

# Value Agile

BCS SPA 2 Hours Digital Course  
Co-sponsor Specialist Group Quality

For initial presentation  
Wednesday 20th May 2020, 18:00 to 20:00 UK +

Video URL= [https://www.youtube.com/playlist?list=PLK8hokJ0qd3\\_wlvr0j85YhmNfNj8ZJ8M-](https://www.youtube.com/playlist?list=PLK8hokJ0qd3_wlvr0j85YhmNfNj8ZJ8M-)  
(General site of videos, SPA and my courses and talks)

The following videos are there now: Technoscopes, Value Requirements, Value Design, Value Management

Slide Location Pdf : = <http://concepts.gilb.com/dl974>

Value Agile Slides Folder (PPTX or Keynote slide copy)

<https://www.dropbox.com/sh/qfkgv4slajv3s0m/AAAHAS-w7AV5lxTzDbwa7k6na?dl=0>

The 'Value Agile' free Book: [tinyurl.com/ValueAgile](http://tinyurl.com/ValueAgile)

By Tom Gilb, in Norway (Kolbotn, near Oslo)

[tom@Gilb.com](mailto:tom@Gilb.com) (questions welcome)

[www.Gilb.com](http://www.Gilb.com) (lots of materials!)

@ImTomGilb (Twitter).

[www.linkedin.com/in/tomgilb](http://www.linkedin.com/in/tomgilb)

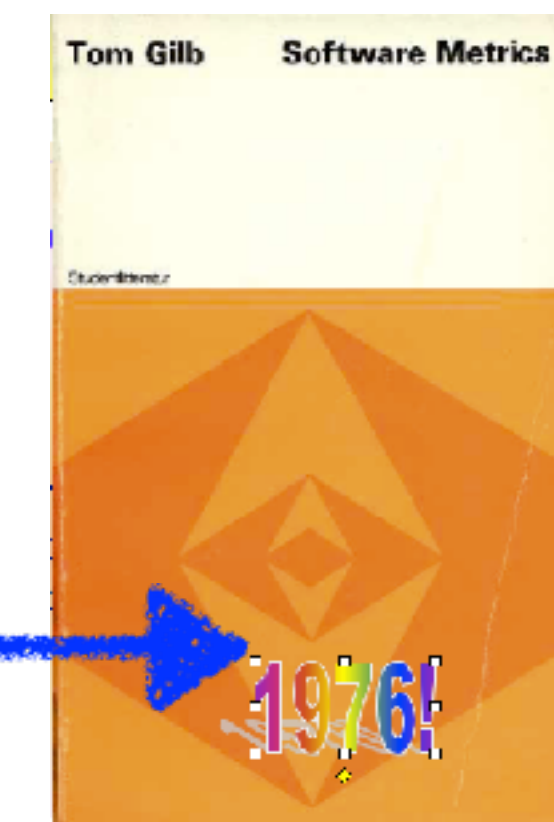
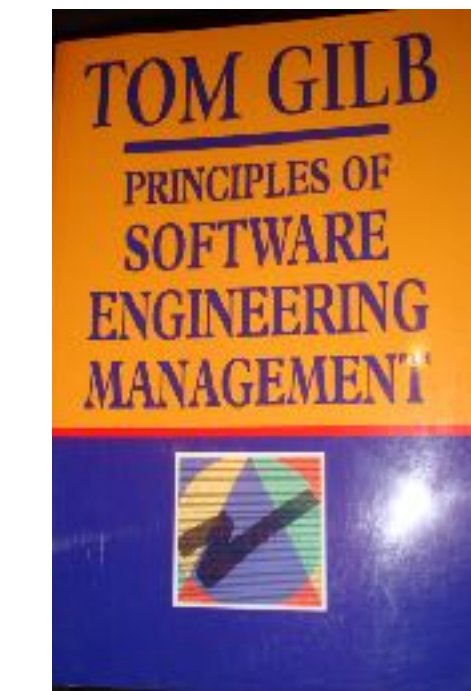


Pictures from Spring edition of the Masterclass  
5 / 41



# Agile Credibility

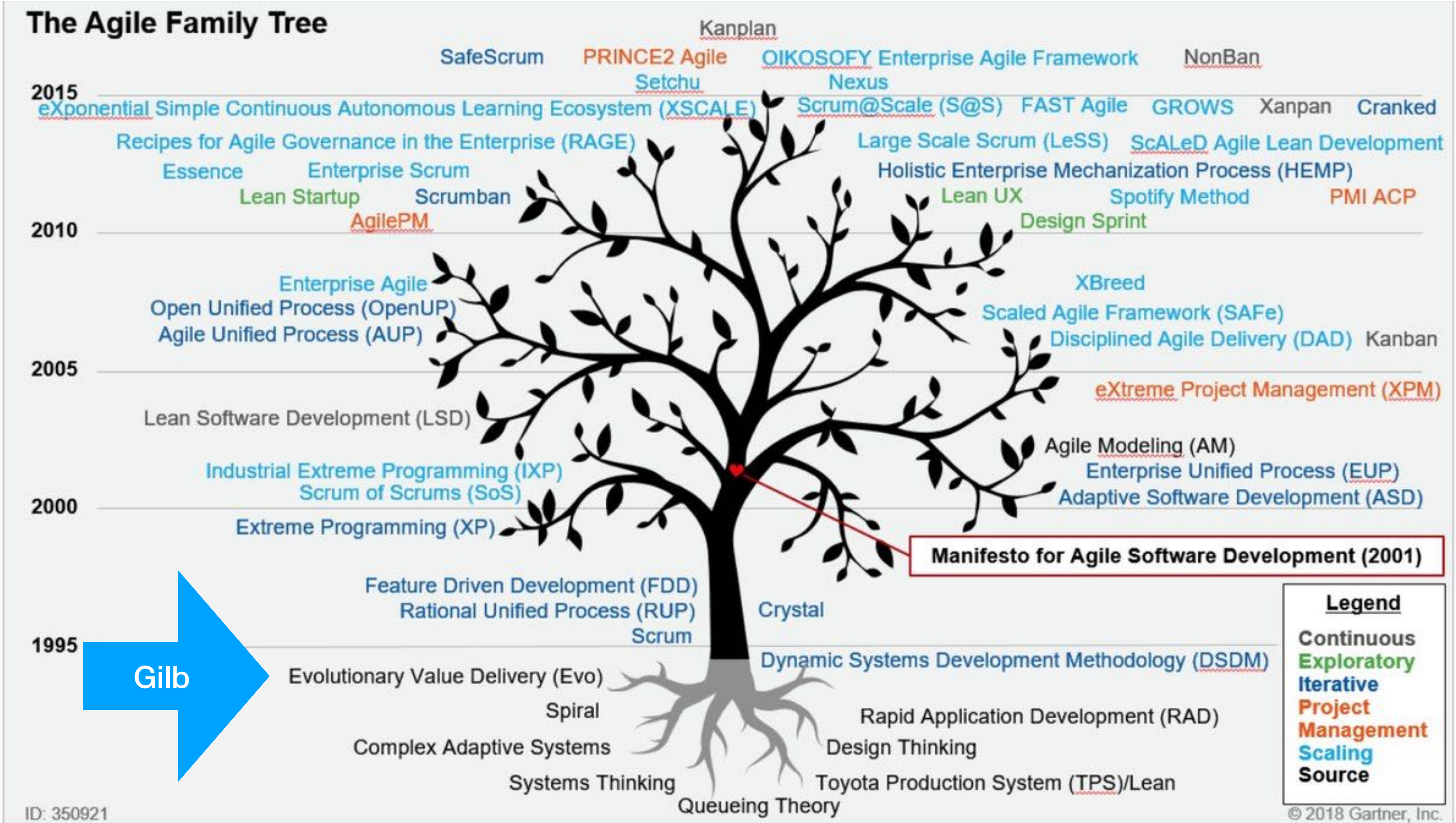
- Agile ‘Grandfather’ (Tom)
  - Practicing ‘Agile’ IT Projects since 1960 (Dobloug, Oslo, 20 value delivery steps)
  - Preaching Agile since 1970’s (Computer World, Gilb’s Mythodology column UK, and other outlets)
  - Acknowledged Pioneer by Agile Manifesto Gurus, and Research
    - See Presenter’s Notes to this slide for detail (Sutherland, Cohn, Beck, Highsmith, Martin)
- Agile Practice
  - IT: decades (Kai and Tom)
  - Organisations: Decades. Some selected examples.
    - Citigroup, JP Morgan, Deutsche Bank, UBS, Credit Suisse, US DoD, Siemens
    - Intel, HP, Boeing, Confrimit AS 2003, Universitetsforlaget 1968, Ericsson, NTNU IT, Philips
- Books (with clear agile content):
  - ‘Software Metrics’ (1976) \*\*
  - ‘Principles of Software Engineering Management’ (1988)
  - ‘Competitive Engineering’ (2005)
  - ‘Evo’: (Kai, evolving, 55 iterations)
  - ‘Value Planning’ (2014-2019). [gilb.com](http://gilb.com)
  - 5 Books in 2018 (see [gilb.com](http://gilb.com)):
    - Life Design, Innovative Creativity, 100 Project Planning Principles, Technoscopes, Clear Communication
  - 5 Books in 2019: Value Requirements, Value Design, Value Management, Value Agile, Sustainability Planning (all free digitally at the moment) [https://www.dropbox.com/sh/adcrki52xo5zb36/AABMD\\_2GOX4rT6c-HRCmT-Qua?dl=0](https://www.dropbox.com/sh/adcrki52xo5zb36/AABMD_2GOX4rT6c-HRCmT-Qua?dl=0)



See this slide’s Presenter Notes for more detail on Credibility, citations from others, even 1976 SM book quotes), LIKE :

**\*\* ‘A complex system will be most successful, if it is implemented in small steps, and if *each* step has a clear measure of successful achievement, as well as a "retreat" possibility to a previous successful step, upon failure.’**  
(SM BOOK 1976 p. 214)







This course, and these slides,  
are  
Based on my Digital Book 2019. —>

- \* A Copy is free for you,
- \* as another way for your to review the course material,
- \* and share it with other people

PS if anybody wants to make paper editions or translations, whole or part, talk to me



£1,000 Normal Price.  
Free for People  
who are generous  
with help and wisdom

[tinyurl.com/ValueAgile](https://tinyurl.com/ValueAgile)



# Book Contents, Course Outline

WHO IS THIS BOOK FOR?	2
Introduction	4
How Well Does the Agile Manifesto Align with Principles that Lead to Success in Product Development?	4
Background	4
Chapter 1. The Four Values of the Agile Manifesto	7
Value 1. Individuals and interactions over processes and tools	9
Value 2. Working software over comprehensive documentation	11
Value 3. Customer collaboration over contract negotiation	13
VALUE 4. ‘Responding to change over following a plan’	16
Chapter 2.	18
The Twelve Agile Manifesto Principles	18
PRINCIPLE 1: ‘Our highest priority is to satisfy the customer through early and continuous delivery of valuable software’.	18
PRINCIPLE 2. ‘Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage’.	21
PRINCIPLE 3. ‘Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale’.	24
PRINCIPLE 4. ‘Business people and developers must work together daily throughout the project’.	26
PRINCIPLE 5. ‘Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done’.	30
Principle 6. ‘Enable face-to-face interactions’.	33
Principle 7. ‘Working software is the primary measure of progress’.	38
PRINCIPLE 8. ‘Agile processes promote sustainable development.	40
The sponsors, developers, and users should be able to maintain a constant pace indefinitely’.	40
PRINCIPLE 9. ‘Continuous attention to technical excellence and good design enhances agility’.	43
PRINCIPLE 10. ‘Simplicity - the art of maximizing the amount of work not done - is essential’.	46
Principle 11. ‘The best architectures, requirements, and designs emerge from self-organizing teams’.	48
Principle 12. ‘At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly’.	50
References for the ‘agile manifesto’ chapter	52
	94

CHAPTER 3:	59
Principles of Project Failure: How to sabotage a project, without anyone noticing you.	59
THE MAIN PRINCIPLES OF FAILURE	66
1. Do not analyze stakeholders, stick to ‘customers’ and ‘users’.	66
2. Do not clarify stakeholders values. Give them the technology they say they want.	71
3. Commit to all the ‘nice-sounding’ designs and strategies. Especially the ones on the managers’ PowerPoint slides.	78
4. Make use of the most widespread project development methods.	81
Popularity is a sure sign of oversimplified training, and methods which do oversimplify training, have failure rates that are over 50%, for years on end, and no one does anything effective about it.	81
5. Make sure no one ever estimates how effective a design or strategy will be. Or what it will cost in the short term or long term.	84
Such estimates are rarely perfect and might distract from using perfectly nice and modern-sounding designs.	84
Like AI, blockchain. Or big data.	84
6. For goodness sake. Do not waste energy trying to estimate the side-effects of exciting strategies, on your critical objectives and costs.	86
Such insights would delay your ‘will to get on with it’, and overrun the deadline.	86
7. No cure, NO pay	89
Summary	92
References. Project Failure	
Chapter 4	99
What is ‘agile As it should be’?	99
The real agile	99
Let’s define ‘cost-effectiveness’	100
What are the consequences of ‘agile cost-effectiveness’ as a tool?	102
What about the core agile idea: ‘adaptability’ to changes, and to new insights?	104
Value Agile	105



# Who is 'Value Agile' for?

- \* Open-minded IT and Systems professionals
  - \* Not for Agile Programmers
    - \* This is NOT the 'agile programming' process
  - \* But this is for Agile System Developers
  - \* For successful and sustainable systems
- \* Who want improve the world (at least IT)
- \* By really delivering Value to Stakeholders
- \* And are prepared to work hard and long to influence people who (being human)
  - \* Prefer 'simplistic methods' (like agile programming)
  - \* Even if they fail far too frequently
    - \* (Google 'Agile IT Failure Rates'), the facts are out there.
  - \* and, to influence people who are 'in denial' about that failure level
    - \* "it is 6 x faster, but only 40% failure" (JS)
- \* For Value Agile Leaders:
  - \* People who want to lead improvement, in successful IT-and-Systems projects
  - \* More bluntly: it's for people who want to get, and keep, a good job.
    - \* Succeeding clearly quickly. Where others 'fail, and make excuses, or do not care'.



Next week 'Sustainability Planning'  
<https://tinyurl.com/UNGoalsGilb>

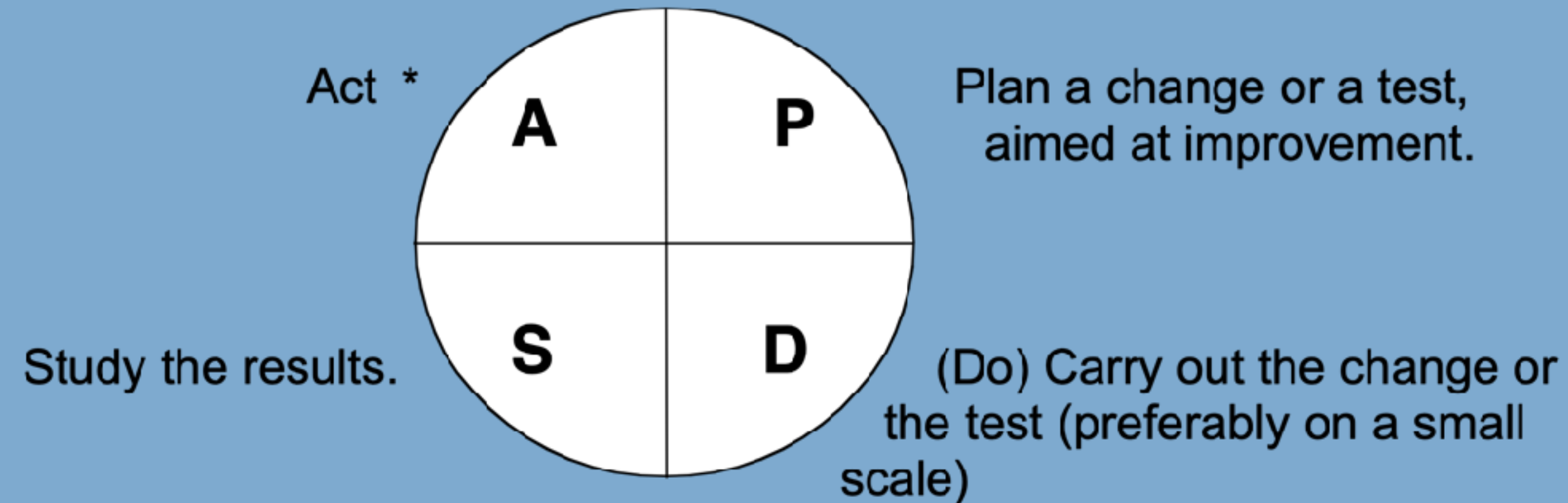


# 'SURVIVAL IS NOT MANDATORY' (DEMING)



My Teacher  
And fellow Ballet  
Aficionado

"The Shewhart Cycle for Learning and Improvement  
The P D S A Cycle



Act. Adopt the change, or Abandon it, or Run through the cycle again, possibly under different conditions. "

Exact reproduction (- '(Do)' from a letter to Tom Gilb from W. Edwards Deming 18 May, 1991



\*"Deming/Shewhart Cycle' is an early method formalization of incremental result delivery (agile).

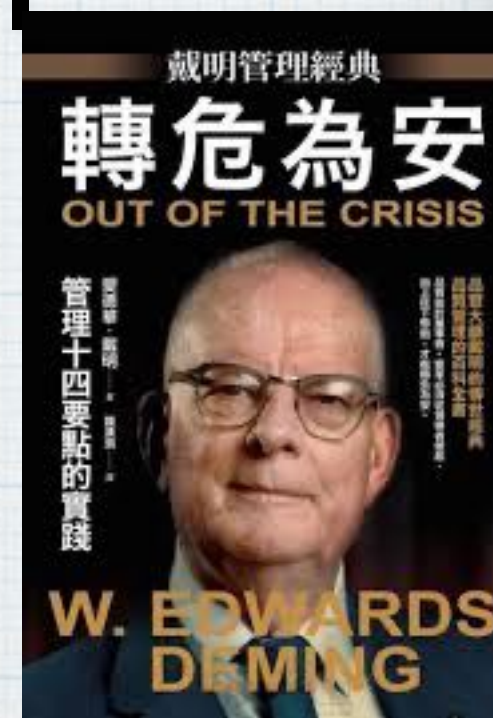
\* Long before 'software'.

\* He is saying that

\*if you make *bad choices* in *your* development methods,

\*you might totally fail.

\*But that is not *his* problem.



