cohn (Mr. User Stories) specifically referred to our Planguage methods, when asked on his website what to do about qualities and quantification.
- As I said, I like the fact that the User Story does not merely have a 'requirement' idea, but that it specifically includes information about the 'stakeholder' (ok, 'User' only), and the justification (because)
- As a method for 'generating ideas about requirements', and possible values and qualities, for small and less-critical systems (no state of the art competition levels, no huge national health systems) user stories are quite OK.
- My problem is, that I see user stories being used way beyond their 'level of competence', and I think user stories, as a primary requirements culture, are probably one initial cause of project failure.
- Success and failure are not defined by user stories; they are more of a detail. But as we have pointed out earlier, the Value level 'Tolerable level' defines a failure border, and Goal level defines success.
- User stories just do not deal with values and qualities, so we need something more, operating at at a higher level of controlling the system stakeholder results, values, and qualities.
- My advice, if you are committed to using them, is to use them as an Ambition Level, a simplified departure point, and then analyze what the real, but implied-only, 'value level' has to be (derive a Scale and a Wish for example).



15.2 Use Cases

Use Case Success Rate

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

Status 80 Wish 95

Wish [Stakeholder = **User Registered**, Use Case = **(Submit Profile, Make Friends,Join Groups,Post News)**, Actions = **None**] @ 14 Jul 2029 : **95** %

Ambition Level: Do the use cases correctly

Space Cases

Scale:

% Successful Completion of **[Stakeholder]** **[Use Case]** **[Actions]**

Target Time Units: Calendar Date

Actions: defined as:

Uses, Extends

Stakeholder: defined as:

User Registered, Performer, Admin, Event Planner, Visitor

Use Case: defined as:

Submit Profile, Make Friends, Join Groups, Post News, Manage Media Content, Monitor Requests, Add Venue, Select Venue, Plan Event, Find Performer, Publicize Event, Buy Tickets

Source:

Tom Gilb for Use Case example in Chapter 14 Value Requirements book

Status: 80 % [Stakeholder = **User Registered**, Use Case = **(Submit Profile, Make Friends,Join Groups,Post News)**, Actions = **None**] When 14 Jul 2019

Tag.**Wish:**

95 | 14/07/2029 (dd/mm/yyyy) | %

Qualifiers: Copy from... Add additional qualifier

[Stakeholder] =

× User Registered

[Use Case] =

× Submit Profile × Make Friends × Join Groups × Post News

[Actions] =

× None

<——— Using Use cases as a Scale Parameter in Planguage

15.3 Earned Value Management (EVM)

I recommend this EVM overview

https://en.m.wikipedia.org/wiki/Earned_value_management

I would have hoped that EVM would deliver exactly what the name implies. But it does not.

It does not deal with a set of critical Value requirements, at all.

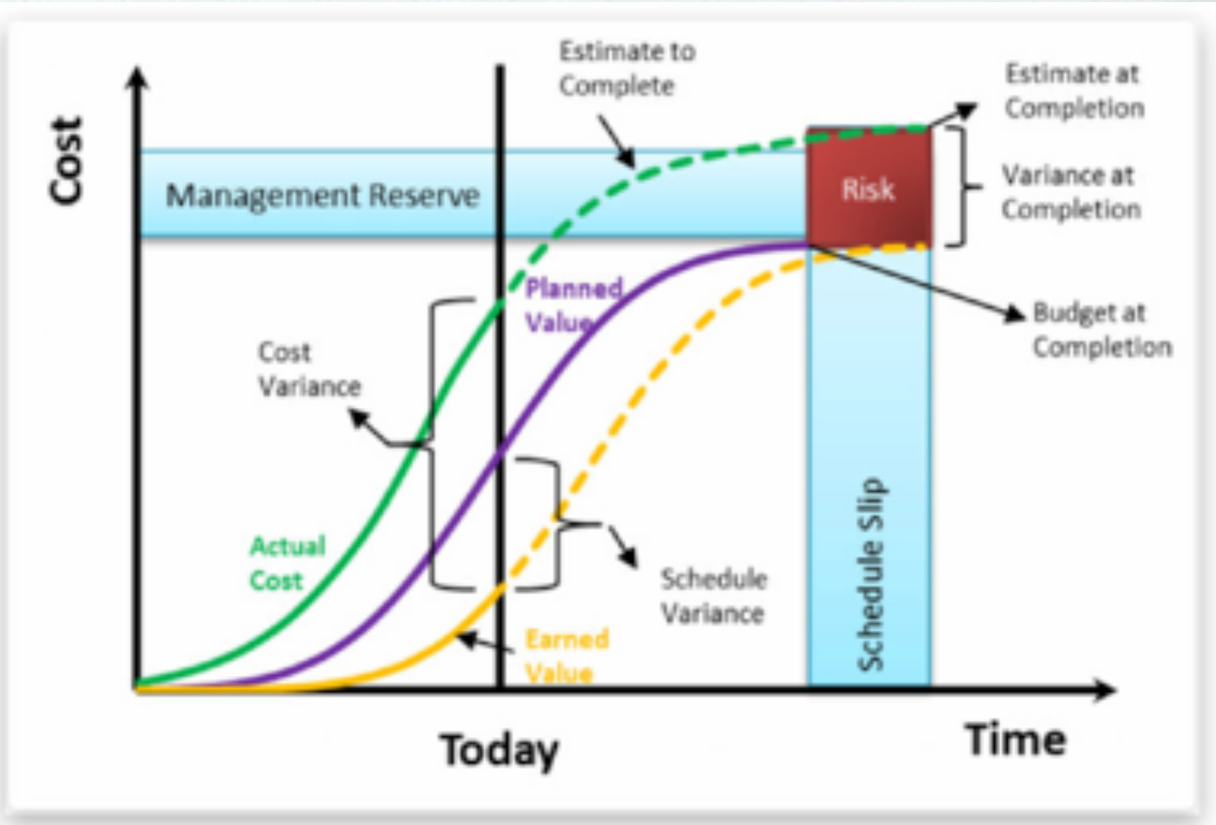
It does assume Big Bang waterfall model pretty much, and 'value' is really just 'work done', or 'tasks', sometimes simply '% of budget spent' !

I recommend the blogs of a professional friend who spends his time fighting for non-corrupted, honest versions of EVM in US Government Projects, https://www.pb-ev.com. Paul Solomon, who has written a book on the subject with another friend that I have worked on several US Government Projects with, RalphYoung.

These guys are honest idealists, so you can trust what they say about EVM.

Of course when a desperate Government, *dictates* EVM, in an attempt to control greedy, and technically incompetent subcontractors, it gets used, and abused.

There is little EVM interest outside of those circles.



https://martinsitconsulting.com/cost-earned-value-management/

15.4 Functional Requirements and Non-functional Requirements

* We fail to define the non functional requirements
* And we fail to focus on delivering these stakeholder values

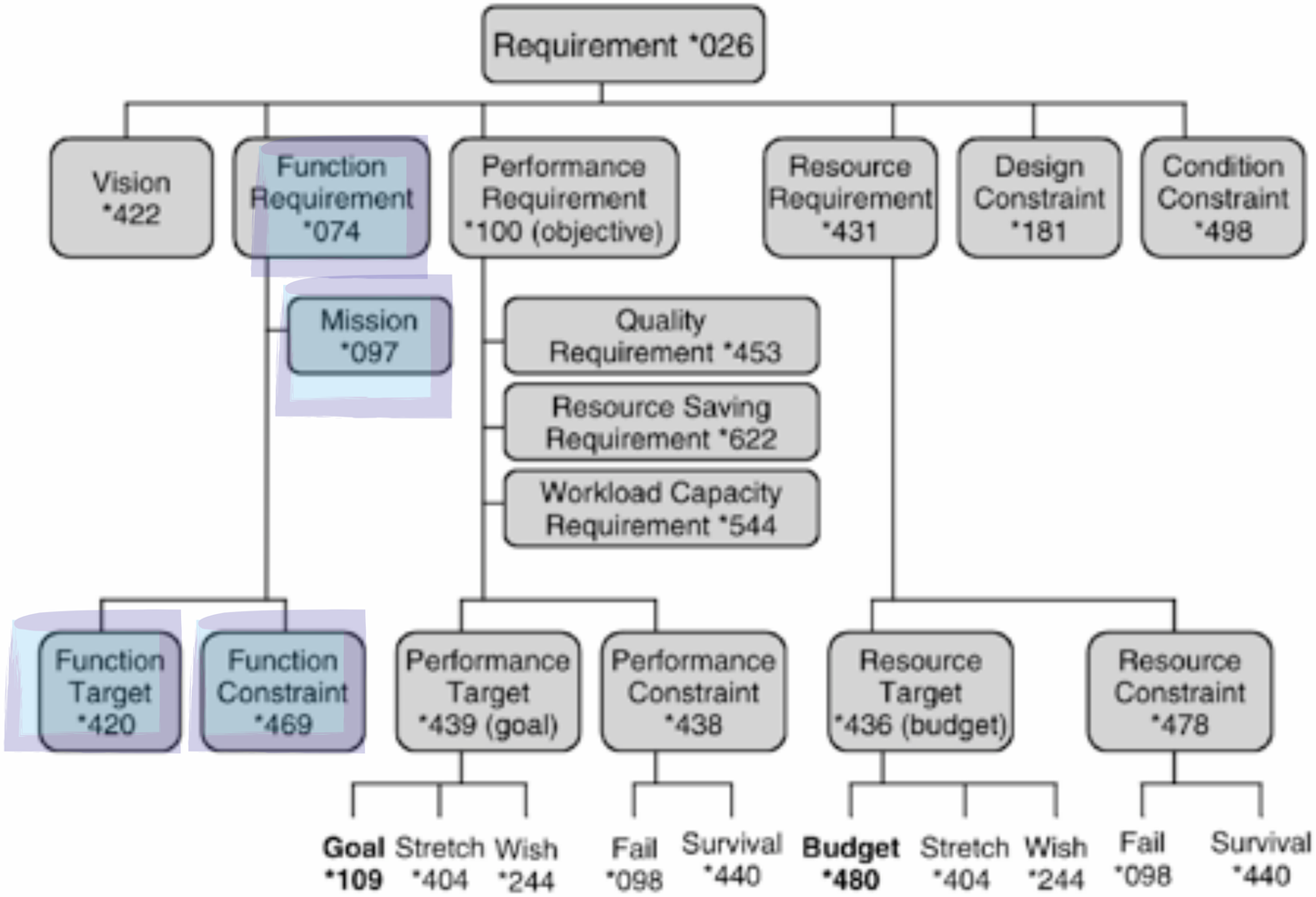


Figure G20
Requirement Concepts.

15.5 Balanced Scorecard

- Non Financials can be totally undefined and unquantified (Better Productivity)
- Irrelevant scales
  - Learning: reuse costs !
- Huge lack of detail
  - Source, Priority, Correltion, Strategies

FINANCIAL OBJECTIVES	TARGET	CURRENT	TO TARGET
Increase Total Revenue	€5.458K	€6.197K	(+13.5%)
Increase Net Profit	€1.208K	€1.173K	(-2.9%)
Increase Profit Margin	25.5%	23%	(-9.8%)
Increase Profit per Customer	€100	€97	(-3%)

LEARNING & GROWTH OBJECTIVES	TARGET	CURRENT	TO TARGET
Open New Sales Regions	4	4	(0)
Reduce Operating Costs	€30K	€26K	(-13%)
Develop New Products per Year	2	3	(+1)
Standardise Staff Training Hours (monthly)	8 HRS	5 HRS	(-3 HRS)

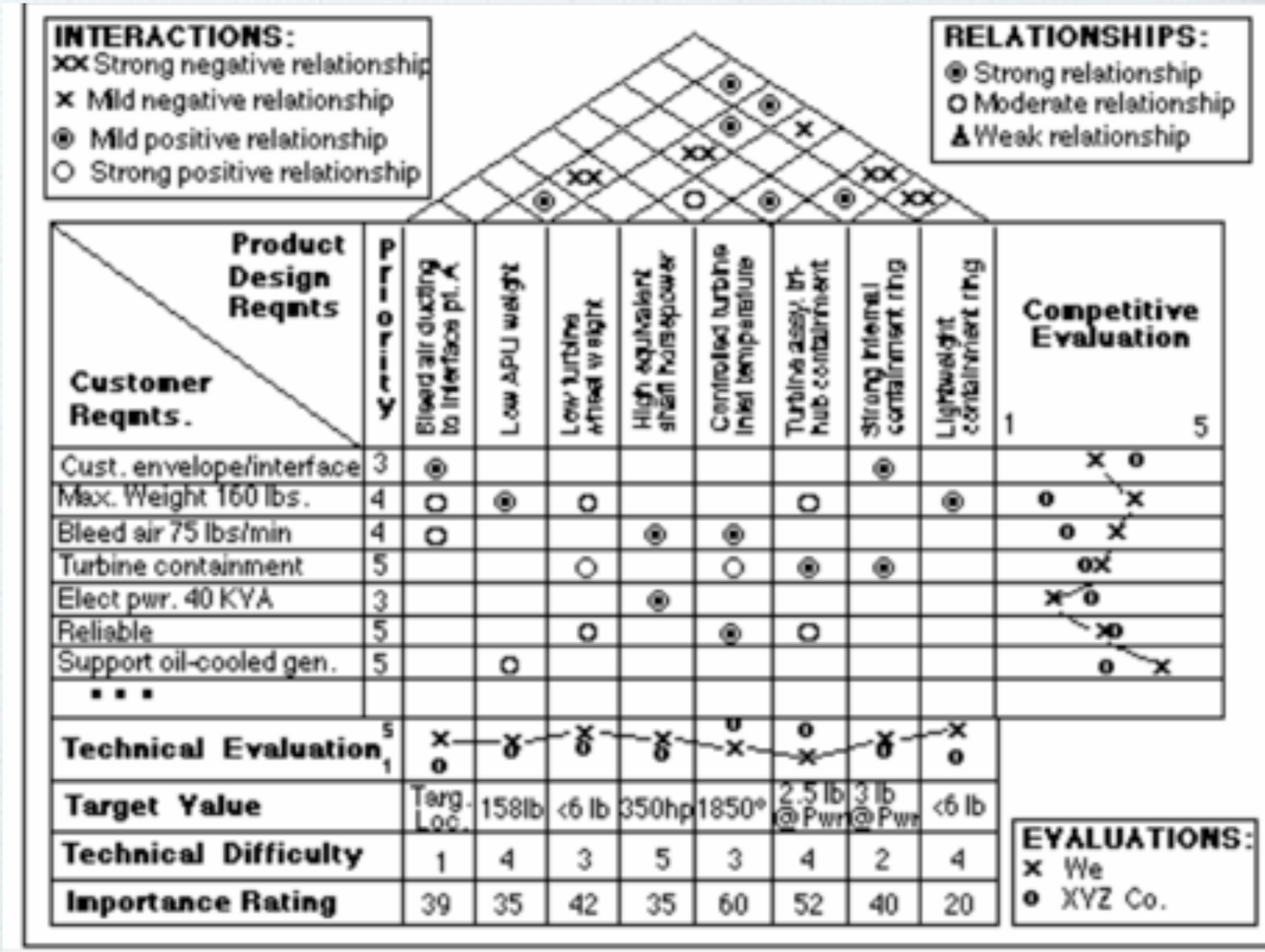
CUSTOMER OBJECTIVES	TARGET	CURRENT	TO TARGET
Increase Active Customers	5300	4.993	(-5.8%)
Increase Customer Signups	876	819	(-6.5%)
Sustain Customer Retention	151	128	(-14.6%)
Improve Customer Satisfaction	85%	73%	(-14.1%)