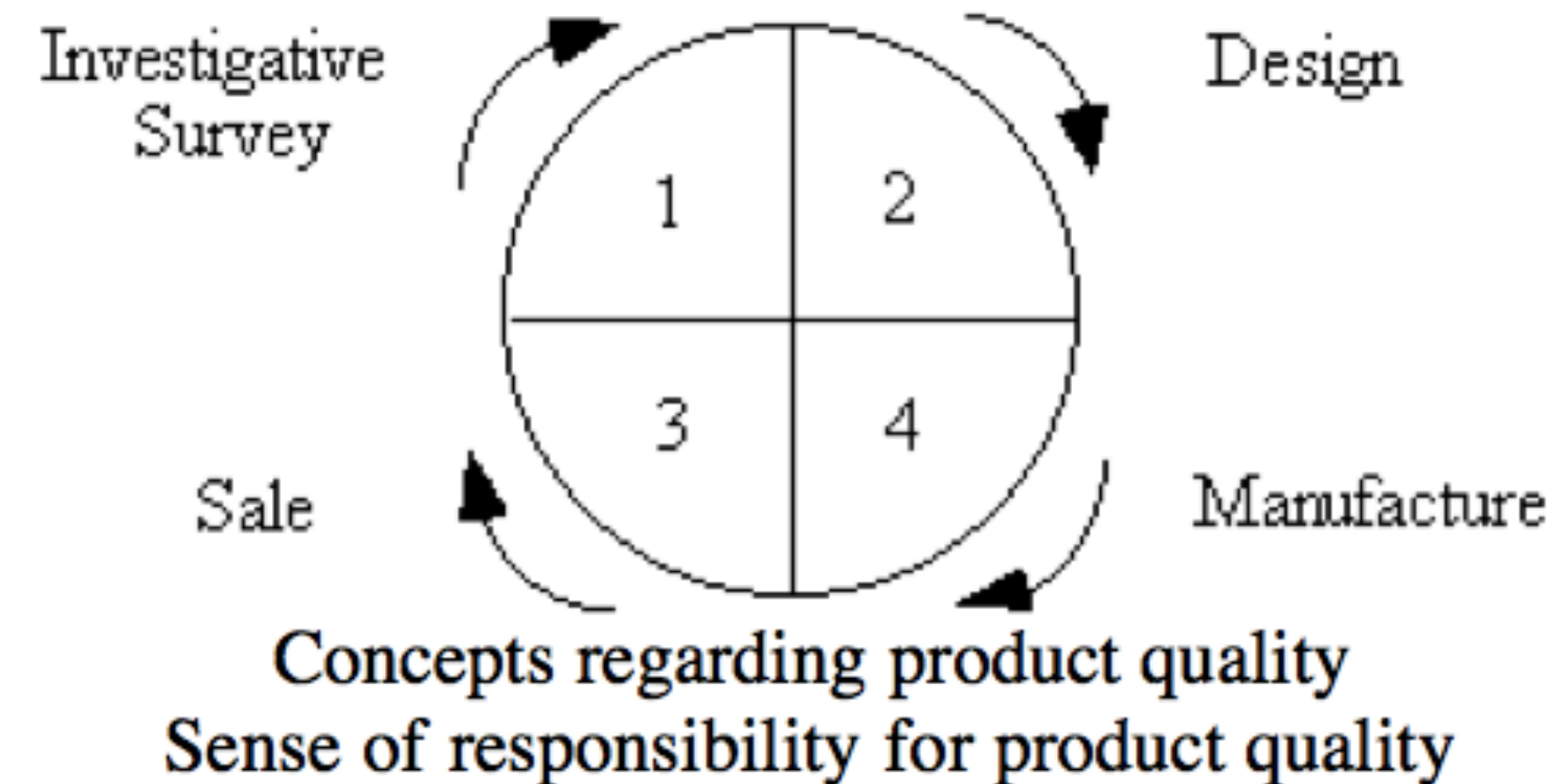


# FROM DEMING'S 1950 SPEECH TO JAPANESE



Agile  
From  
1950  
And  
before  
That  
1920s  
Shewhart

At the end of my discussion of market surveys, I would like to explain my thoughts on the problem of statistical product quality administration with a diagram. This diagram not only makes clear my thoughts on product quality administration and market surveys, but I think it is extremely easy to understand. Below I have drawn a pie graph "wheel" divided into four sections:

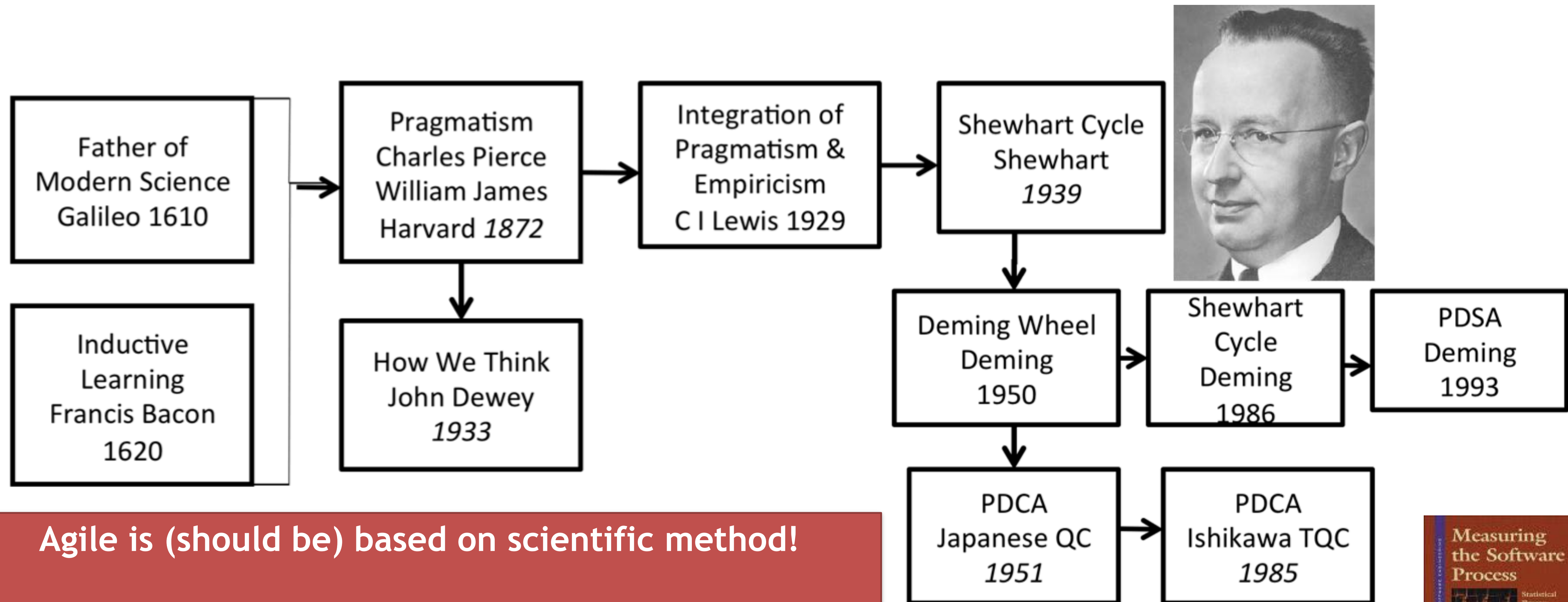


This wheel rolls along the line of "concepts regarding product quality" and "sense of responsibility for product quality." The fact that the four stages of the wheel are connected one to the other with no

<http://hclelectures.blogspot.no/1970/08/demings-1950-lecture-to-japanese.html>



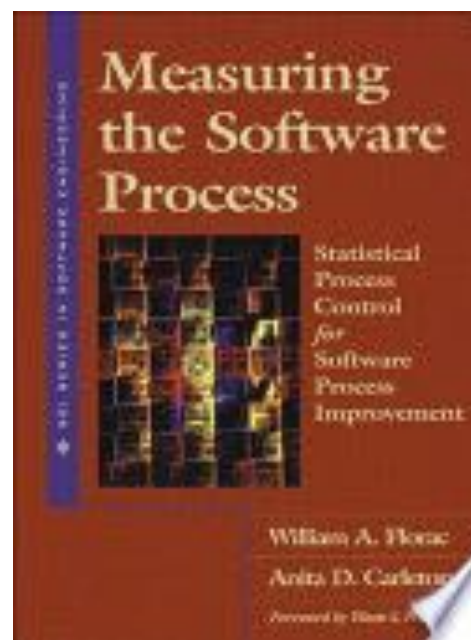
**Figure 1 – Evolution of the Scientific Method and the PDSA Cycle**



Agile is (should be) based on scientific method!

**BUT**

Not merely iteration, but *measurement, learning*  
Like Value Agile



[https://deming.org/uploads/paper/PDSA\\_History\\_Ron\\_Moen.pdf](https://deming.org/uploads/paper/PDSA_History_Ron_Moen.pdf)





The Gilb Evo Cycle  
Our Agile Cycle.  
© [gilb.com](http://gilb.com)



# Chapter 1

## Four Agile Manifesto Values

- \* These are
  - \* too **vague**
  - \* and too **simplistic**
  - \* for my taste
- \* About the level of
  - \* 'America First'
  - \* 'Make America Great Again'

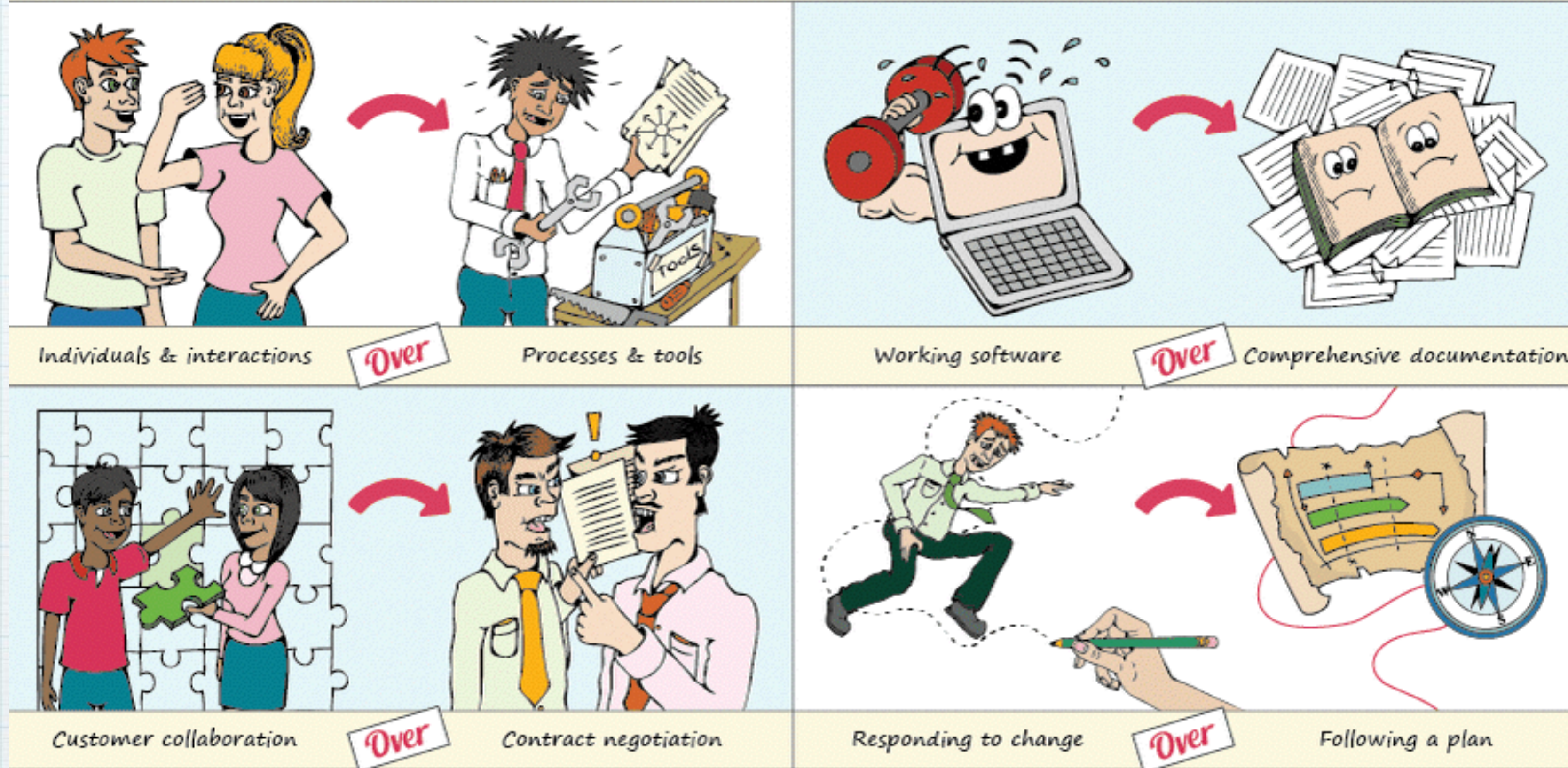
AMERICA FIRST



# Manifesto for Agile Software Development\*

"We are uncovering better ways of developing software by doing it and helping others do it.

Through this work **we have come to value:**



Knowledge  
TRAIN

That is, while there is value in the items on the right, we value the items on the left more."

Copyright © 2018 Knowledge Train Limited. \*Quoted from [www.agilemanifesto.org](http://www.agilemanifesto.org)

This 'we value X OVER Y'

Is **dangerous immature oversimplified** methods.

It does not say **why**, and does not say **how** to modify priorities

[gilb.com/DL60and](http://gilb.com/DL60and) see Presenter Note for more

President Woodrow Wilson



# Value 1. Individuals and interactions over processes and tools

\* Well, of course. 'Live human reality' beats 'theory and planning'.

\* But I prefer,

- \* **'stakeholders first'** and
- \* **Stakeholder** 'interactions with requirements and systems',
- \* before bureaucracy, like 'theory and planning'.

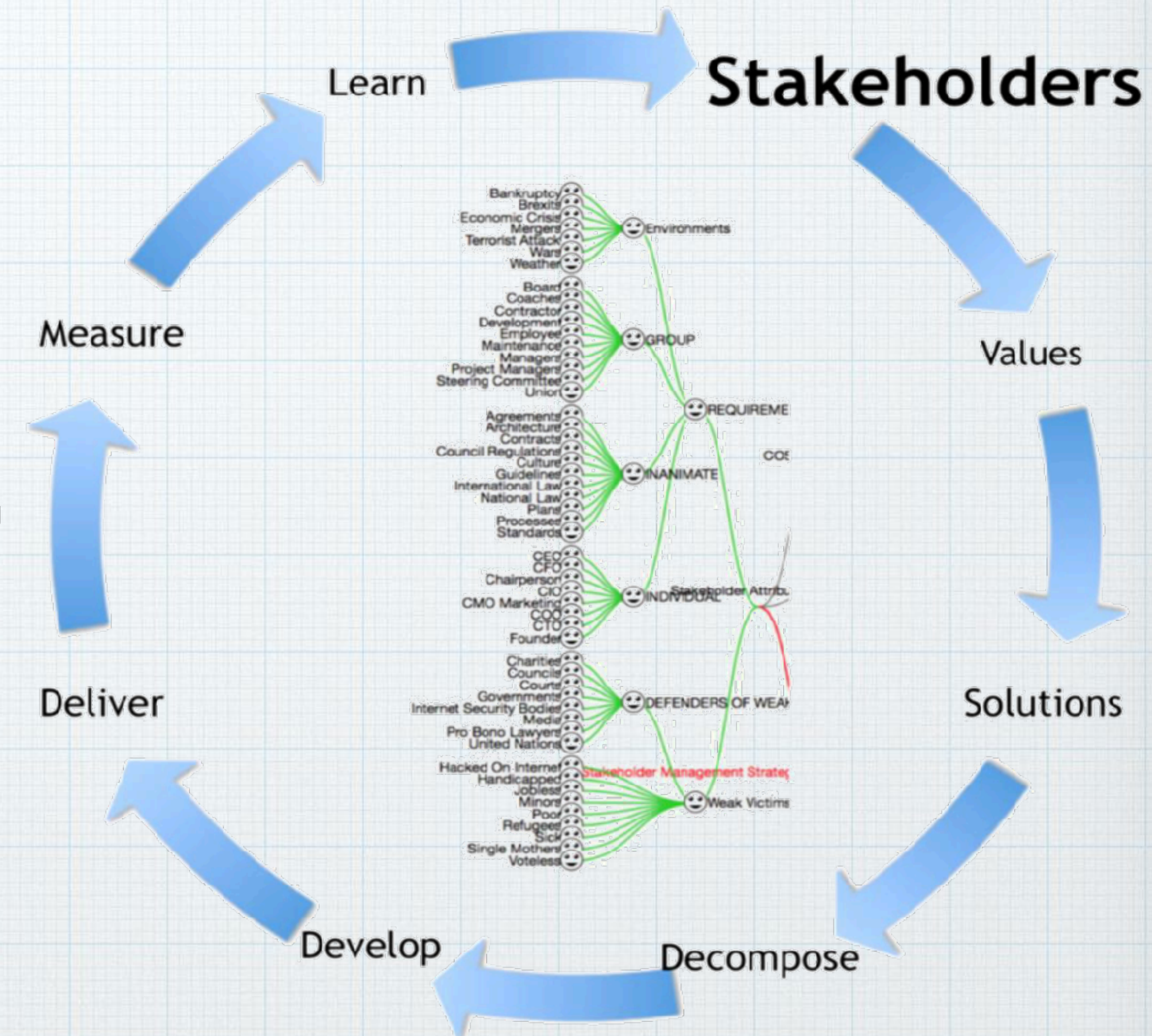
\* Because:

- \* staring at the 'live human reality' of looking computer programs being executed,
- \* when it is the *wrong code design*,
  - \* because of the *wrong requirements*,
  - \* because of the *wrong stakeholders*
- \* Is not a useful view of reality. It is the *wrong reality*.