INTERNAL OBJECTIVES	TARGET	CURRENT	TO TARGET
Reduce Average Absenteeism	6 DAYS	3 DAYS	(-3 DAYS)
Maximise Overall Labour Effectiveness	87%	76%	(-12.6%)
Internally Fill Vancancies	30	29	(-1)
Decrease Average Overtime (per Year)	30	30	(+0)

LIKELIHOOD OF REACHING TARGET: VERY LIKELY · LIKELY · INDETERMINABLE · UNLIKELY · VERY UNLIKELY

15.6 Quality Function Deployment

- No quantification of critical requirements ('Reliable')
- Subjective anonymous priority weights (1 to 5)
- Subjective rough no-evidence relationships
  - (S M W)



How problems with Quality Function Deployment's (QFD's) House of Quality (HoQ) can be addressed by applying some concepts of Impact Estimation (IE)
http://www.gilb.com/DL119

15.7 Togaf

* Does not walk the talk
* Says a lot of nice ideas
* But there is no supporting detail
  * Like how to quantify requirements

My conclusion regarding requirements for Togaf is that the quantified practice they recommend, and I agree with, is simply not taught or practiced.

circle.visual-paradigm.com

ArchiSurance * - Visual Paradigm Enterprise

Dash | Project | ITSM | UeXceler | Diagram | View | Team | Tools | Modeling | Window | Help

... > Document Business Domain Requirements > Architecture Requirements

Open Reference

Architecture Requirements

Business Domain Requirements

Requirement	Description
Digital customer management capability	The Digital Customer Intimacy strategy of ArchiSurance requires the digital channel management capability, to be realized by social media competency and social media apps.
Data-driven insurance capability	The Digital Customer Intimacy strategy of ArchiSurance requires the data acquisition and data analysis capability.

Constraints

ID	Title	Description	Priority	Consequences
C1	Consistenct behavior in different devices.	The interaction patterns between devices (Android, iOS) must be consistent.	High	A unified UX design has to be made for different devices.

Assumptions

ID	Title	Description	Date	Source	Owner
here.		Enter input here.	Input date here (y...	Enter input here.	Enter input here.

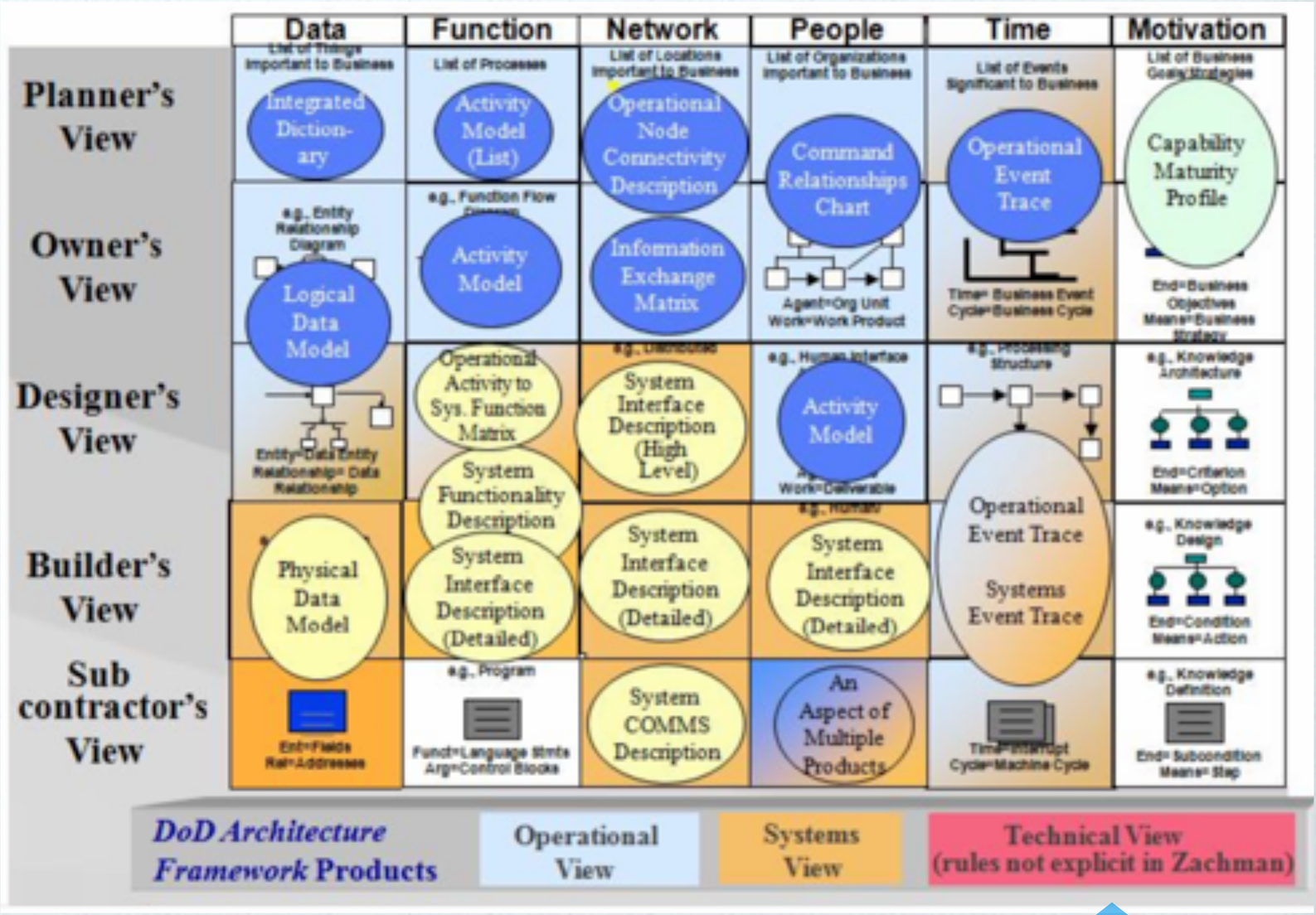
Figure 15.7.2 : Here is the best concrete example I could find , on the internet, of Togaf requirements practice. Just above this example, is text about 'quantified requirements' being mandatory. Do you see any quantification?

15.8 Zachmann Framework

I confess to a weakness for Zachmann.

His framework covers a lot of bases, a lot of the system.

* As this figure reminds us, we cannot expect much detail of a framework.
* It is up to the user of the framework to fill the intersections with specific methodology.



15.9 UML: Unified Modeling Language

- UML models a lot of things,
  - but values, costs and qualities are not amongst them.
- From my point of view this makes UML, and many others like it, quite inadequate for modeling the real world,
  - and some of its most important aspects (Values, qualities, and costs).
- I believe this is due to the built-in in narrow-mindedness of a computer-programming culture,
  - where the program can be constructed without reference to costs and qualities.

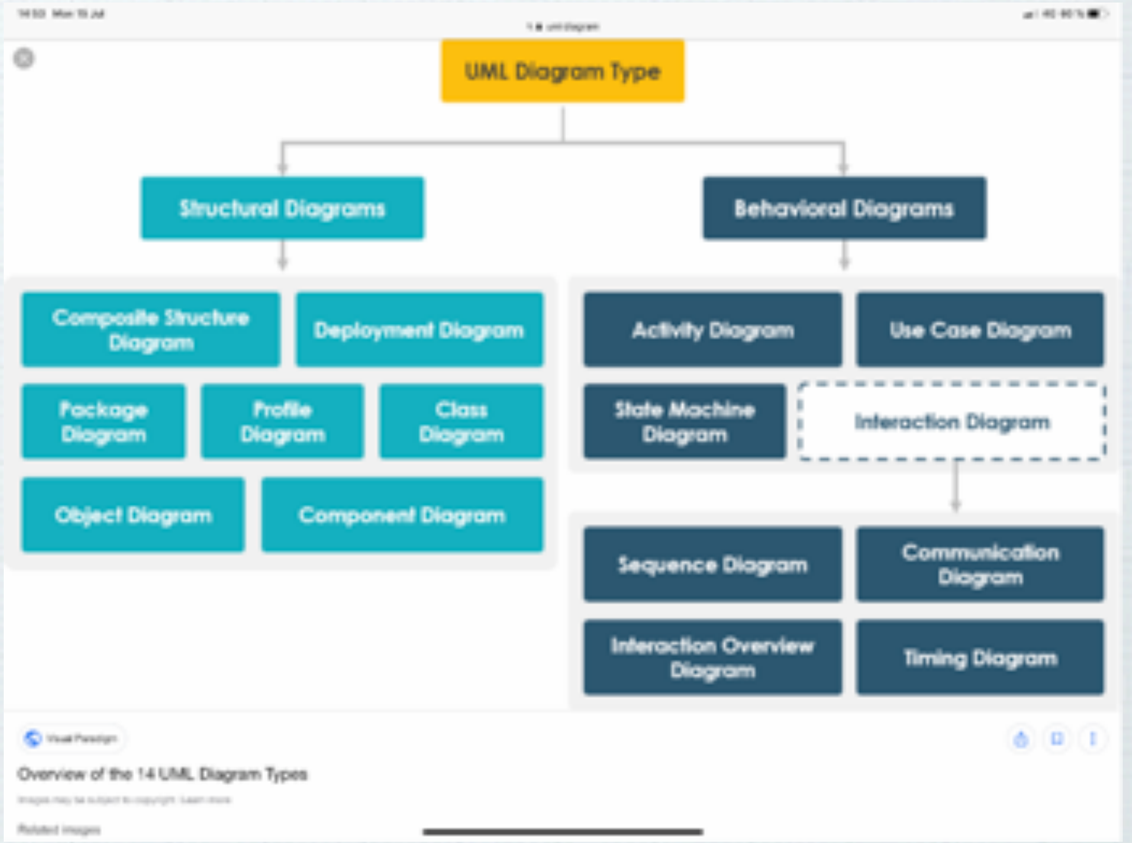


Figure 15.9: a set of UML modeling diagrams.