\* Professionals have to be taught suitable processes to support *stakeholders*,

\* and the Manifesto hardly mentions 'stakeholders':

\* in the Manifesto we see only the narrow category 'users and customers' dominates



The Gilb Evo Cycle

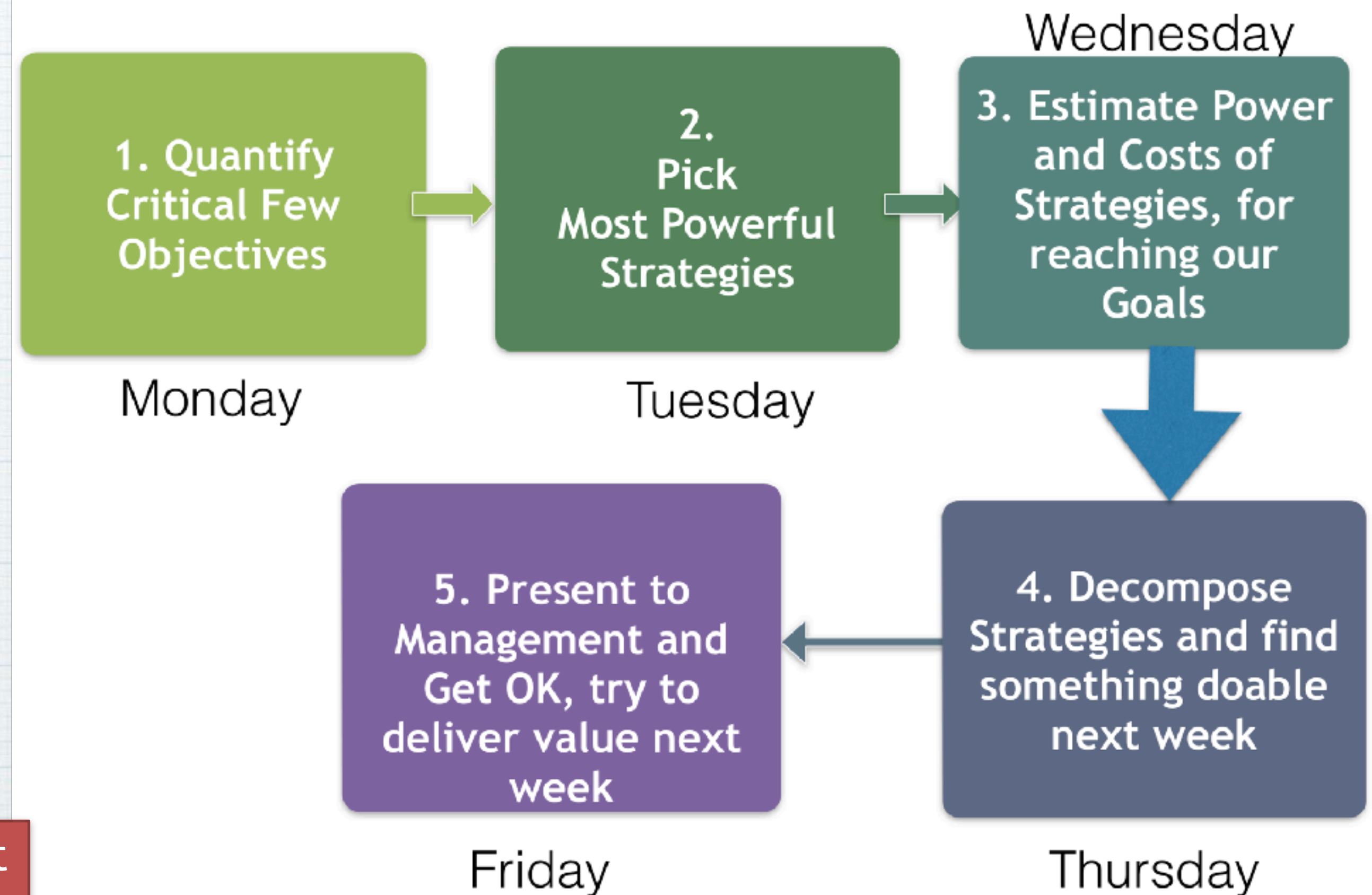


## Value 2. **Working software** over comprehensive documentation

- \* Of course we do not want Waterfall 'comprehensive documentation'
- \* Not 500 to 50,000 pages
- \* But we do need to think about clear requirements and design for a week before doing coding sprints
- \* How about 5 x 1 page specs
  - \* Stakeholders, requirements, designs, decompositions, Value Tables

See much more detail about This agile project startup. -> Process in Part 4 of these Slides

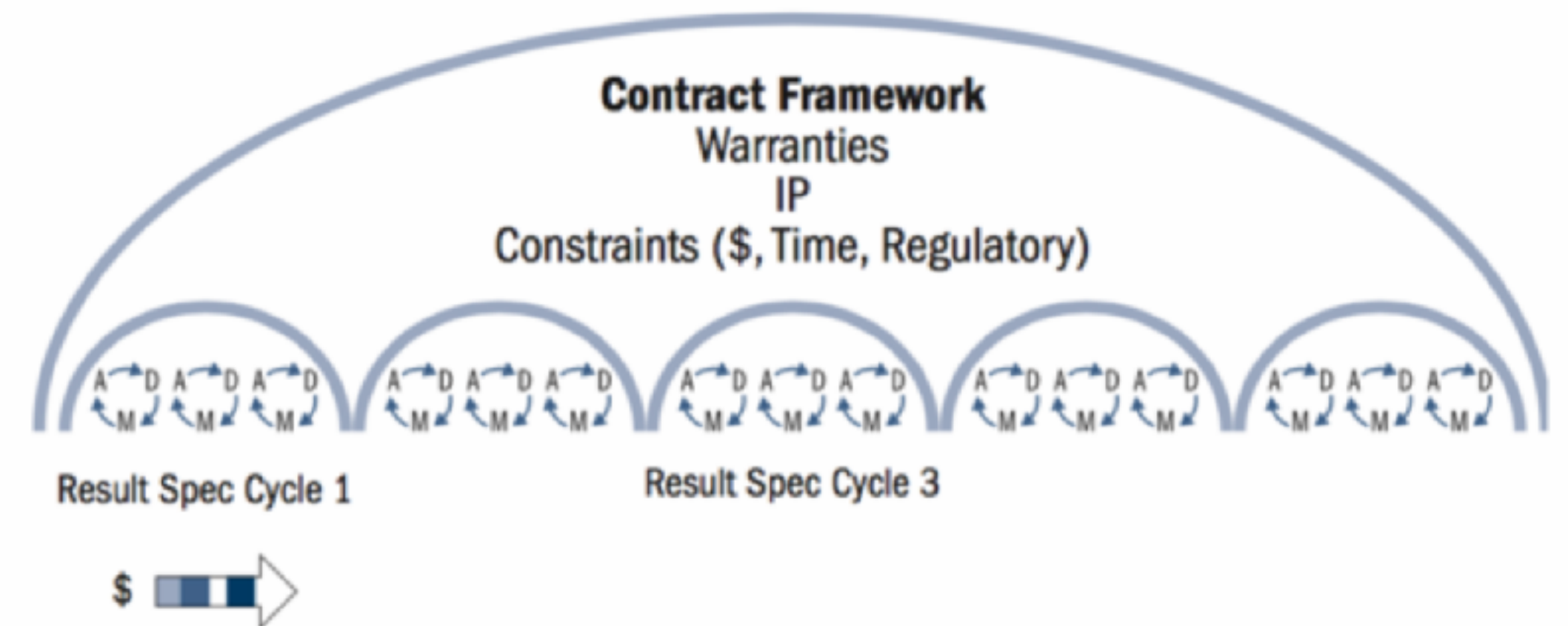
### Evo Startup Week: Formal Process





## Value 3 A. Customer collaboration \* over contract negotiation

- \* I believe this 'Manifesto Value 3' notion, was prompted by inadequate USA/DoD contracting practices,
  - \* compounded by even worse development processes: waterfall, fixed price, and fixed dates,
  - \* with contracted 'technical design' specifications,
  - \* instead of contract results specifications.
- \* Some professional friends of mine have built
  - \* a simple legal framework for doing agile.
  - \* There is no fixed long-term cost, or specs, or deadline.
  - \* [flexiblecontracts.com](http://flexiblecontracts.com)
- \* It is all worked out in 'collaboration with the customer' step by step.
- \* If step results are measurably delivered, payment is due.
- \* 'Negotiation' is done step by step, as we learn, get results, and build confidence.



We need frequent customer  
Interaction with  
Measured value delivery  
And  
Contracted payment  
For these value results

<http://concepts.gilb.com/dl864> source, Contracting for Value slides

\* 'Collaboration' was what we, occupied by Nazis, countries called 'actively helping the enemy'. Not my first choice of term.



## Value 3 B. Customer collaboration over contract negotiation. A Large scale, long term case of doing this with great success: always

- \* Rather than trying to estimate costs, for high-end qualities for space and military projects, IBM Cleanroom, used 2% (monthly for 4 year project for example).
  - \* IBM *measures value* (like availability level) and *costs* (time to deadline and use of lowest bidder fixed price budget)
  - \* At *each* value delivery cycle
  - \* And the architect (Robert Quinnan, see links below right) acts on bad deviations (low quality, high costs) and he *re-designs the architecture*, or does *tradeoffs* on requirements,
  - \* In order to bring things 'back into balance'.(value within resource constraints)
- \* Make no mistake this is an *engineering* method.
- \* It is identical to my Evo method (*Competitive Engineering*, 2005)
- \* It is value-and-cost quantitatively driven, and is radically different (better) from all of the stuff called agile today (Scrum, SAFe, etc)
- \* Let me retitle this as:
  - \* 'Value Engineering Feedback to meet Fixed Contract Requirements'

### Mills on Design to Cost

- "To meet cost/schedule commitments based on imperfect estimation techniques, a software engineering manager must adopt a manage-and-design-to-cost/schedule process.
- That process requires a continuous and relentless rectification of design objectives with the cost/schedule needed to achieve those objectives."
- in IBM sj 4 80 p.420

[http://trace.tennessee.edu/cgi/viewcontent.cgi?article=1004&context=utk\\_harlan](http://trace.tennessee.edu/cgi/viewcontent.cgi?article=1004&context=utk_harlan)



MILLS AND QUINNAN IBM CLEANROOM CASE  
IN GILB, BCS SPA 'VALUE DESIGN' 2 HOUR COURSE.  
Video URL= [https://www.youtube.com/playlist?list=PLKBhokJ0qd3\\_wlv0j85YhmNfNj8ZJ8M-](https://www.youtube.com/playlist?list=PLKBhokJ0qd3_wlv0j85YhmNfNj8ZJ8M-)  
Slide Location: = <http://concepts.gilb.com/dl972>



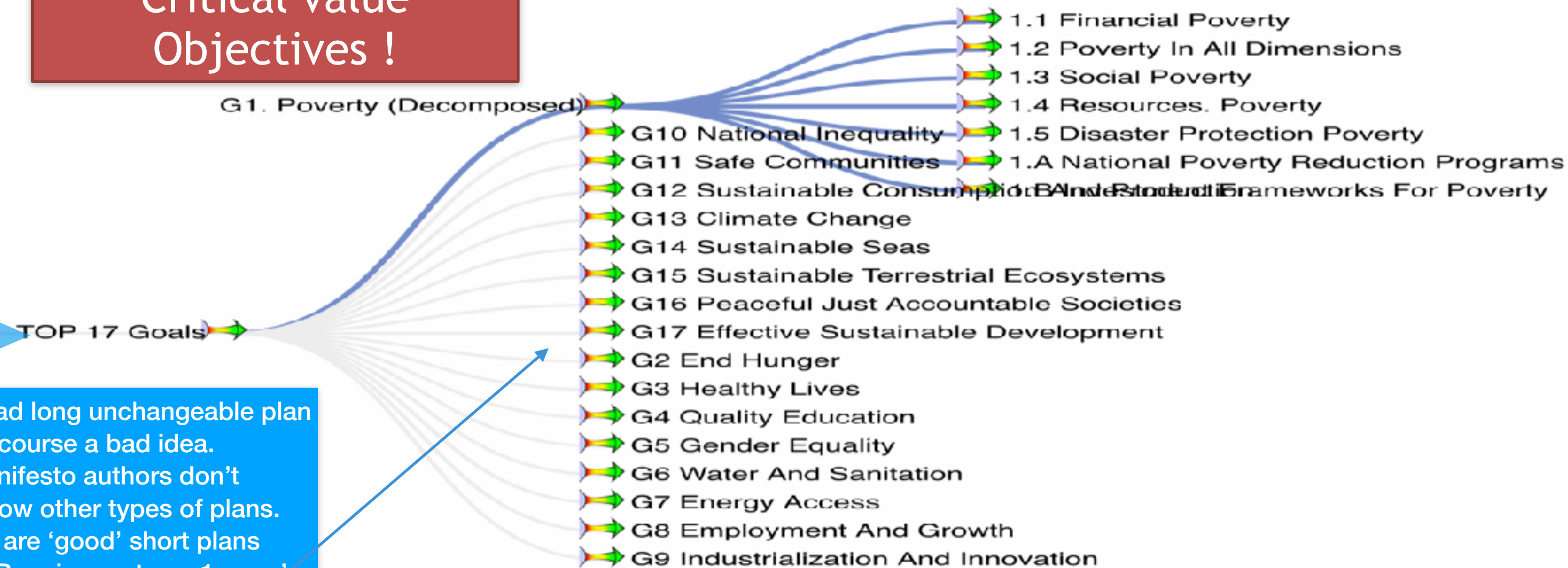
## VALUE 4. 'Responding to change over following a plan'

How can you 'intelligently' respond to change, if you have no plan?

The 'plan' should be to  
Reach your  
Critical value  
Objectives !

- \*Of course, I agree with the above 'platitude', as noted previously. This is the essence of 'agile' ; *responsiveness*.
- \*But, there are several kinds of 'plans', for example:
  - \* immature fixed ones, that are based on lack of deep understanding of complex stakeholder values;
  - \* 'plans which specify badly-designed architecture', rather than 'end results' for stakeholders.
- \*My preference is '
  - \*plans that focus on a few critical, quantified, top-level, long-term value improvements'.
- \*Of course, these quantified plans are subject to incremental change,
  - \*for example, change
    - \*directed by high-level guidance, from top management,
    - \*on behalf of their stakeholders,
    - \*providing good directions of change and improvement.
- \*I believe [1] that we need much better, and much higher level 'plans' [1, 5A],
  - \* and that our responses need to be caused by 'numeric deviation from plans',
  - \* or numeric need to change these numeric plans *to reflect the real world*.
- \*This is both because
  - \*we get to understand that 'real world',
    - \*by trying to deliver change,
  - \*and because the real world itself needs to change top-level requirements
    - \*(business, market, and society changes, for example).
  - \*and thirdly because of
    - \*the necessity of change
    - \*to improved top-level *architectures*
      - \*(technology change).

Following a bad long unchangeable plan is of course a bad idea.  
The Manifesto authors don't seem to know other types of plans.  
But there are 'good' short plans  
Like 'Value Requirements on 1 page'  
Which allow you to respond to change in the resulting value stream cycles



More detail?  
"Sustainability  
Planning"  
Digital Book 2019  
<https://tinyurl.com/UNGGoalsGilb>  
Later see  
Gilb.com

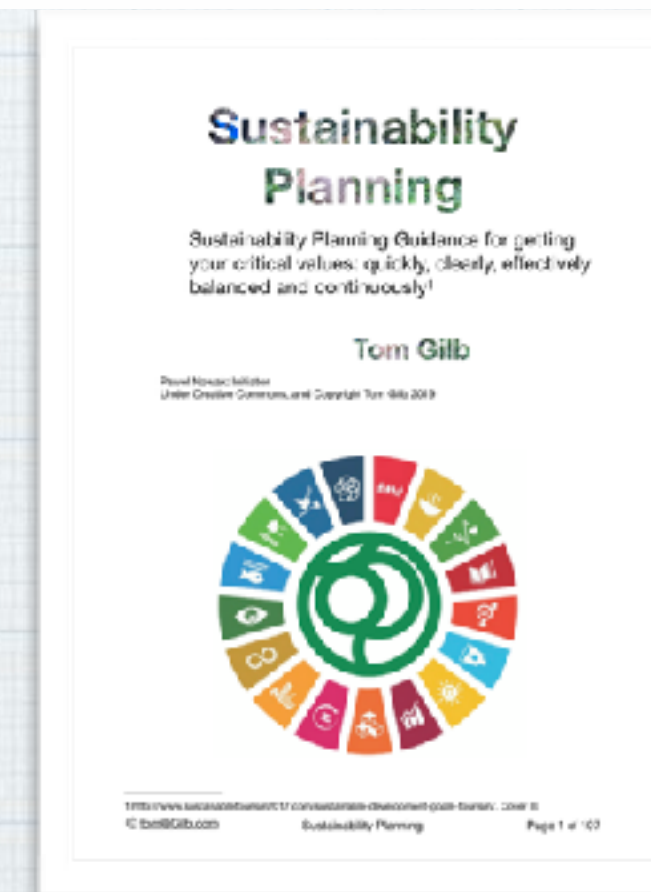


Figure 1.8 A few critical top level long term goals. In this case

the 18 United Nations Sustainability Goals, with some decomposition.

From my book Sustainability Planning, 2019.

(next weeks 2 hours BCS Course).

These 'goals' can be viewed as strategies for reaching the higher level Objective

of a better world



# Chapter 2.

## The Twelve Agile Manifesto Principles

Reference: <http://agilemanifesto.org/principles.html>

I provided my personal counter-proposal for Agile Principles in 2010 <http://www.gilb.com/DL431>

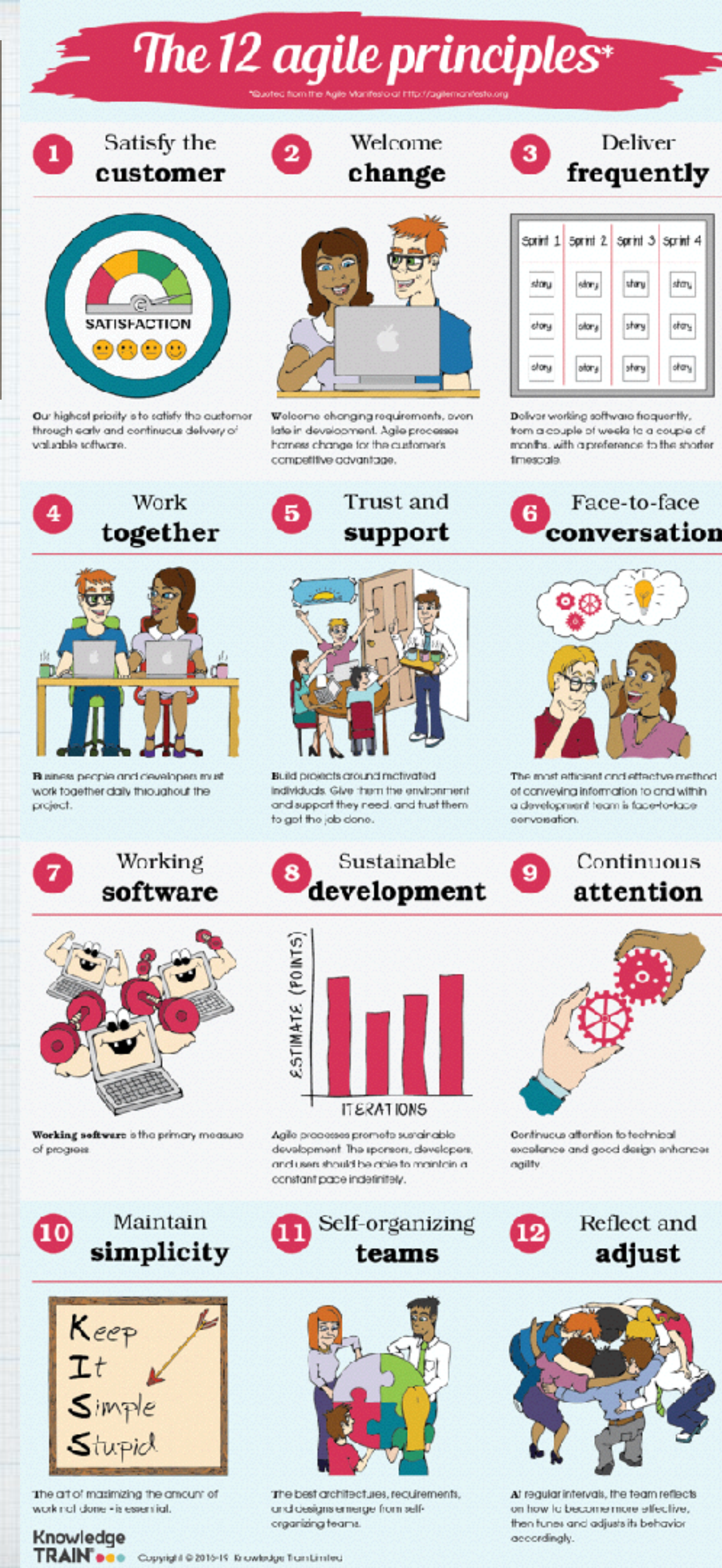
I believe that the 'principles' statements provided in my counter-proposal (and here), are much better, and clearer, than those in the Manifesto.

But then, I *would* say that!

What do YOU think?

I give you now, my direct comments, on the principles as published,

And I am polite, but not nice!

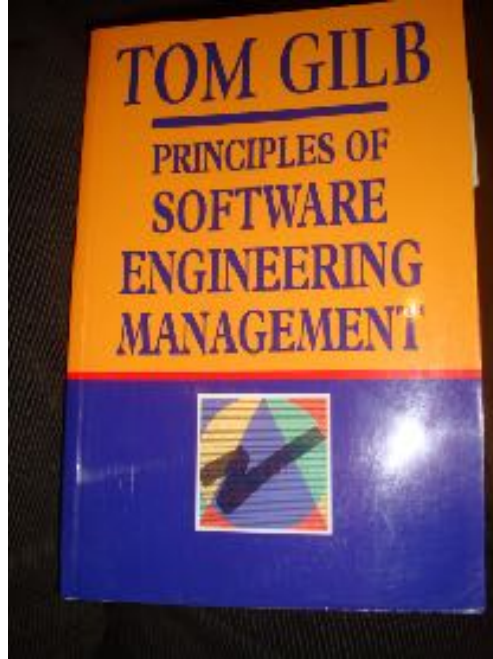




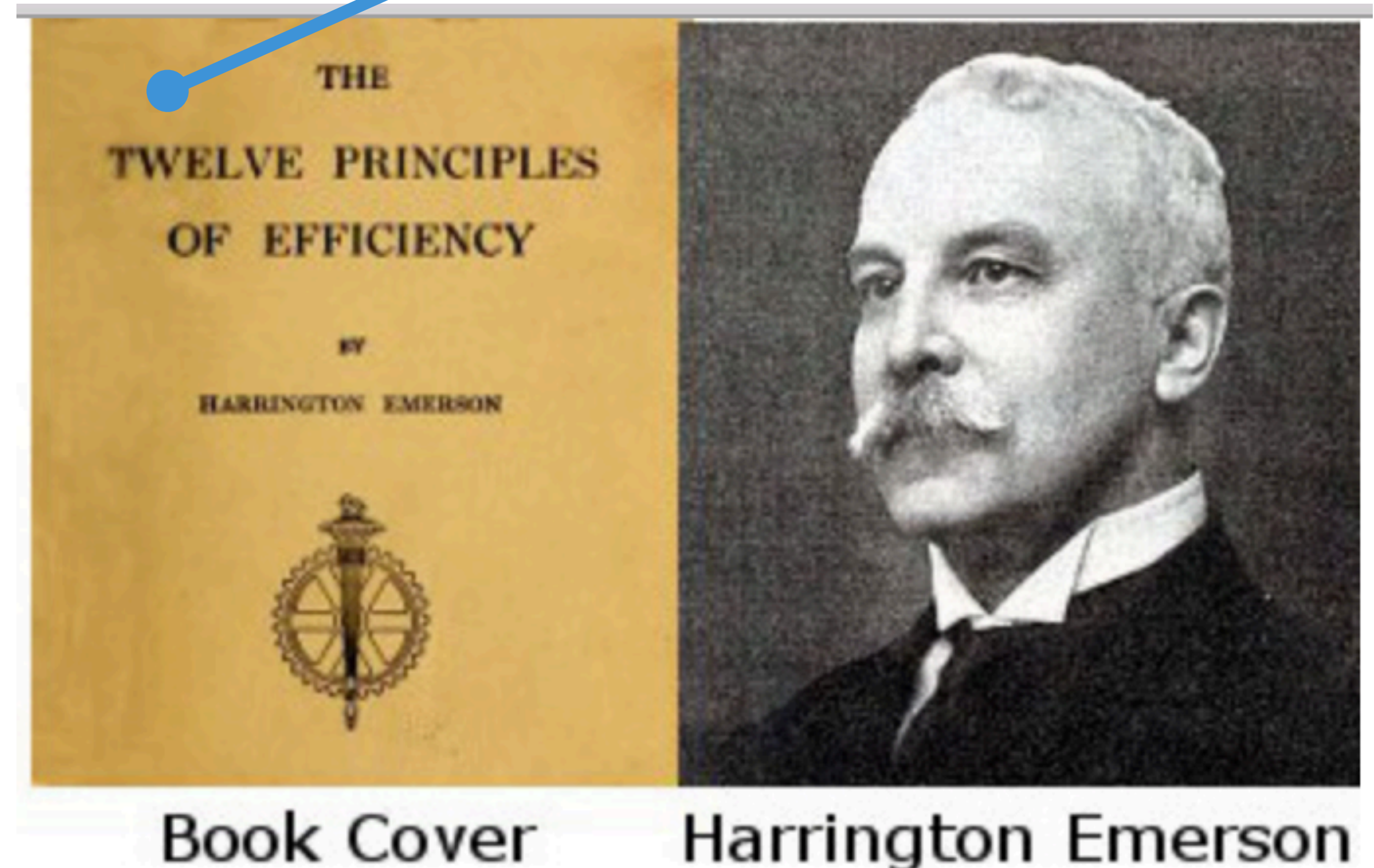
# The Principle that 'Principles beat methods'

144 Principles,  
1988

12 x 12 = 144 :)



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



August 2, 1853 – September 2, 1931

- **Harrington Emerson,**

**Do principles trump methods?**



# PRINCIPLE 1: 'Our highest priority is to satisfy the **customer** through early and continuous delivery of valuable **software**'.

\* Here is my constructive reformulation:

## \* 1. DEVELOPMENT EFFORTS SHOULD

\* ATTEMPT TO DELIVER,

\* MEASURABLY AND

\* COST-EFFECTIVELY,

\* A WELL-DEFINED SET OF

\* PRIORITIZED

\* STAKEHOLDER VALUE-LEVELS,

\* AS EARLY AS POSSIBLE.

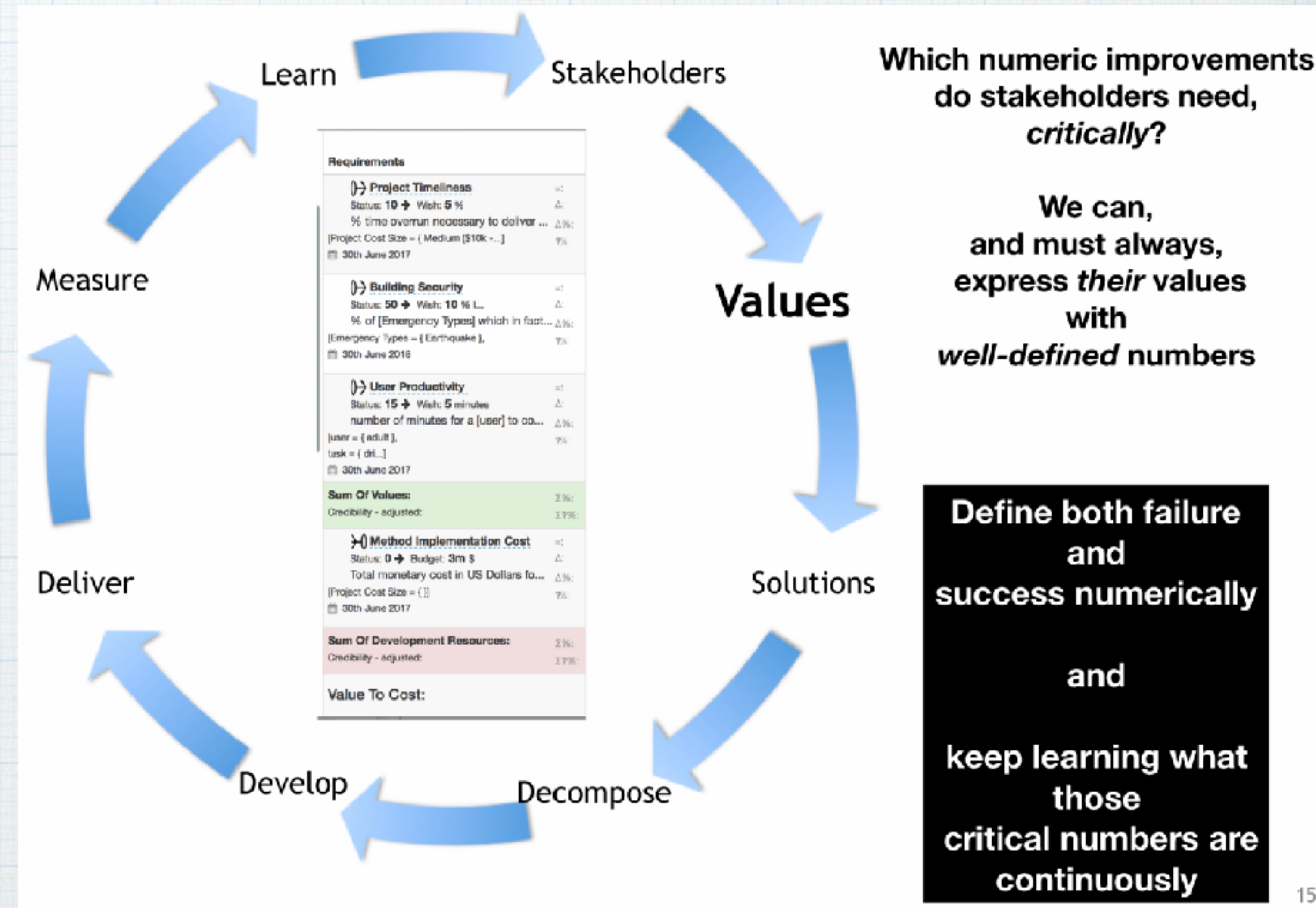


Figure 2.1 The highest priority is delivery of critical stakeholder values, and these values need quantification to understand,

and to manage them. Conventional Agile has totally missed this essential idea.

It even does not seem to recognize that there is more to the world of projects than software.



# PRINCIPLE 2. 'Welcome **changing** requirements, even late in development. Agile processes harness change for the customer's competitive advantage'.

Make the most of changes

Here is my constructive reformulation:

## 2. DEVELOPMENT PROCESSES MUST BE ABLE TO

- \* DISCOVER AND INCORPORATE
- \* CHANGES IN STAKEHOLDER REQUIREMENTS,
- \* AS SOON AS POSSIBLE,
- \* AND TO UNDERSTAND THEIR PRIORITY, THEIR
  - \* CONSEQUENCES
  - \* TO OTHER STAKEHOLDERS,
  - \* TO SYSTEM ARCHITECTURE PLANS,
  - \* AND TO PROJECT PLANS,
  - \* AND CONTRACTS.

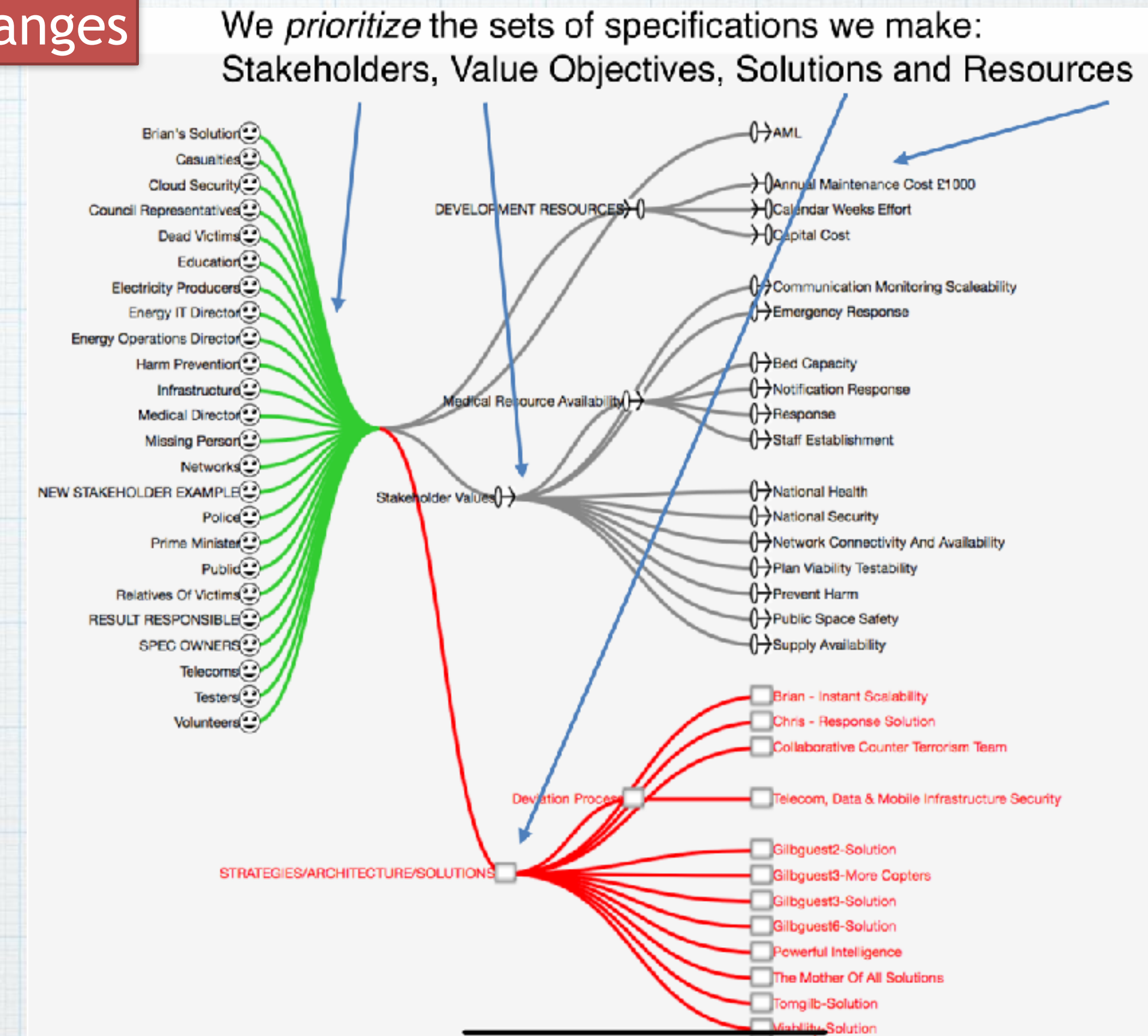


Figure 2.2. . There are many planning components (stakeholders, requirements, designs)

each of which has a partial influence on the priority, the chosen sequence of incremental value delivery.



## PRINCIPLE 3. 'Deliver **working software** frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale'.

\* Here is my constructive reformulation:

\* **3. PLAN TO DELIVER**

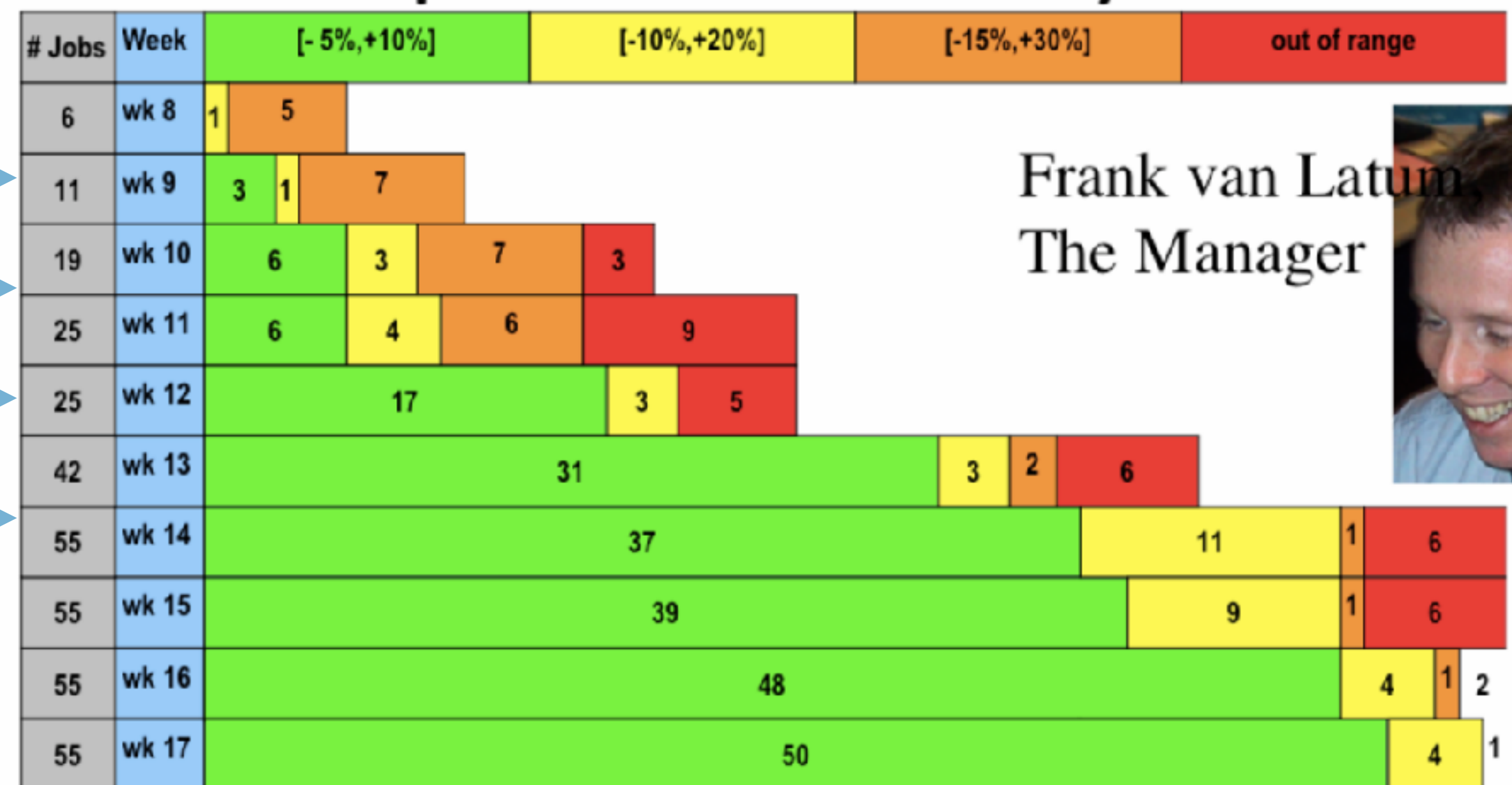
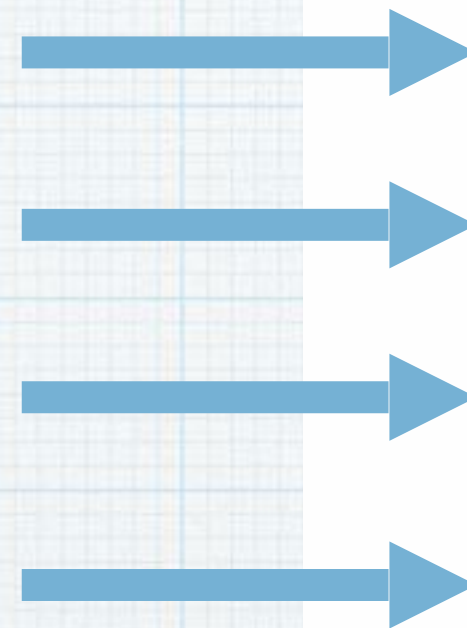
\* **SOME MEASURABLE  
DEGREE OF  
IMPROVEMENT,**

\* **TO PLANNED AND  
PRIORITIZED  
STAKEHOLDER VALUE  
REQUIREMENTS,**

\* **AS SOON,**

\* **AND AS FREQUENTLY,**

\* **AS RESOURCES PERMIT.**



Frank van Latum  
The Manager



Figure 5.6 Philips Value Delivery Cycles Results. The % is the accuracy of predicting a production run of electronic circuits, before that actual run. Green is good, red is bad.