15.10 Design Sprints

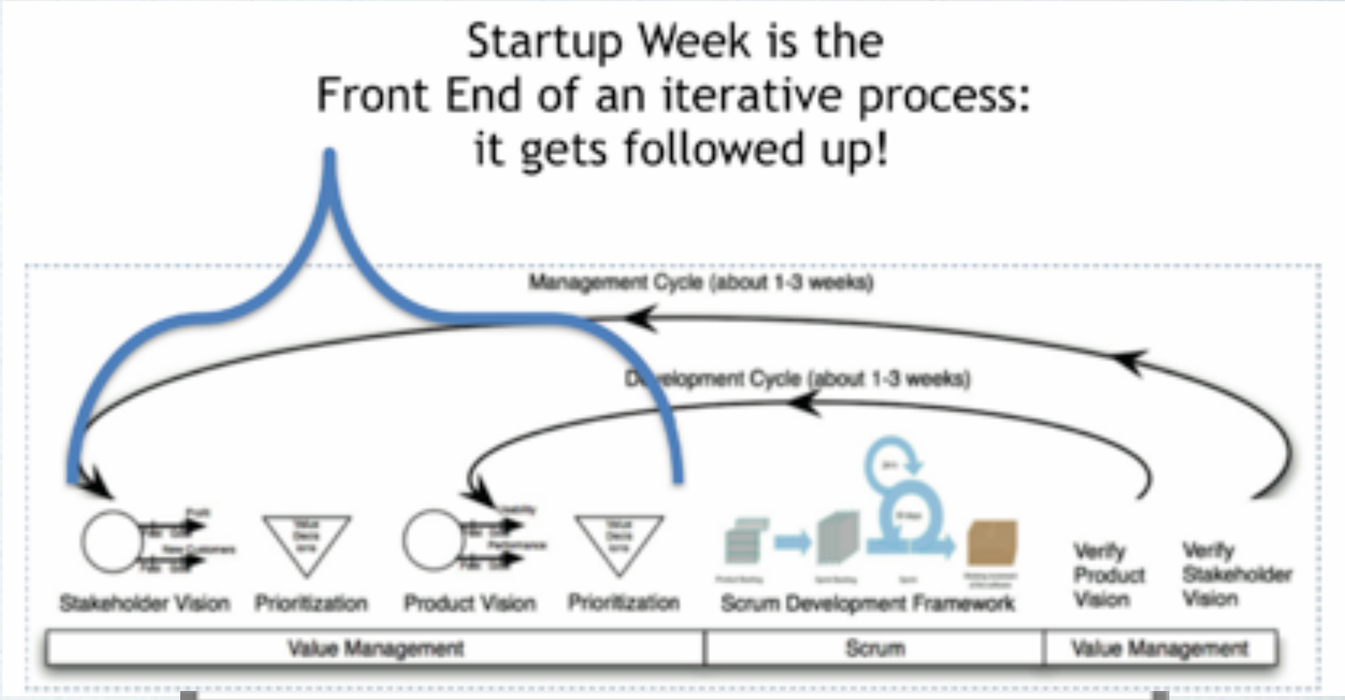
- Design Sprints (1.0, 2.0, 3.0) are getting better, but they do not have any concept of quantification of Values as requirements.
- It is still a yellow-sticky culture, where the emphasis is on finding an app or web design, rather than departing from a clear set of multiple Value and constraint requirements.
- Maybe good for simpler problems: but I have looked and not found any studies comparing Design Sprints to anything else,
  - for example in terms of project success, productivity, value for time spent.
- Planguage offers a structurally similar (week startup) but better startup week idea:
  - The Project Startup Week (ref. K). It has been applied to large banking, aerospace, and defense projects successfully for decades.

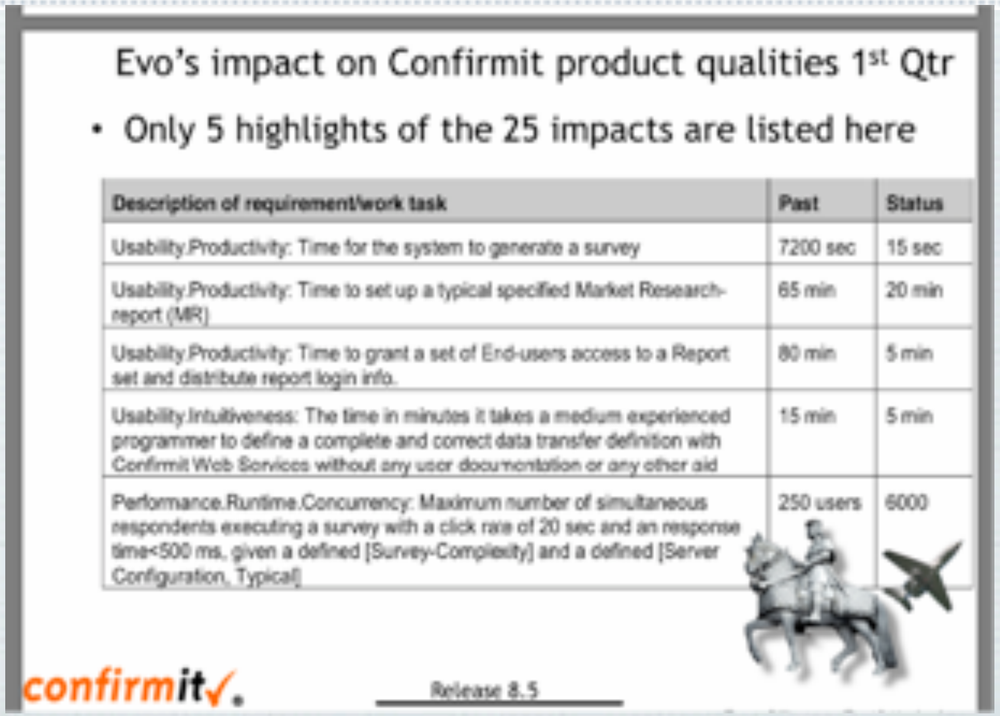
(K) **Project Startup Week**
Agile Project Startup Week Paper in
Gilb's Mythodologies series
gilb.com/dl568

15.1.1 The 'Evo' Project Startup Week Values-Driven Start

- **This startup is primarily driven by a set of quantified critical stakeholder values.**
- **It does not try to get a mock-up, or prototype, working in the first week. It tries to get real measurable results, a value stream, with currently existing products, services and systems.**
- **It tries to learn by stakeholder feedback, and incremental measured results, what works, and what does not.**
-

(K) **Project Startup Week**
Agile Project Startup Week Paper in Gilb's Mythodologies series
gilb.com/dl568





15.12 OKR Objectives and Key Results

- But I have had problems finding any studies of OKR, and even good case studies, which *illuminate the values we get for the costs.*
  - What was the OKR result at Intel? Do they still use OKR?
- OKR is in no way a replacement for Value Requirements,
  - which operate on larger systems (products, organizations, services)
  - and guides us to find ways to create measurable value in the short and long term.
- For clarity, I do not think that our Value Requirements methods are appropriate for this level of individual task planning.
- I would like to think that these same individuals, are all part of some larger projects, and that they are interested in, and committed to, improving Value requirement levels.

Figure 15.12.1 : an example of OKR planning.

Notice the objectives are not quantified and clear ("Create an engaging newsletter").

The 'Key Results' are not business results, they are individual *tasks* (Interview 3 people"),

which the individual *assumes* (hopes?) will produce the vague objective. Good luck!

OKR Objectives and Key Results: what's wrong and how to fix it.
http://concepts.gilb.com/dl879

15.13 MoSCoW: Prioritization Method.

- Prioritization of actions is necessary when resources are limited, as they always are.
- I am not impressed with most well-known prioritization methods,
  - this one included,
  - and especially fixed-weighting methods,
  - as are found in for example Balanced Scorecard, and Quality Function Deployment (see above BSC, QFD).
- But I'll admit that bad prioritization methods are better than none.