Figure 2.4. One of my clients, Philips, was able to break out of a 'no results' situation

by using my methods of decomposition, to deliver value early and weekly. To cumulate the long term values.

Frank was the hero, the project manager who decided to go with my advice when his director did not believe it could work at all.

He later won applause from the director and his team for the results he could deliver to Philips.



## PRINCIPLE 4.

‘Business people and developers must **work together** daily throughout the project’.

Here is my constructive reformulation:

### 4. ALL PARTIES TO A DEVELOPMENT EFFORT (STAKEHOLDERS),

- \* NEED TO HAVE A RELEVANT VOICE

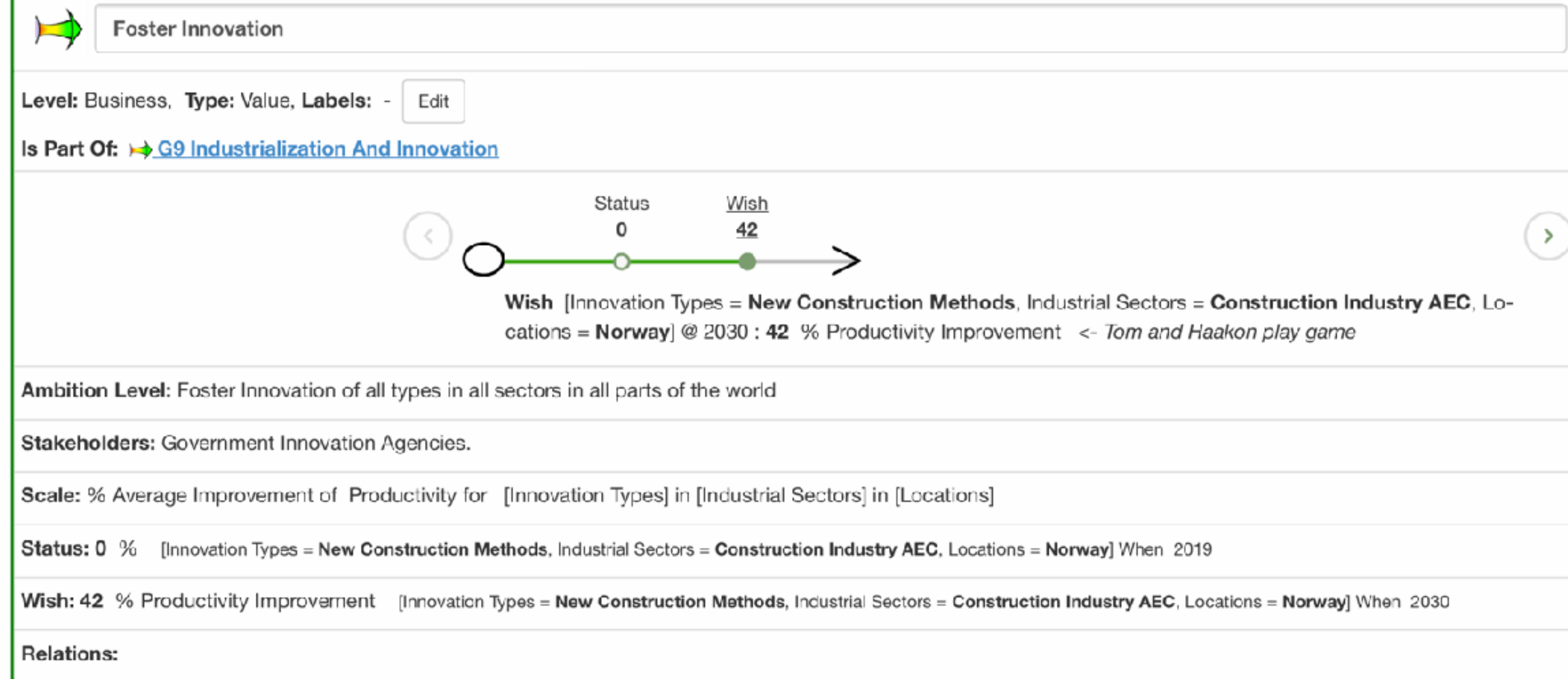
- \* FOR THEIR INTERESTS (REQUIREMENTS),

- \* AND AN INSIGHT INTO THE PARTS OF THE EFFORT THAT THEY WILL POTENTIALLY IMPACT,

- \* OR WHICH CAN IMPACT THEM,

- \* ON A CONTINUOUS BASIS,

- \* INCLUDING INTO OPERATIONS AND DECOMMISSIONING OF A SYSTEM



Communicate  
Clearly  
About Critical  
Values

Figure 2.5. An example of quantifying a value, to ‘Foster Innovation’. The fuzzy source, before quantification and structuring (see the Scale) is in the ‘Ambition Level’ statement.

I am suggesting that this language (Planguage) for communicating, in this case for a ‘value requirement’, is superior to a ‘face to face’ explanation of the requirement.

We can communicate more exact and rich information using this Planguage format. We can update this info from anywhere at anytime. We can link and exploit this information digitally as part of the larger total picture of all requirements, designs, stakeholders. Daily developer-to-business cannot do this at all.



# PRINCIPLE 5. 'Build projects around **motivated** individuals. Give them the environment and support they need, and trust them to get the job done'.

\* Here is my constructive reformulation:

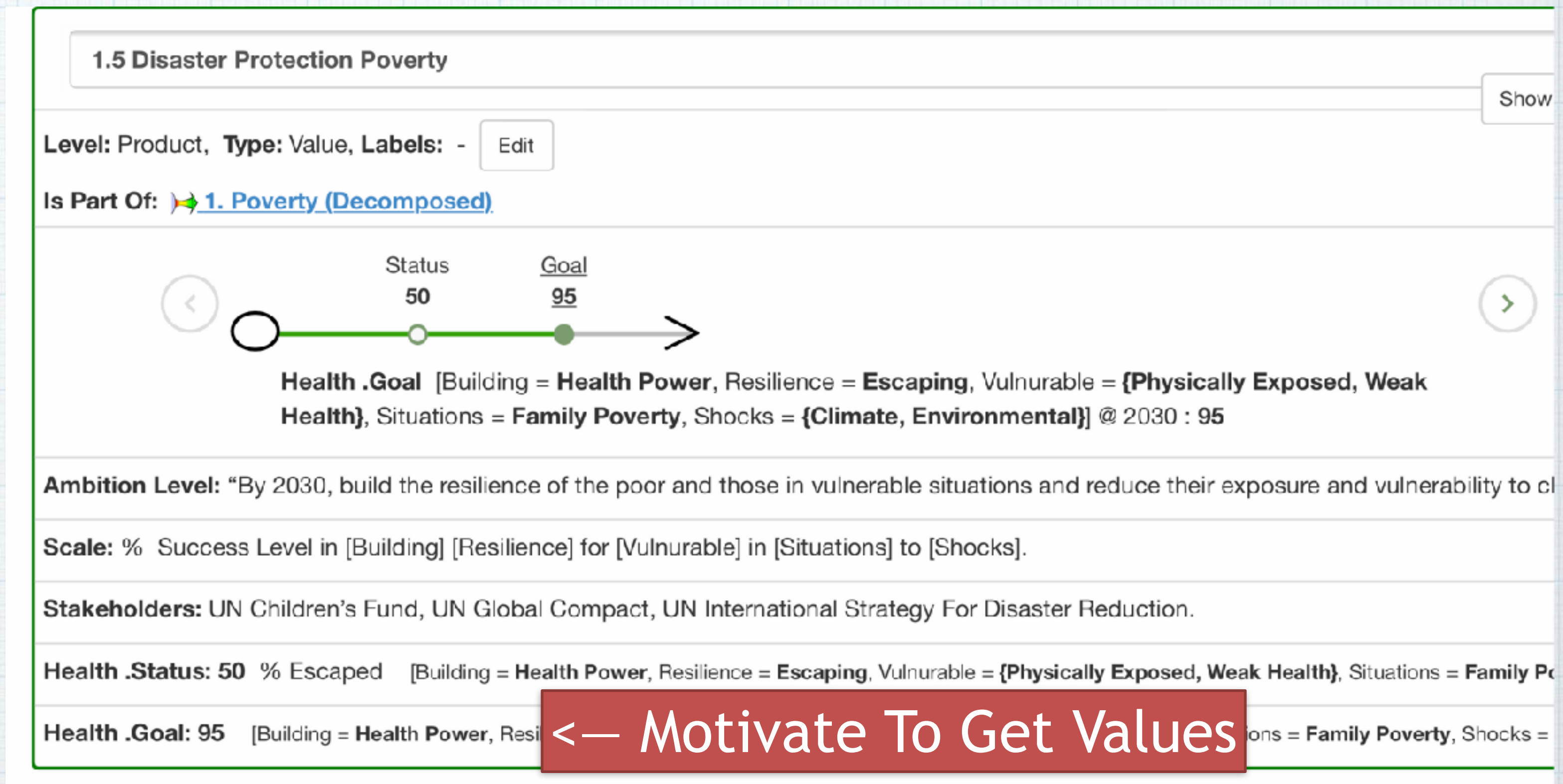
\* **5. MOTIVATE  
STAKEHOLDERS AND  
DEVELOPERS,**

\* **BY**

\* **AGREEING ON THEIR  
CLEAR CRITICAL HIGH-  
LEVEL PRIORITY  
OBJECTIVES,**

\* **AND GIVE THEM FREEDOM  
TO FIND**

\* **THE MOST COST-  
EFFECTIVE SOLUTIONS.**



**FIGURE 2.6. PEOPLE NEED TO BE MOTIVATED IN THE RIGHT DIRECTION: THE SPECIFIC VALUES AND THEIR SPECIFIC LEVELS NEEDED AND THE DEADLINES.**

They need additional motivational elements such as which stakeholders they are serving.

People need to be motivated by detailed, clear, updated, numeric specifications.



## Principle 6. 'Enable face-to-face interactions'.

Here is my constructive reformulation:

### 6. ENABLE CLEAR COMMUNICATION,

\* IN WRITING,

\* IN A COMMON PROJECT DATABASE.

\* ENABLE COLLECTION AND  
PRIORITIZATION,

\* AND CONTINUOUS UPDATES,

\* OF ALL CONSIDERATIONS ABOUT

\* REQUIREMENTS,

\* DESIGNS,

\* ECONOMICS,

\* CONSTRAINTS,

\* RISKS,

\* ISSUES,

\* DEPENDENCIES,

\* AND PRIORITIZATIONS.

Use communication  
Suitable for the  
Complexity

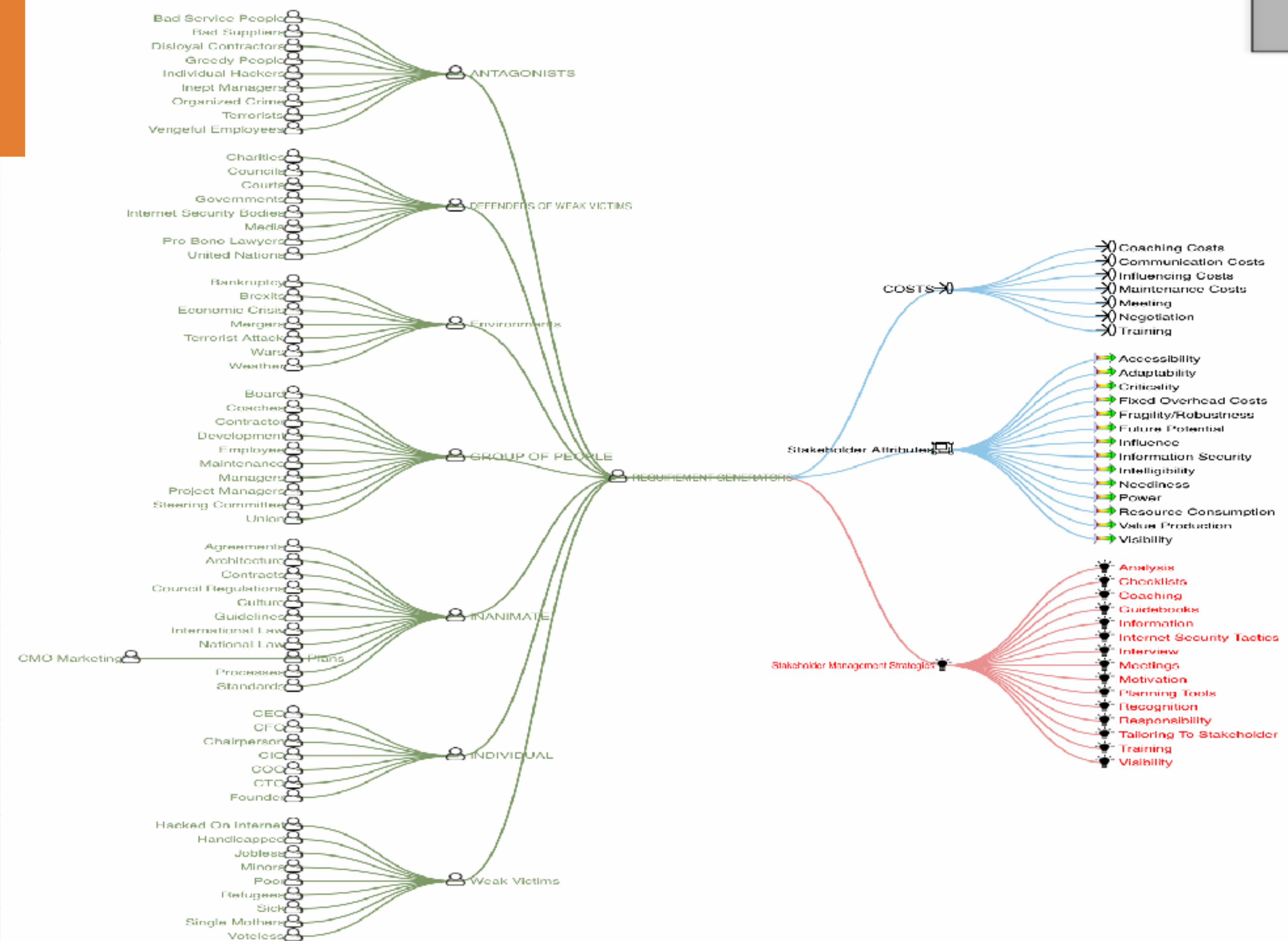


Figure 2.9. This is a summary diagram model over some factors relating to stakeholders. Each individual item might be defined in a page of detail. Maybe 10 or more items of specification for each one item. Every item has many relationships. Now imagine discussing this face to face. But *without* the diagram.



## Principle 7.

*‘Working software is the primary measure of progress’.*

\* Here is my constructive reformulation:

### \* 7. THE PRIMARY MEASURE OF DEVELOPMENT PROGRESS IS

\* THE 'DEGREE OF ACTUAL STAKEHOLDER-DELIVERED PLANNED VALUE LEVELS'

\* WITH RESPECT TO PLANNED RESOURCES,

\* SUCH AS BUDGETS AND DEADLINES.

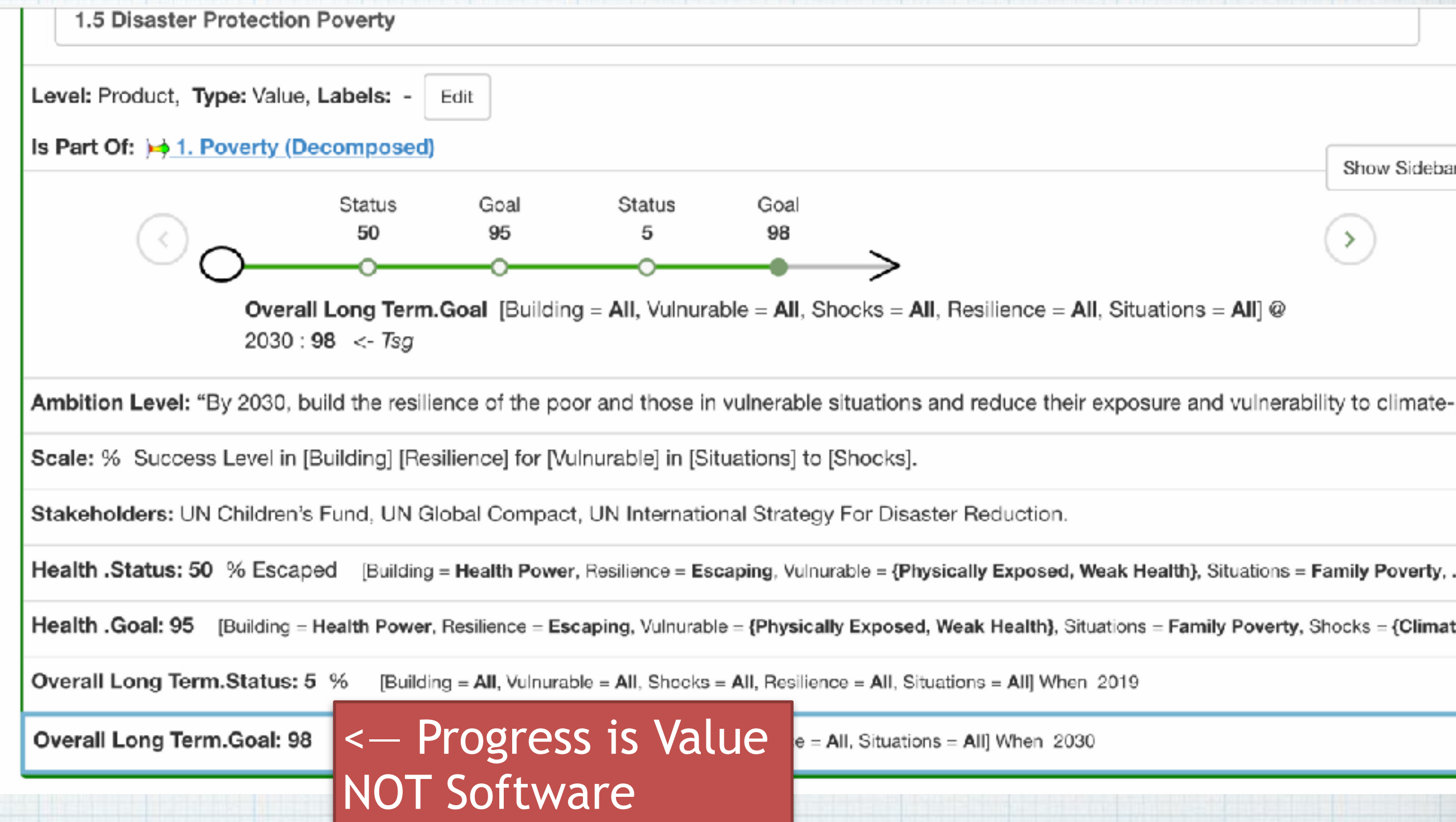


Figure 2.10. Example of a quantified and well-defined objective. This has got nothing to do with 'working software', or 'user stories'. The value is saving poor people from disasters.

Do the poor want a user story, a chunk of software, or a roof over their heads?