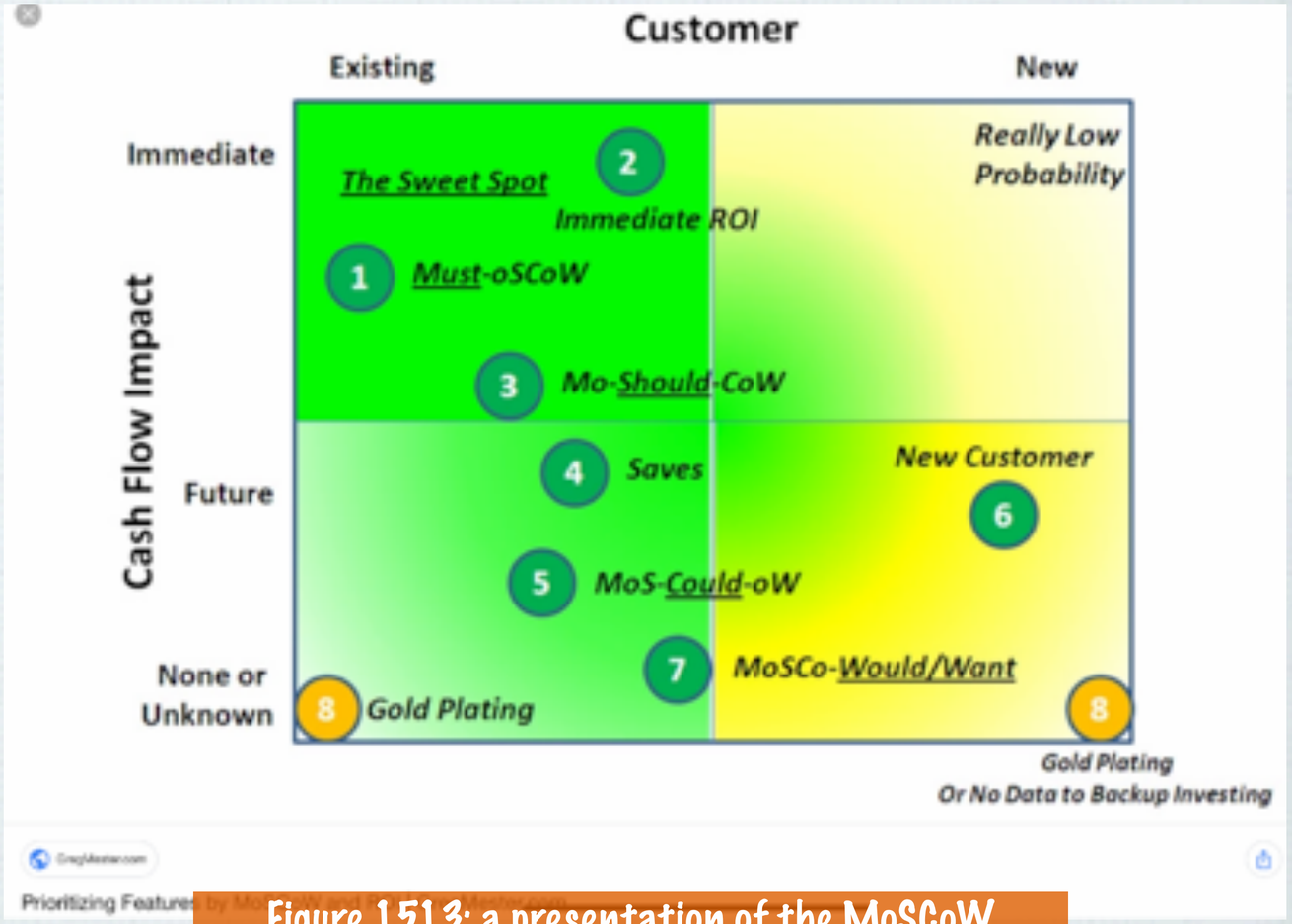
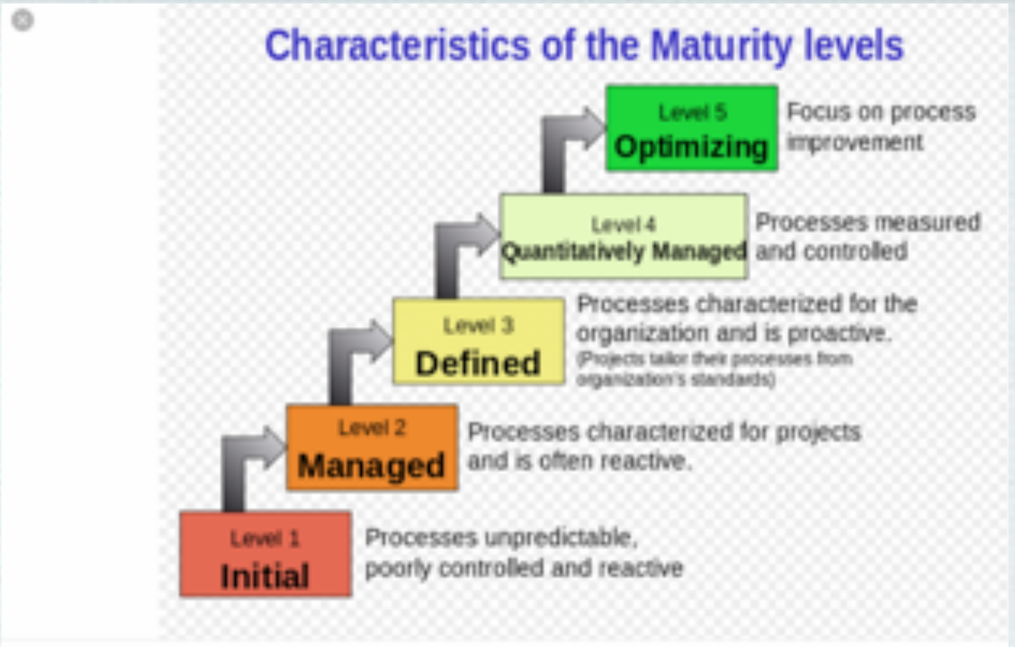


Figure 15.13: a presentation of the MoSCoW prioritization method, which tries to bring in financial and market factors in the decision-making.

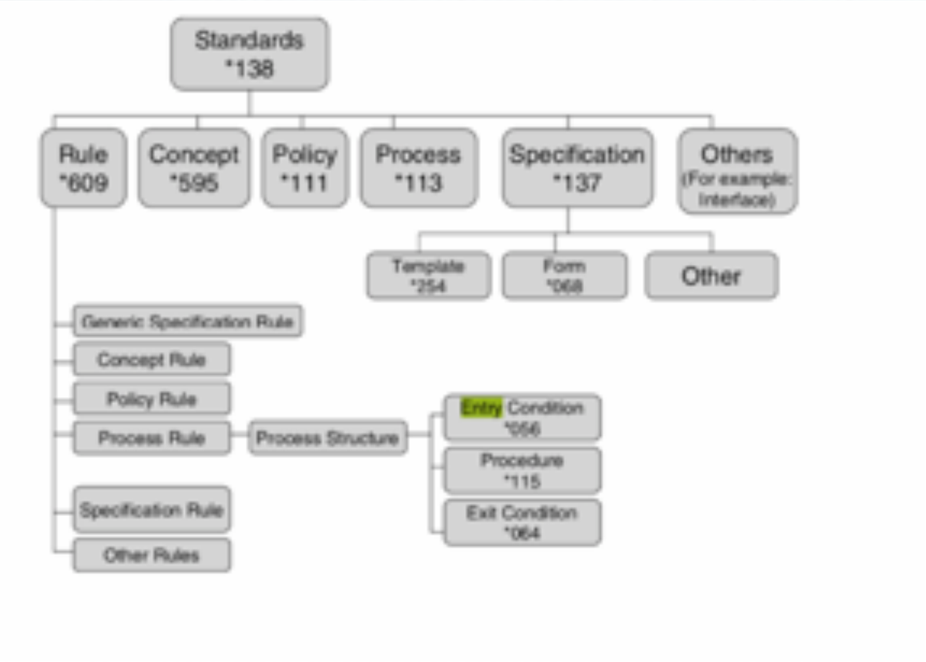
Value Planning book, Chapter 6 Prioritization
https://www.dropbox.com/sh/341lx1a7ckyagxl/AAA0pDzSxN5WmoP9l0KR0Mpca?dl=0

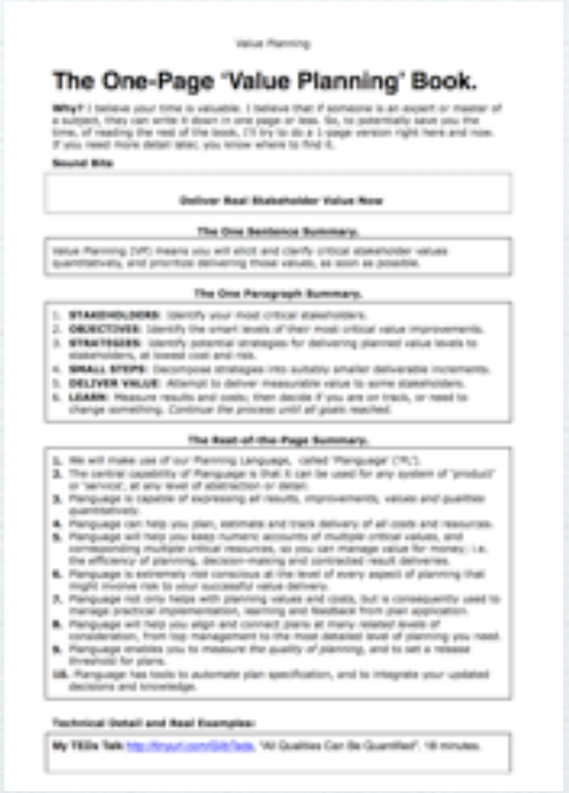
15.14. CMM: CMMI, Capability Maturity Model

- The CMM level 4, 'quantitatively managed' is mainly based on my 1976 Software Metrics book ideas,
- according to Ron Radice, who invented it at IBM.
- But I think the focus on development processes, rather than stakeholder value, is a mistake.
- We need stakeholder value first, and *process* cost-effectiveness second.



Part 16.
Formal standards for Value Specifications





Value Planning

The One-Page 'Value Planning' Book.

Why? I believe your time is valuable. I believe that if someone is an expert or master of a subject, they can write it down in one page or less. So, to potentially save you the time, of reading the rest of the book, I'll try to do a 1 page version right here and now. If you need more detail later, you know where to find it.

Sound Bite

Deliver Real Stakeholder Value Now

The One Sentence Summary.

Value Planning (VP) means you will elicit and clarify critical stakeholder values quantitatively, and prioritize delivering those values, as soon as possible.

The One Paragraph Summary.

1. **STAKEHOLDERS:** Identify your most critical stakeholders.
2. **OBJECTIVES:** Identify the smart levels of their most critical value improvements.
3. **STRATEGIES:** Identify potential strategies for delivering planned value levels to stakeholders, at lowest cost and risk.
4. **SMALL STEPS:** Decompose strategies into suitably smaller deliverable increments.
5. **DELIVER VALUE:** Attempt to deliver measurable value to some stakeholders.
6. **LEARN:** Measure results and costs; then decide if you are on track, or need to change something. *Continue the process until all goals reached.*

The Rest-of-the-Page Summary.

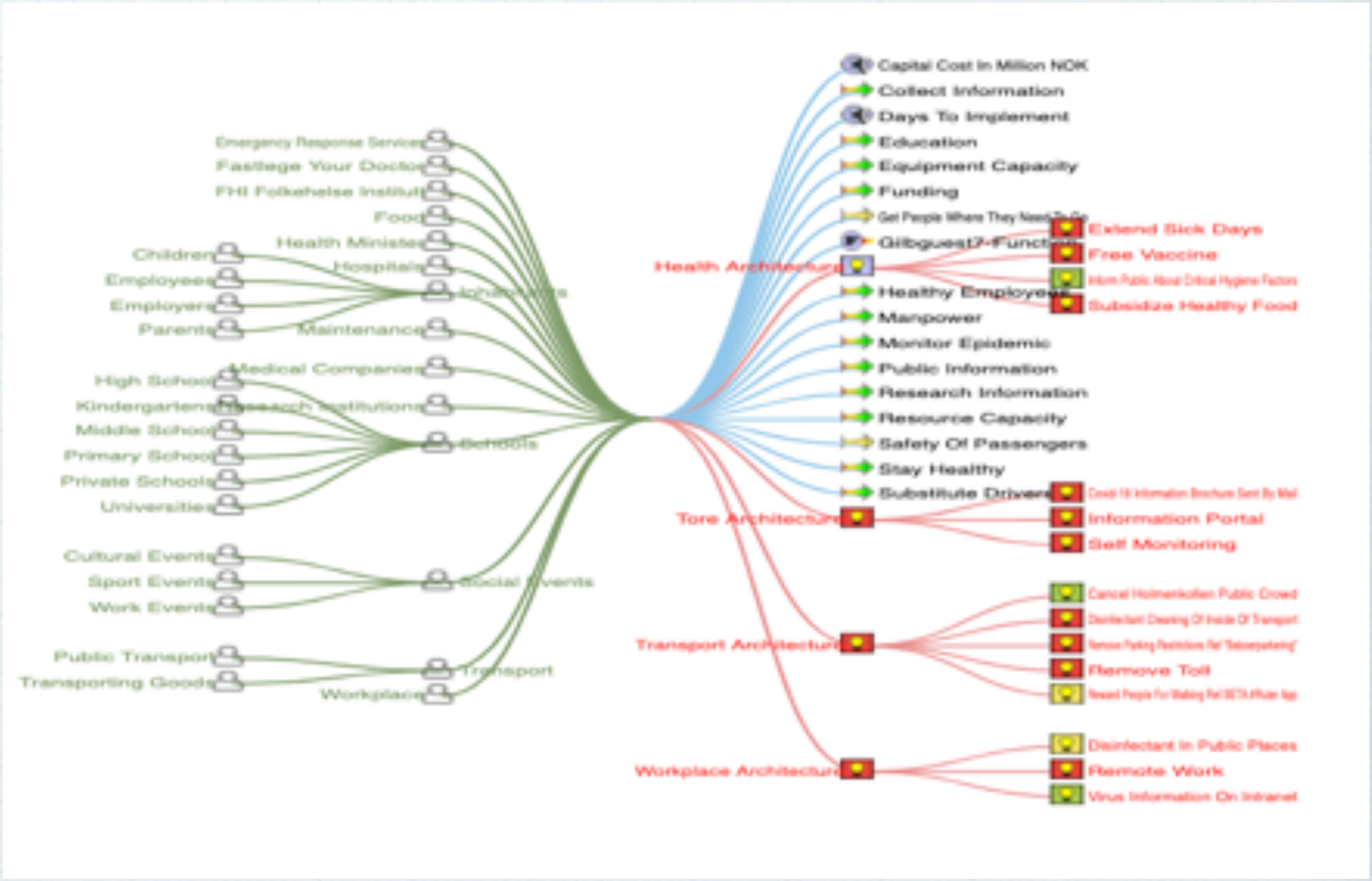
1. We will make use of our Planning Language, called 'Planguage' ('PL').
2. The central capability of Planguage is that it can be used for any system of 'product' or 'service', at any level of abstraction or detail.
3. Planguage is capable of expressing all results, improvements, values and qualities quantitatively.
4. Planguage can help you plan, estimate and track delivery of all costs and resources.
5. Planguage will help you keep numeric accounts of multiple critical values, and corresponding multiple critical resources, so you can manage value for money; i.e. the efficiency of planning, decision-making and contracted result deliveries.
6. Planguage is extremely risk conscious at the level of every aspect of planning that might involve risk to your successful value delivery.
7. Planguage not only helps with planning values and costs, but is consequently used to manage practical implementation, learning and feedback from plan application.
8. Planguage will help you align and connect plans at many related levels of consideration, from top management to the most detailed level of planning you need.
9. Planguage enables you to measure the quality of planning, and to set a release threshold for plans.
10. Planguage has tools to automate plan specification, and to integrate your updated decisions and knowledge.