# PRINCIPLE 8. 'Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely'.

\* Here is my constructive reformulation:

\* 8. WE BELIEVE THAT

\* A WIDE VARIETY OF STRATEGIES,

\* ADAPTED TO CURRENT LOCAL CULTURES,

\* CAN BE USED TO

\* MAINTAIN A REASONABLE WORKLOAD

\* FOR DEVELOPERS, AND OTHER STAKEHOLDERS;

\* SO THAT STRESS AND PRESSURES,

\* WHICH RESULT IN FAILED SYSTEMS,

\* NEED NOT OCCUR.

Exactly which agile processes?  
And what is the numeric evidence?  
What is *cost-effectiveness*?  
What are *side-effects*, if any?

Constant pace  
Is nice  
But not critical  
For other stakeholders

## IBM MN & NC DP Experience

### • 2162 DPP Actions implemented

- between Dec. 91 and May 1993 (30 months) <- Kan
- RTP about 182 per year for 200 people. <- Mays 1995
  - 1822 suggested ten years (85-94)
  - 175 test related
- RTP 227 person org <- Mays slides
  - 130 actions (@ 0.5 work-years)
  - 34 causal analysis meetings @ 0.2 work-years
  - 19 action team meetings @ 0.1 work-years
  - Kickoff meeting @ 0.1 work-years
  - TOTAL costs 1% of org. resources

- ROI DPP 10:1 to 13:1, internal 2:1 to 3:1
- Defect Rates at all stages 50% lower with DPP



Figure 2.11. The Defect Prevention Process (DPP) is an agile method, for long term improvement of the product development process. This reduces stress and pressures. What I love about this method is that it is driven by grass roots insights, not directors or external consultants. And it works in the long term, measurably. Software Inspection, 1993

It is NOT based on a Manifesto declaration without evidence, or consideration of other methods!



## PRINCIPLE 9.

**'Continuous attention to technical excellence and good design enhances agility'.**

Here is my constructive reformulation:

### 9. 'TECHNICAL EXCELLENCE'

IN PRODUCTS, SERVICES, SYSTEMS AND ORGANIZATIONS,  
CAN AND SHOULD BE QUANTIFIED, FOR ANY SERIOUS  
DISCUSSION OR APPLICATION.

THE SUGGESTED STRATEGIES OR ARCHITECTURES,  
FOR REACHING THESE 'QUANTIFIED EXCELLENCE  
REQUIREMENTS',

SHOULD BE ESTIMATED, USING VALUE DECISION TABLES  
AND THEN MEASURED  
IN EARLY SMALL INCREMENTAL DELIVERY STEPS.

To be even more specific:

- \* 9. REQUIRED QUALITIES MUST BE QUANTIFIED,
- \* AND SUPPORTING DESIGNS FOR QUALITIES
  - \* MUST BE ESTIMATED
  - \* AND MEASURED.

	💡 DATABASE DESIGN	💡 Symbology	💡 Language Script	💡 COPYRIGHT REVIEW...	💡 Ape Apple Usability	Sum	Show Sideba
<b>Requirements</b>							
(→) Usability Status: 0 → Wish: 95 % ... % of [Users] able to complete a [De... [Users = All, ...] 18th June 2022	???? ± 0 0 % of ... 0 ± 0 % (x 0.0) 0 % (x 0.0) ???	65 ± 25 65 % of ... 68 ± 26 % (x 0.1) 7 % (x 0.1) 68%	70 ± 25 70 % of ... 74 ± 26 % (x 0.1) 7 % (x 0.1) 74%	-2 ± 2 -2 % of ... -2 ± 2 % (x 0.3) -7 % (x 0.3) -2%	65 ± 25 65 % of ... 68 ± 26 % (x 0.1) 7 % (x 0.1) 68%	26%: 208 ± 80 %	
(→) Copyright Law Compliance: Status: 0 → Wish: 100 % ... % [National Copyright Legislation] ... [National Copyright Legislation = 16th July 2019	15 ± 5 15 % [N... 15 ± 5 % (x 0.7) 15%	22 ± 0 22 % [N... 22 ± 0 % (x 0.0) 0 % (x 0.0) 22%	2 ± 0 2 % [N... 2 ± 0 % (x 0.0) 0 % (x 0.0) 2%	95 ± 20 95 % [N... 95 ± 20 % (x 0.3) 29 % (x 0.3) 95%	-30 ± 10 -30 % [N... -30 ± 10 % (x 0.1) -3 % (x 0.1) -30%	26%: 104 ± 35 %	
(→) Technical Debt Management Status: 0 → Wish: 50 Min... % [Cost] spent on [Technical Mainten... [Cost = Total Technical Budge...] 22nd June 2016	100 ± 15 100 Minima... 200 ± 30 % (x 0.6) 200%	60 ± 10 60 Minima... 120 ± 20 % (x 0.8) 96 % (x 0.8) 120%	10 ± 1 10 Minima... 20 ± 2 % (x 1.0) 20 % (x 1.0) 20%	0 ± 20 0 Minima... 0 ± 40 % (x 0.2) 0 % (x 0.2) 0%	55 ± 10 55 Minima... 110 ± 20 % (x 0.2) 0 % (x 0.2) 110%	26%: 450 ± 112 %	
(→) Database Security Status: 0 → Wish: 95 Hac... % of [Unauthorised Access] to [Syst... [Unauthorised Access = All 1st June 2022	90 ± 50 90 Hackers 95 ± 53 % (x 0.2) 19 % (x 0.2) 95%	40 ± 0 40 Hackers 42 ± 0 % (x 0.0) 0 % (x 0.0) 42%	45 ± 20 45 Hackers 47 ± 21 % (x 0.3) 14 % (x 0.3) 47%	10 ± 0 10 Hackers 11 ± 0 % (x 0.4) 4 % (x 0.4) 11%	65 ± 10 65 Hackers 68 ± 11 % (x 0.0) 0 % (x 0.0) 68%	26%: 263 ± 85 %	
<b>Sum Of Values:</b>	Σ%: 310 ± 88 % (x 310)	252 ± 48 % (x 562)	143 ± 49 % (x 705)	104 ± 62 % (x 809)	216 ± 67 % (x 1025)		
<b>Worst Case:</b>	Σ±%: 222 %	206 %	94 %	42 %	149 %		
<b>Credibility adjusted:</b>	Σ7%: 150 %	103 %	42 %	32 %	4 %		
<b>Worst Case Cred. - adjusted:</b>	Σ±7%: 117 %	84 %	31 %	18 %	0 %		

Figure 2.13. This Value Decision Table illustrates a serious engineering and scientific approach to 'good design' (a set of designs which meet the quantified goals and constraints).

This numeric multidimensional approach is not directly related to agility. It can be used in both agile and waterfall projects. However, as I illustrate in depth, this table can be used to systematically decompose both objectives and designs, so that they can be intelligently prioritized, into value delivery sprints. Agile, 'as it should be'. But this is not in any way envisaged by the Manifesto principles.

Each column (strategies) and each row (objectives) and every combination of them are potential opportunities for a sprint: an implementation of a partial strategy in order to deliver a degree of one value, at least.



## PRINCIPLE 10.

‘Simplicity - the art of maximizing the amount of **work not done** -  
is essential’.

More essential than what?  
<— Delivering values to stakeholders?

\* Here is my constructive reformulation:

\* **10. WE NEED TO**

\* **LEARN AND APPLY**

\* **A WIDE VARIETY OF  
RELEVANT METHODS,**

\* **OF WHICH THERE ARE MANY  
AVAILABLE,**

\* **TO HELP US UNDERSTAND  
COMPLEX SYSTEMS**

\* **AND COMPLEX RELATIONS.  
AND TO SUCCEED IN MEETING**

\* **OUR GOALS**

\* **IN SPITE OF COMPLEXITY.**

### TEN PRINCIPLES OF SIMPLIFICATION: ‘MAKING COMPLEXITY UNCOMPLICATED’

COPYRIGHT [TOM@GILB.COM](mailto:TOM@GILB.COM) 2015-2017

**Lord Kelvin’s Principle 1:** If you *quantify* a variable attribute, it becomes more intelligible.[8]

**Lord Kelvin’s Principle 2:** If you *measure* a variable attribute the system becomes more intelligible. [8]

**The Deming Feedback Principle:** PDSA [10]: If you compare attribute measurements with earlier estimations of them, you will get more understanding of a complex system.

**The Cartesian Decomposition Principle:** If you decompose a high-level generic attribute (like Usability, Maintainability, Security [CE5]) into a set of sub-attributes, you will get a tool for understanding the system, and understanding some critical aspects of it.

**Santayana’s Learning from History Principle:** If you attempt to determine ‘benchmark’ points (such as Past levels, Records and Trends [CE, VP]) on critical attribute scales of measure, you will get, and can share, basic insights about a complex system, at a level *above* the complexity that generates the benchmark levels.

**6. The Scientific Experiment Principle:** If you build or modify any system, using new design strategies, one small incremental step at a time; you can get an understanding of the multiple performance values’ contributions, and multiple costs, that are due to that particular strategy, using incremental measurement [Evo].

**7. The Side Effects Principle:** if you consciously model, estimate and measure the side-effects of your individual designs and strategies, for example using an Impact Estimation table [CE, VP] you will better understand the bottom-line effects of a complex set of system components, interactions, and their architecture.

**8. The Small Change Principle:** If you undertake to measure incremental effects of any change to a system, early, frequently, and with small doses, then you will gain a better understanding of the system component’s interactions.

**9. The Who, Where, What and If Principle:** If you decompose your objectives, and your strategy choices, by ‘critically different dimensions’ of people, tasks, places and events; then you can much better understand the behavior of the system, under these specified conditions. (Scale parameters, VP 1.9)

**10. The Formal Model Principle:** if you take the effort to build a multi-dimensional and multi-level quantified relationship model of your system, using Planguage, for example and a tool [9], you have a much better chance to understand aspects of the system, on an as-needed basis.

Figure 2.14 My suggested simplification principles

‘Making Complications Simple: using Planguage’ <http://www.gilb.com/dl854>.

CE and VP, in the Principles above, are my book references,

Competitive Engineering, Value Planning., Technoscopes (2018)



Best Arch. = ??

# Principle 11.

‘The **best** architectures, requirements, and designs emerge from **self-organizing teams**’.

\* Here is my constructive reformulation:

\* **11A. THE MOST USEFUL VALUE AND QUALITY REQUIREMENTS WILL BE QUANTIFIED,**

\* **AND WILL USE OTHER MECHANISMS,**

\* **INCLUDING CAREFUL CORRESPONDING-STAKEHOLDER ANALYSIS [1, 51, AND 52],**

\* **TO FACILITATE UNDERSTANDING.**

\* **11B. THE MOST COST-EFFECTIVE DESIGNS/ARCHITECTURE, WITH RESPECT TO OUR QUANTIFIED VALUE-AND-RESOURCE REQUIREMENTS, WILL BE**

\* **ESTIMATED, AND PROGRESS TRACKED,**

\* **UTILIZING A VALUE DECISION TABLE,**

\* **WITH ITS EVIDENCE, SOURCES, AND UNCERTAINTY.**

\* **THEY WILL BE PRIORITIZED BY VALUES/RESOURCES**

\* **WITH RESPECT TO RISKS [45].**

\* Simplified:

\* **11. WE WILL USE ENGINEERING QUANTIFICATION**

\* **FOR ALL VARIABLE REQUIREMENTS,**

\* **AND FOR ALL ARCHITECTURE.**

Requirements	ProductDesign	Financials	MarketingStrategy	DistributionMethod	Sum
Demographic Past: 0 → Wish: 50 %	20 ± 5 % Δ%: 40 ± 10 %	27 ± 5 % Δ%: 54 ± 10 %	23 ± 3 % Δ%: 46 ± 6 %	10 ± 0 % Δ%: 20 ± 0 %	160 ± 26 %
Millionaire Past: 1 → Wish: 100000 \$	450000 ± 150000 \$ Δ%: 45 ± 15 %	400000 ± 100000 \$ Δ%: 40 ± 10 %	100000 ± 50000 \$ Δ%: 10 ± 5 %	200000 ± 100000 \$ Δ%: 20 ± 10 %	115 ± 40 %
MarketSegment Past: 4 → Wish: 1 Market Rank	1 ± 1 Market Rank Δ%: 100 ± 33 %	4 ± 1 Market Rank Δ%: 0 ± 33 %	2 ± 1 Market Rank Δ%: 67 ± 33 %	3 ± 1 Market Rank Δ%: 33 ± 33 %	200 ± 132 %
Geography Past: → Wish: 100 %	5 ± 5 % Δ%: 5 ± 5 %	10 ± 4 % Δ%: 10 ± 4 %	40 ± 5 % Δ%: 40 ± 5 %	30 ± 5 % Δ%: 30 ± 5 %	85 ± 19 %
Market Past: 0 → Wish: 100 %	40 ± 10 % Δ%: 40 ± 10 %	5 ± 3 % Δ%: 5 ± 3 %	40 ± 10 % Δ%: 40 ± 10 %	20 ± 5 % Δ%: 20 ± 5 %	105 ± 28 %
Sum Of Performance:	Σ%: 230 ± 73 %	Σ%: 109 ± 60 %	Σ%: 203 ± 59 %	Σ%: 123 ± 53 %	
TimeToMarket Past: 1 → Wish: 8 Weeks	2 ± 0.5 Weeks Δ%: 14 ± 7 %	2 ± 0.5 Weeks Δ%: 14 ± 7 %	3 ± 0.75 Weeks Δ%: 29 ± 11 %	4 ± 1 Weeks Δ%: 43 ± 14 %	100 ± 39 %
ShowMeTheMoney Past: 0 → Wish: 5005 £	1200 ± 200 £ Δ%: 24 ± 4 %	205 ± 200 £ Δ%: 4 ± 4 %	2100 ± 500 £ Δ%: 42 ± 10 %	1500 ± 0 £ Δ%: 30 ± 0 %	100 ± 16 %
Sum Of Resources:	Σ%: 38 ± 11 %	Σ%: 18 ± 11 %	Σ%: 71 ± 21 %	Σ%: 73 ± 14 %	
Performance To Cost:	6.05	6.06	2.86	1.68	
Ratio (Worst Case)	3.20	1.69	1.57	0.60	
	157/49 = 3.2				

Figure 2.15. A simplified example of quantification of requirements and design impacts.

Including values to cost ratios, and worst case understanding



Is this a pure mental internal process, or are ideas tried out?

## Principle 12. 'At regular intervals, the team **reflects** on how to become more **effective**, then tunes and adjusts its behavior accordingly'

Clear measurable idea for them: 'Effective'?

\* Here is my constructive reformulation:

### \* 12. A PROCESS

\* *LIKE THE DEFECT PREVENTION PROCESS (DPP),*

\* *OR ANOTHER MORE-SUITABLE FOR CURRENT CULTURE, WHICH DELEGATES POWER TO*

\* *ANALYZE AND CURE ORGANIZATIONAL WEAKNESSES,*

\* *WILL BE APPLIED:*

\* *USING PARTICIPATION FROM SMALL SELF-ORGANIZED TEAMS*

\* *TO DEFINE AND PROVE*

\* *MORE COST-EFFECTIVE WORK ENVIRONMENTS, TOOLS, METHODS, AND PROCESSES.*

\* *SIMPLER SHORTER REFORMULATION*

\* **12. INCREMENTALLY ENGINEER ORGANIZATIONAL EFFECTIVENESS**

Here is one example of getting a lot more specific

## My 10 Principles of Improvement

### Work Environment

1. Delegate to the doers
2. Measure the improvements
3. Let troops identify common cause defects
4. Let them suggest root causes
5. Let them suggest and try cures

### Product Development

6. Let troops choose the value goal to work on
7. Let them estimate the power of their ideas
8. Let them decide which design to implement
9. Let them measure the results, this week and total to date
10. Credit them for the results, and reward success

Figure 2.16 . Some of my summary personal opinions about improvement based on the case data (particularly DPP and Confront) in the 'Power To The Programmers' slides and my other books and papers. <http://concepts.gilb.com/dl841>



# Re-defining Manifesto Values (the ‘objectives’ of projects)

~~1. Individuals and Interactions Over  
Processes and Tools~~

~~2. Working Software Over  
Comprehensive Documentation.~~

~~3. Customer Collaboration Over  
Contract Negotiation~~

~~4. Responding to Change Over  
Following a Plan.~~

1. Stakeholder Values *first*.

2. Deliver *real measurable*  
*stakeholder values*.

3. Zero failures, to deliver  
values.

4. *Change* the architecture  
*fast*, if it does not deliver  
values.

PS I have, for fun, quantified all *these* ‘manifesto’ values as  
objectives, in Planguage.

Ask me for them if you are interested in going so deep ([tom@gilb.com](mailto:tom@gilb.com)).  
The quantification in Planguage makes it clear that the above is highly  
ambiguous B\*\*\*hit. Billions of possible interpretations. Do you think the  
Manifesto writers had a clear common understanding of these values? (no  
way)



'How Well Does the Agile Manifesto Align with Principles that Lead to Success in Product Development?'

[https://www.ppi-int.com/wp-content/uploads/2018/02/SyEN\\_62.pdf](https://www.ppi-int.com/wp-content/uploads/2018/02/SyEN_62.pdf)



# Re-defining Manifesto Principles (1->5)

## (the 'means' to attain the 'values')



~~1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.~~

~~2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.~~

~~3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.~~

~~4. Business people and developers must work together daily throughout the project.~~

~~5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.~~

1. Development efforts should attempt to deliver, measurably and cost-effectively, a well defined set of prioritized stakeholder value levels, as early as possible.

See 2018 book '100 Practical Project Planning Principles' ([gilb.com](http://gilb.com))

2. Development processes must be able to discover and incorporate changes in stakeholder requirements, as soon as possible, and to understand their priority, their consequences to other stakeholders, to system architecture plans, to project plans, and contracts.

3. Plan to deliver some measurable degree of improvement to planned and prioritized stakeholder value requirements, as soon, and as frequently, as resources permit.

4. All parties to a development effort (stakeholders), need to have a relevant voice for their interests (requirements), and an insight on the parts of the effort that they will potentially impact, or which can impact them, on a continuous basis, including into operations and decommissioning of a system.

5. Motivate stakeholders and developers, by agreeing their high-level priority objectives, and give them freedom to find the most cost-effective solutions.

'How Well Does the Agile Manifesto Align with Principles that Lead to Success in Product Development?'

[https://www.ppi-int.com/wp-content/uploads/2018/02/SyEN\\_62.pdf](https://www.ppi-int.com/wp-content/uploads/2018/02/SyEN_62.pdf)



# Re-defining Manifesto Principles (6->10)

## (the 'means' to attain the 'values')



**6. Enable face-to-face interactions.**

**7. Working software is the primary measure of progress.**

**8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.**

**9. Continuous attention to technical excellence and good design enhances agility.**

**10. Simplicity--the art of maximizing the amount of work not done--is essential.**

6. Enable crystal clear communication, in writing, in a common project database. Enable collection and prioritization and continuous updates of all considerations about requirements, designs, economics, constraints, risks, issues, dependencies and prioritization.

See 2018 Book 'Clear Communication' at [gilb.com](http://gilb.com)

7. The primary measures of development progress is the degree of actual stakeholder-delivered planned value levels with respect to planned resources such as budgets and deadlines.