Technical Detail and Real Examples:

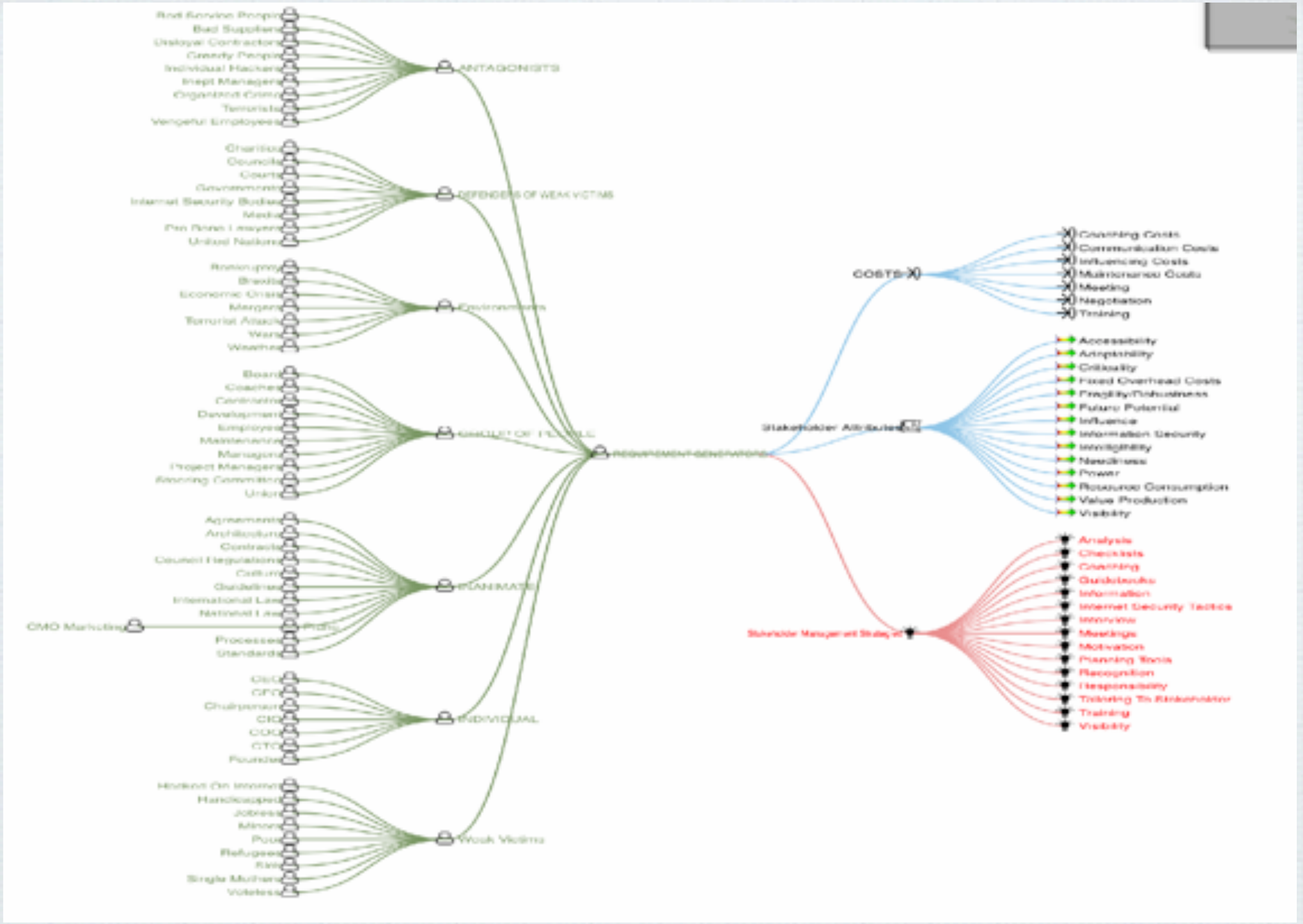
My TEDx Talk http://tinyurl.com/Gilb Tedx, 'All Qualities Can Be Quantified'. 18 minutes.

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

Part 17.
'Valplan' and other apps for Value specification. Make your own app in Excel!



Part 18.
Stakeholders and the planning environment


ANTAGONISTS
DEFENDERS OF WEAK VICTIMS
Environments
INANIMATE
INDIVIDUAL
Weak Victims
CMO Marketing
COSTS
Coaching Costs
Communication Costs
Influencing Costs
Maintenance Costs
Meeting
Negotiation
Training
Accessibility
Adaptability
Criticality
Fixed Overhead Costs
Fragility/Robustness
Future Potential
Influence
Information Security
Intelligibility
Neediness
Power
Resource Consumption
Value Production
Visibility
Analysis
Checklists
Coaching
Guidebooks
Information
Internet Security Tactics
Interview
Meetings
Motivation
Planning Tools
Recognition
Responsibility
Training
Visibility

Last Slide

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

Pictures from Spring edition of the Masterclass

7 / 41

BCS will publish a URL for this 3 hour course after the live presentation

GO BACK TO LAST SLIDE