8. We believe that a wide variety of strategies, adapted to current local cultures, can be used to maintain a reasonable workload for developers and other stakeholders; so that stress and pressures which result in failed systems need not occur.

9. Quality must be quantified, supporting designs for quality must be estimated and measured.

10. We need to learn and apply methods, of which there are many, to help us understand complex systems and complex relations and succeed in meeting our goals in spite of them.

See 2018 'Technoscopes' book at Gilb.com

'How Well Does the Agile Manifesto Align with Principles that Lead to Success in Product Development?'

[https://www.ppi-int.com/wp-content/uploads/2018/02/SyEN\\_62.pdf](https://www.ppi-int.com/wp-content/uploads/2018/02/SyEN_62.pdf)



# PART 3: (Chapter 3 in 'Value Agile' book)

## Principles of Project **Failure**:

### How to **sabotage** a project, without anyone noticing you.

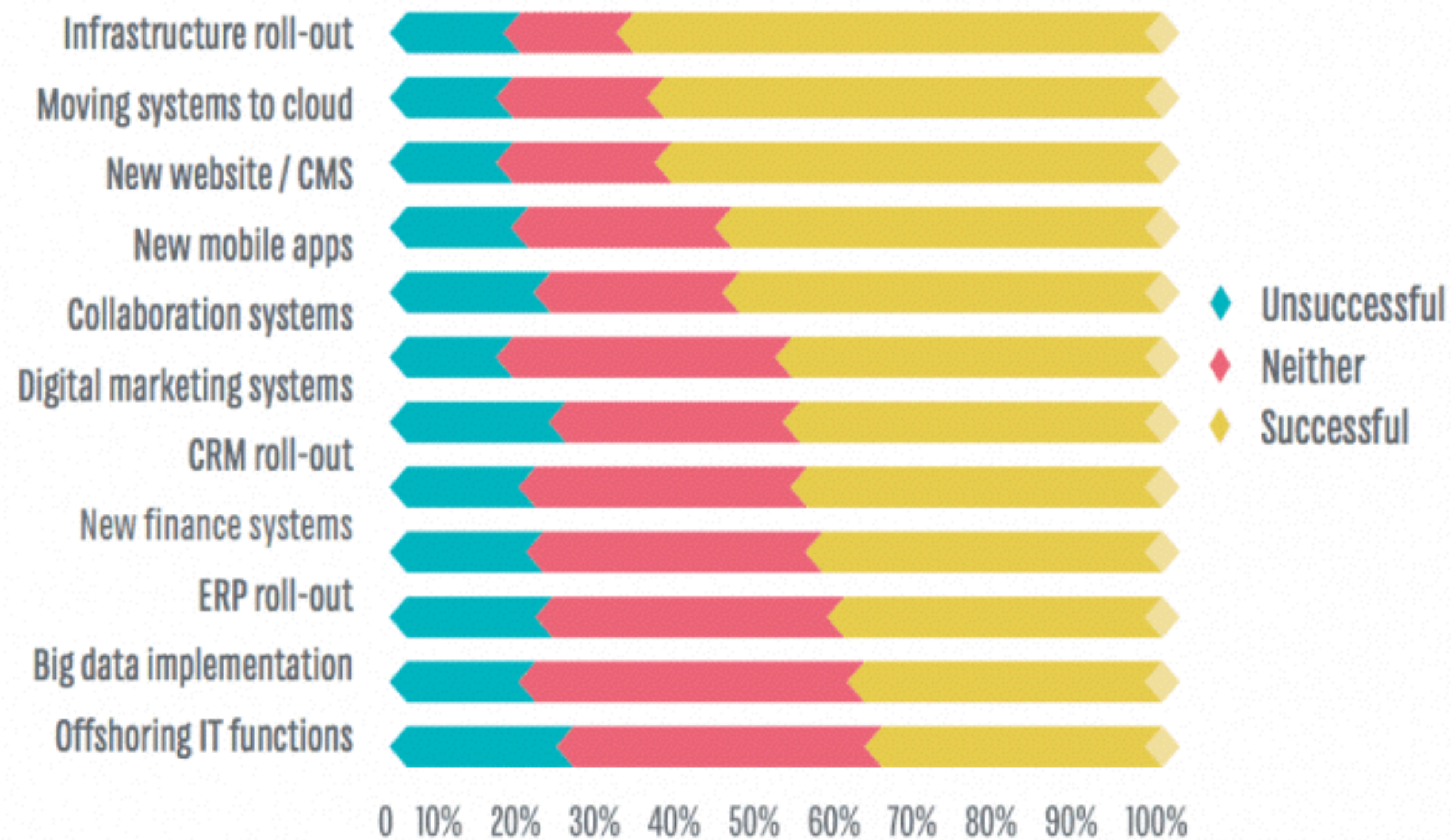


Chart 7: For those projects completed during the last TWO YEARS, indicate how successful you feel the project was.

CHAOS RESOLUTION BY AGILE VERSUS WATERFALL

SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

The resolution of all software projects from FY2011-2015 within the new CHAOS database, segmented by the agile process and waterfall method. The total number of software projects is over 10,000.



# THE MAIN PRINCIPLES OF FAILURE

Part 3.1A. Do not analyze **stakeholders**, stick to 'customers' and 'users'.  
Ignore other voices.

- Bezos is aware of
  - the multiple aspects of customer experience,
  - and the necessity of incrementing these different experiences (*“every important aspect”* →)
  - constantly in a better direction. (*“a little bit better”* →)
- In other words, he is *really* focussed on stakeholders
  - (customers, and *those who provide* the customer experiences, the *party*, the *hosts*),
  - on multiple customer values (*“every important aspect”*),
  - on quantified 'daily' (*“every day”*) increments (agile 'as it should be') towards 'better' (numerically better, I assume).
- So, it is a 'little richer picture', than narrow 'customer focus'.

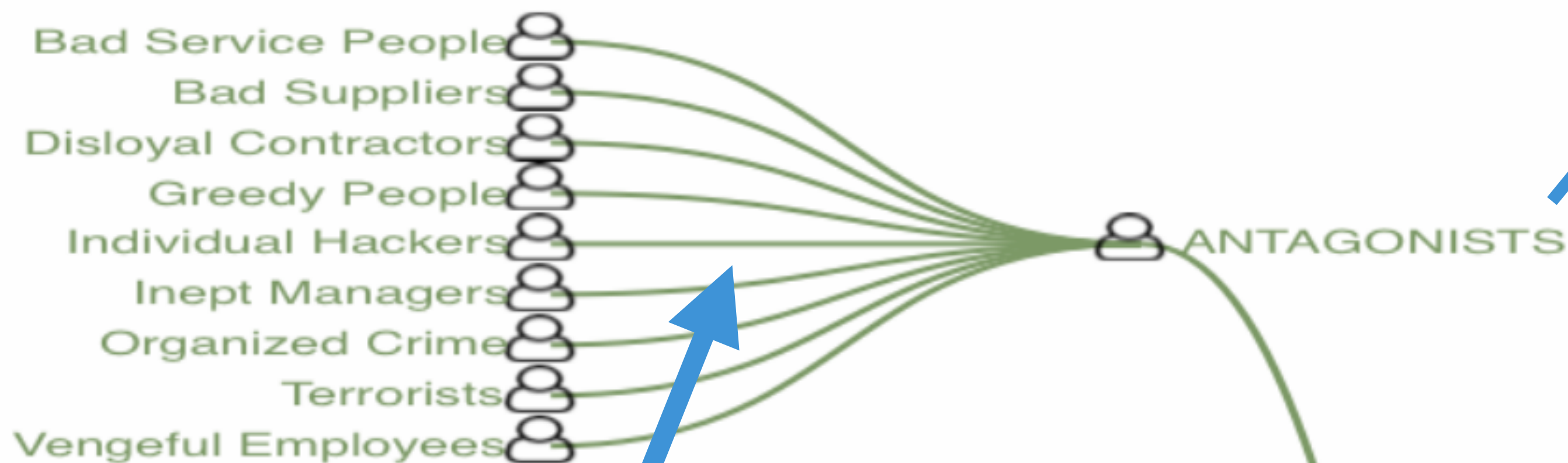
“We see our customers as invited guests to a party, and we are the hosts. It's our job every day to make every important aspect of the **customer experience** a little bit better.”

Jeff Bezos, Amazon



# THE MAIN PRINCIPLES OF FAILURE

Part 3.1B . Do not analyze **stakeholders**, stick to **'customers'** and **'users'**. As convention dictates.



Stakeholder categories.

Notice at top

the **'Antagonists'**,  
your *enemies* and  
*competitors*.

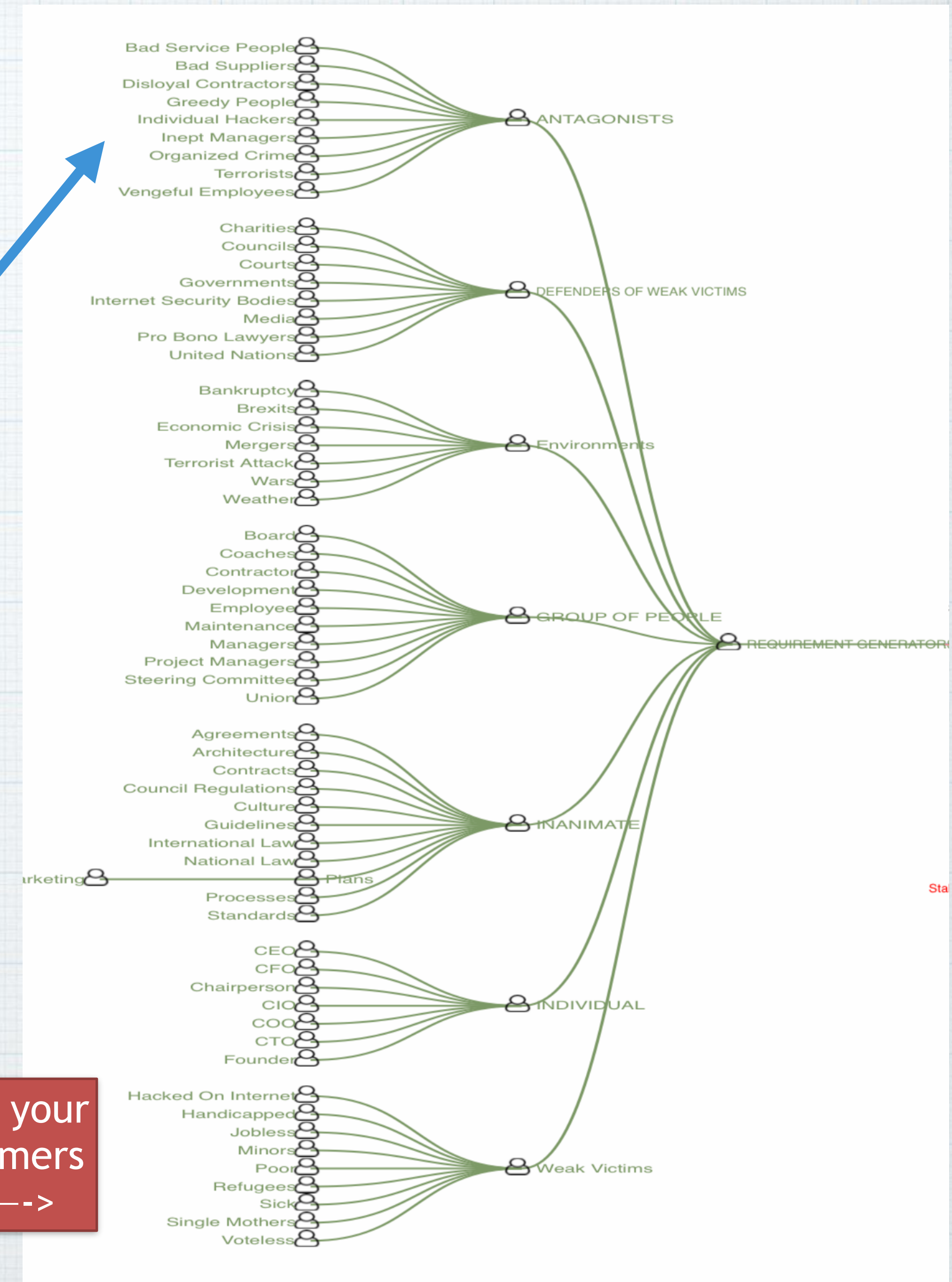
Not

*users* and *customers*, only.

Forget about these **'saboteurs'** in your requirements,

And you will be sabotaged, sooner or later.

Maybe, some of your  
Users and customers  
Are down here -->





Part 3.2. Do not clarify stakeholders values.  
Give them solutions they say they want.

(you can interpret their unclear requirements, any way you want, the cheapest option).

Consider the following blog for agile:

“AGILE PRODUCT MANAGEMENT IS LIGHTWEIGHT, CONTINUOUS, SMALLER IN TERMS OF EFFORT, AND LESS LINEAR.

THE ERA OF BUILDING BIG, LONG-TERM STRATEGIES DESIGNED UPFRONT, BOTH FOR BUSINESS MODELS AND PRODUCT LINES, IS BEHIND US. AGILE HAS ENABLED BUSINESSES TO ACCELERATE THEIR VALUE DELIVERY TO THE MARKET – BUT OFTEN AT THE EXPENSE OF PRODUCT STRATEGY. THE REASON FOR THIS IS THAT, LED BY A WIDELY PERVASIVE AND MISTAKEN VIEW THAT AGILE IS ONLY ABOUT DELIVERING SOFTWARE AND A DESIRE TO GET ON THE “AGILE TRAIN,” BUSINESSES FAILED TO DETERMINE THE ROLE OF STRATEGY, LONGER-TERM PLANNING, AND CUSTOMER RESEARCH IN AN AGILE ORGANIZATION. AGILE WAS BEING USED TO CREATE PRIORITIZED BACKLOGS FOR DELIVERING VALUE – OFTEN IN THE FORM OF WIDGETS OR FEATURES THAT MAY OR MAY NOT HAVE BEEN WHAT CUSTOMERS NEED MOST – AND MOST WERE HAPPY JUST TO DELIVER SOMETHING ON TIME AND WITHIN BUDGET.

TODAY WE RECOGNIZE THE WEAKNESS IN LACKING PRODUCT STRATEGY AND CUSTOMER UNDERSTANDING: CUSTOMERS DON’T CARE ABOUT MORE FEATURES. THEY CARE ABOUT SOLVING THEIR PROBLEMS – AND AGILE PRODUCT MANAGEMENT RESTORES AN ORGANIZATION’S CAPABILITY TO DETERMINE WHAT CUSTOMERS NEED AND WHAT MARKET OPPORTUNITIES MIGHT EXIST OR NEED TO BE CREATED. AGILE PRODUCT MANAGEMENT, AMONG OTHER THINGS, ENSURES THAT PRODUCT BACKLOGS REPRESENT OUR BEST LEARNING ABOUT CUSTOMER NEEDS AND DESIRES, WHILE HELPING REALIZE SUCCESSES HYPOTHESIZED BY AGILE PRODUCT MANAGERS. ”

This is a real example of some ‘agile’ guy just blabbing on, with nice sounding words, of no real content, proof, evidence, or cases. You can see from the Z, he is North American. (see presenter notes source) Does he think we are stupid and gullible? Will buying his services lead to project disaster?

How many ambiguous, untestable, unmeasurable, unintelligible words can you spot, in this typical real set of requirements?

Projects and Major Connections		What about: at least 50, Ambiguity defects?
Deliver connections work to the customer according to their preferred timelines	CC-12	Projects: Meet the customer’s preferred date for delivery of a quote
	CC-13	Load - Reduce average time to quote to 35 calendar days with 100% within 75 calendar days
	CC-14	100% of Projects & Major customers will have delivery on their preferred date (conditional: timescales for streetworks notices, system emergencies, severe weather events preventing work from being carried out.)
	CC-15	We provide up-front informal dialogue before the formal application to help customers make a decision and application that suits their needs
We are fit to compete	CC-16	Customers have a greater choice of commercial arrangements for the services we offer (e.g. flexible payments)
	CC-57	Proactively engage with customers if ‘interactivity’ applies, recognise, manage customer expectations and remaining capacity or redesign
	CC-17	Provide sales account managers to top 50 (approx.) Projects customers and all Major Connections customers for developing long-term relationships, business development and to provide clear visibility of upcoming work
	CC-58	Workflow captures work stages and events to enable management and regulatory reporting
	CC-61	The processes and systems allow novel and innovative commercial terms to be used for connections (e.g. demand and generation limitations for network constraints)



### Part 3.3. (continued)

Commit to all the 'nice-sounding'

designs and strategies ('means').

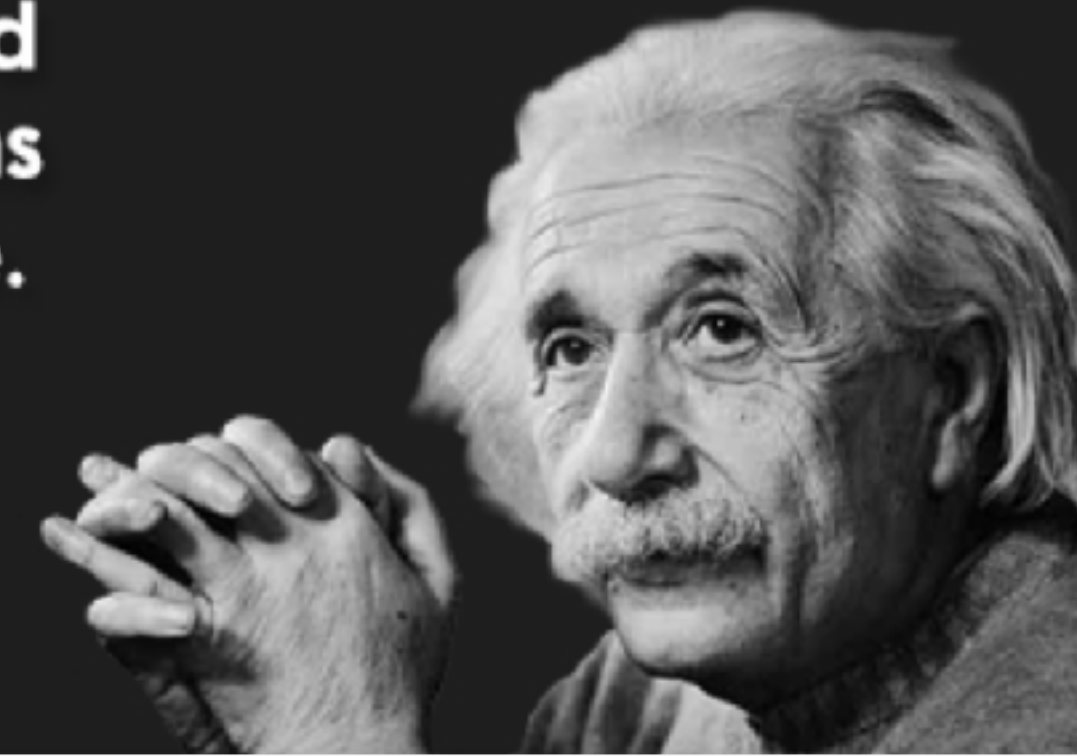
Especially the strategies on the managers' PowerPoint slides.

Case: \$100 mill. Sabotage in 1 year.

- "Achieve 'One Bank' vision
  - **through** globally integrated IT Portfolio Management,
  - **by implementation of** a single toolset
  - **supporting** existing (and consistent) processes across our IT
- Perform accurate measurement and tracking of project and non-project related IT expenses.
- Track and allocate human resources based on skills, level of work commitment and timing.
- Enable business alignment
  - **through the ability to** manage critical initiatives on a portfolio basis
  - **and support** faster time to market
  - **providing the potential for** increase in revenues.
- Enable the business and SMT to make sound management decisions around the portfolio, and optimize the IT spend
  - **so as to** effectively prioritize IT spend, and maximize business value.
- Replace resource intensive and disparate Portfolio Management tool, with industry "best in breed" capabilities.
- Improvement in the time it takes IT
  - **to** respond to business changes.
- Reduction in costs
  - **through** eliminating redundant projects.
- Better planning and tracking capabilities,
  - **so as to** reduce project cost and time overruns." <- The CIO (\$100 mill. IT failure)

Perfection of means and  
confusion of ends seems  
to characterize our age.

Albert Einstein



Real top-management (CIO of 1,000s) IT project objectives. I had the job of cleaning up this mess.

There are at least 10 'solutions' in that page. "eliminating redundant projects."

"And as many badly-defined values. "Business Alignment"

As proven by the underlined bold 'link words

'Enable business alignment'. <— VAGUE OBJECTIVE 'through the ability to' <— LINK WORDS

"manage critical initiatives on a portfolio basis". <— MEANS

What are the IT staff going to do?

1. Implement the **unclear objectives**, ignoring the suggested means, or
2. Implement the **unclear means** suggestions from their CIO, and
3. hope the vague objectives will be reached.
4. Or at least assume that the CIO is too unenlightened about IT objectives to notice that he has been sabotaged?

<--- FOR DETAILS AND FOR QUANTIFIED REWRITE SEE  
<http://www.gilb.com/dl532>



Part 34. Make use of the most **'widespread agile'** project development methods.  
**Popularity** is a sure sign of oversimplified training,

Methods which oversimplify training,  
have **failure rates** (total and partial) that are **over 50%**, for years on end,  
and no one does anything effective about it!

"Do not repeat tactics just because  
they have gained you one victory. Let  
**your methods** be regulated by the  
infinite variety of circumstances."  
- Sun Tzu



# 8 REASONS

## Why Agile Fails in Large Enterprises

in LukSchools

Sanjay Zalavadia

1

Lack of clarity

Continual reliance on legacy methods

2

3

Inadequate experience with agile

Lack of collaboration in teams composed by  
different companies

4

5

Lack of a Testing Strategy

Lack of alignment in other areas of the enterprise

6

7

Larger teams and big pyramid structures

Not changing the objectives

8



Part 3.5. Make sure no one ever **estimates how effective a design or strategy will be.**  
 Or what it will cost in the short term or long term.  
 Such estimates are rarely perfect  
 and might distract from using perfectly nice and modern-sounding designs.  
 Like AI, blockchain. Or big data.

Requirements	💡 Ideal XAI Technol...	💡 Notional Technique	💡 UC Berkeley Deep ...	💡 UCLA Pattern Theory+	💡 Charles	Show Sideba
(→) <b>AI Accountability</b> =: Status: 1 → Wish: 90 % of [AI S...Δ%:	100%	56%	????	????	????	
(→) <b>AI Controllability</b> =: Status: 50 → Wish: 98 % of [AI S...Δ%:	100%	88%	????	????	????	
(→) <b>AI Ethicality</b> =: Status: 5 → Wish: 90 % #Respect...Δ%:	100%	-4%				
(→) <b>AI Privacy</b> =: Status: 1 → Tolerable: 0.5 % of [Priv...Δ%:	100%	80%				
(→) <b>AI Robustness</b> =: Status: 70 → Wish: 99.9 % #Intende...Δ%:	100%	33%				
(→) <b>AI Safety</b> =: Status: 100 → Wish: 10 Number of %:	100%	47%	????	????	????	
(→) <b>AI Transparency</b> =: Status: 10 → Wish: 99.998 % of selec...Δ%:	100%	47%	????	????	????	
(→) <b>AI Usability</b> =: Status: 1 → Wish: 0.1 Speed in M...Δ%:	100%	33%	????	????	????	
(→) <b>AI Learning Performance</b> =: Status: 0 → Wish: 50 % [Learnin...Δ%:	100%	84%	????	????	????	

This is a quick mockup without real data:  
 to show the potential for evaluating AI  
 Techniques  
 and their effects on  
 AI Values (or 'Qualities')



Part 3.6. For goodness sake. Do not waste energy trying to estimate the **side-effects** of exciting strategies, on your **critical objectives and costs**. Such insights would delay your 'will to get on with it', and overrun the deadline.

\* Main effect

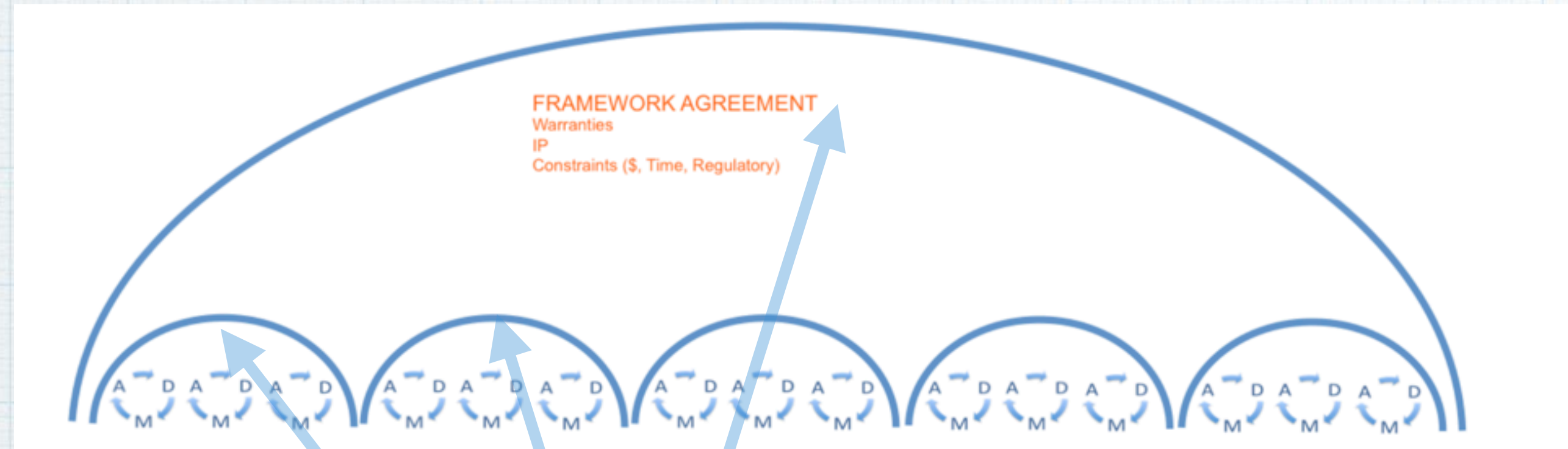
\* Side effects

\* Costs

	💡 S1 End Poverty	💡 S2 End Hunger Str...	💡 S3 Healthy Lives	💡 S4 Quality Educatio
<b>Requirements</b>				
🚦 <b>G1. Poverty (Decomposed)</b> Status: 0 → Goal: 100 % of sub-gΔ%:	95 95 % <div><div>95%</div></div>	???? 0 % <div><div>????</div></div>	???? 0 % <div><div>????</div></div>	???? 0 % <div><div>????</div></div>
🚦 <b>G2 End Hunger</b> Δ: Status: 0 → Goal: 100 % of sub-gΔ%:	42 42 % <div><div>42%</div></div>	96 96 % <div><div>96%</div></div>	???? 0 % <div><div>????</div></div>	Δ:???? Δ%: 0 % <div><div>????</div></div>
🚦 <b>G3 Healthy Lives</b> Δ: Status: 0 → Goal: 100 % of sub-gΔ%:	-42 -42 % <div><div>-42%</div></div>	23 23 % <div><div>23%</div></div>	???? 0 % <div><div>????</div></div>	???? 0 % <div><div>????</div></div>
🚦 <b>G4 Quality Education</b> Δ: Status: 0 → Goal: 100 % of sub-gΔ%:	5 5 % <div><div>5%</div></div>	-12 -12 % <div><div>-12%</div></div>	???? 0 % <div><div>????</div></div>	???? 0 % <div><div>????</div></div>
🚦 <b>G5 Gender Equality</b> Δ: Status: 0 → Wish: 100 % of sub-gΔ%:	-5 -5 % <div><div>-5%</div></div>	0 0 % <div><div>0%</div></div>	???? 0 % <div><div>????</div></div>	???? 0 % <div><div>????</div></div>
🚦 <b>G6 Water And Sanitation</b> Δ: Status: 0 → Wish: 100 % of sub-gΔ%:	???? 0 % <div><div>????</div></div>	42 42 % <div><div>42%</div></div>	???? 0 % <div><div>????</div></div>	???? 0 % <div><div>????</div></div>
🚦 <b>G7 Energy Access</b> Δ: Status: 0 → Goal: 100 % of sub-gΔ%:	0 0 % <div><div>0%</div></div>	3 3 % <div><div>3%</div></div>	???? 0 % <div><div>????</div></div>	???? 0 % <div><div>????</div></div>
🚦 <b>G8 Employment And Growth</b> Δ: Status: 0 → Goal: 100 % of sub-gΔ%:	33 33 % <div><div>33%</div></div>	-12 -12 % <div><div>-12%</div></div>	???? 0 % <div><div>????</div></div>	???? 0 % <div><div>????</div></div>
<b>Sum Of Values:</b> Σ%:	128 %	140 %	0 %	0 %
✂️ <b>Annual Cost Per Dwelling</b> Δ: Status: 5 → Budget: 1 Cost for t... Δ%:	-1 25 % <div><div>25%</div></div>	-0.5 13 % <div><div>13%</div></div>	???? 0 % <div><div>????</div></div>	???? 0 % <div><div>????</div></div>
✂️ <b>Years To Do</b> Δ:	3 <b>41</b>	4	????	????



Part 3.7. Do Not use 'No cure, No pay contracting' (sounds suspiciously agile to me)  
Get a fixed price for badly specified results, and finally pay 8 times more than lowest bid:  
that is the Waterfall Way



\* The Flexible Contract enables modular delivery by operating at two levels:

- The first level is the **Main Agreement**
  - which includes the Schedules.
  - The Main Agreement puts in place what is known as a framework arrangement.
  - There is **no contractual commitment for specific deliverables under the Main Agreement.**
  - The Main Agreement also sets forth the direction and constraints of the working relationship, as well as all of the legal provisions such as warranties, etc which apply to the SOTO.
- The second level comprises the **Statements of Target Outcomes (SOTOs).**
  - These are similar to a statement of work in a traditional contract, only you **deliver measurable outcomes instead of 'work'** in the form of outputs or activities.
  - These are entered into by the parties under the umbrella of the Main Agreement.
- The customer and supplier can focus on different target outcomes under each SOTO as agreed upon by the customer and supplier.
- This in turn means that the parties can build upon the knowledge gained over the course of the course of the project to date.

Traditional Contract Model	Result Contract Model (Agile)
Requirements are contractual and specified up-front in the main contract.	Requirements are specified at the start of each result cycle.
Changes are managed by means of the change control mechanism.	Requirements are more resistant to change than traditional output requirements. Target outcomes are only specified at the start of each result cycle, are operational for shorter periods of time, and therefore are exposed to less change.
Analysis, design, development, and testing occur sequentially. Big Bang or Waterfall.	Each cycle must deliver value, so design and development occur concurrently. A systems view must be taken, providing real results in real life.
An all or nothing solution.	The solution evolves as a series of result deliveries.
Constituent modules of software are worked on independently until integration takes place.	There is continuously working and stable software and hardware system.
Testing is used as a contractual tool at the end of the development process.	Testing occurs throughout the development process, providing feedback for improvements.
Success is measured by reference to conformance with the change-controlled contract.	Success is measured, cycle by cycle, by requirements delivered, driving value to the customer.



- The Project Saboteur,
  - need hardly lift a finger
  - to ensure project-failures.

FAILED PROJECTS  
ARE  
A NATURAL  
EXPECTED  
PART OF  
OUR CURRENT CULTURE

- The failure rate is high and stable over the long term (decades),
- proving that we have no ability
  - to change to success, or
  - to 'zero project failures'

But, wait a minute...

SURELY THERE IS NO  
Real CONSPIRACY  
TO MAKE US FAIL, TOM  
YOU ARE JUST KIDDING  
WITH US, RIGHT?

# Profitability *With* No Boundaries



*Focus*  
*Reduce Waste*  
*Contain Variability*  
*the TOC, Lean, Six Sigma Results*

(Russ) M. Pirasteh  
Robert E. Fox



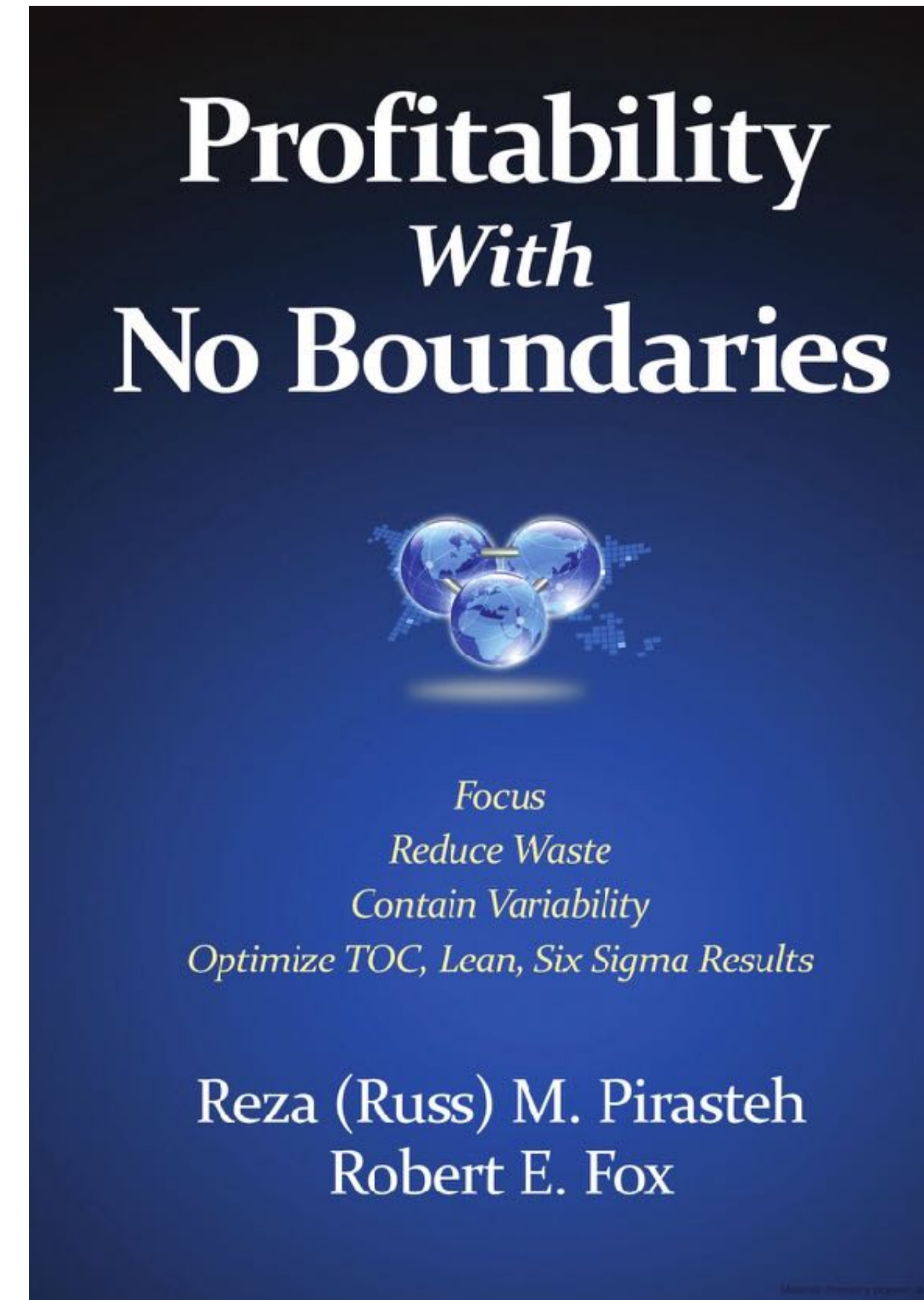
## Part 3.8.2



**He is considered to be the father of the Toyota Production System,**

**which became Lean Manufacturing in the U.S.**

**He devised the seven wastes (or muda in Japanese) as part of this system.**



page 99

Thanks to Lukasz Szostek for Finding this source Nov. 2018



## Part 3.8.3

# The Lean Methods Conspiracy to Destroy (Europe and The USA)

assisting them in copying our process.

Ohno paused, as if to consider his next words, and then said, “I’m proud to be Japanese and I wanted my country to succeed. I believed my system was a way that could help us become a modern industrial nation. That is why I had no problem with sharing it with other Japanese companies, even my biggest competitors. But I was very, very concerned that you Americans and the Europeans would understand what we were doing, copy it, and defeat us in the marketplace.”

He went on to say that when Americans and Europeans came to visit Toyota that he did his best to confuse them as to why Toyota was so successful. He said, “I explained it by talking about techniques, like quicker machine setups, reduction of the seven wastes (muda), and other techniques with Japanese names like kanban and kaizen. I did my best to prevent the visitors from fully grasping our overall approach. Today I am ready to be open and explain fully what we did. We are now strong enough to deal with any competition.”

He elaborated on why his river system was a much more efficient way to make automobiles and many other products. “We have tried to tie all our research and improvement efforts directly to the sales of our cars. That way a



**Profitability  
With  
No Boundaries**



Focus  
Reduce Waste  
Contain Variability  
Optimize TOC, Lean, Six Sigma Results

Reza (Russ) M. Pirasteh  
Robert E. Fox



Part 3.8.4

Takeuchi and Nonaka: The Roots of Scrum - Scrum Inc

<https://www.scruminc.com/takeuchi-and-nonaka-roots-of-scrum/>

Scrum for software was directly modeled after "The New New Product Development Game" by ... by Jeff Sutherland | Oct 22, 2011 | Blog | 7 comments ... Taiichi Ohno, the inventor of the Toyota Production System says everything he knows he ...

Nov 29 - Nov 30    Certified Scrum Master  
Dec 3 - Dec 4     Certified Scrum Master  
Dec 6 - Dec 7     Certified Scrum Product Owner

Oh No!  
Maybe Japanese are not  
credible sources  
Nor Americans  
Who believe them



Jeff Sutherland  
(Warsaw ABE Lecturer, "Scrum 19% Failures")

<https://www.scruminc.com/takeuchi-and-nonaka-roots-of-scrum/>

"Tom Gilb invented Evo, arguably the first Agile process. He and his son Kai have been working with me in Norway to align what they are doing with Scrum.

Kai has some excellent case studies where he has acted as Product Owner. He has done some of the most innovative things I have seen in the Scrum community."

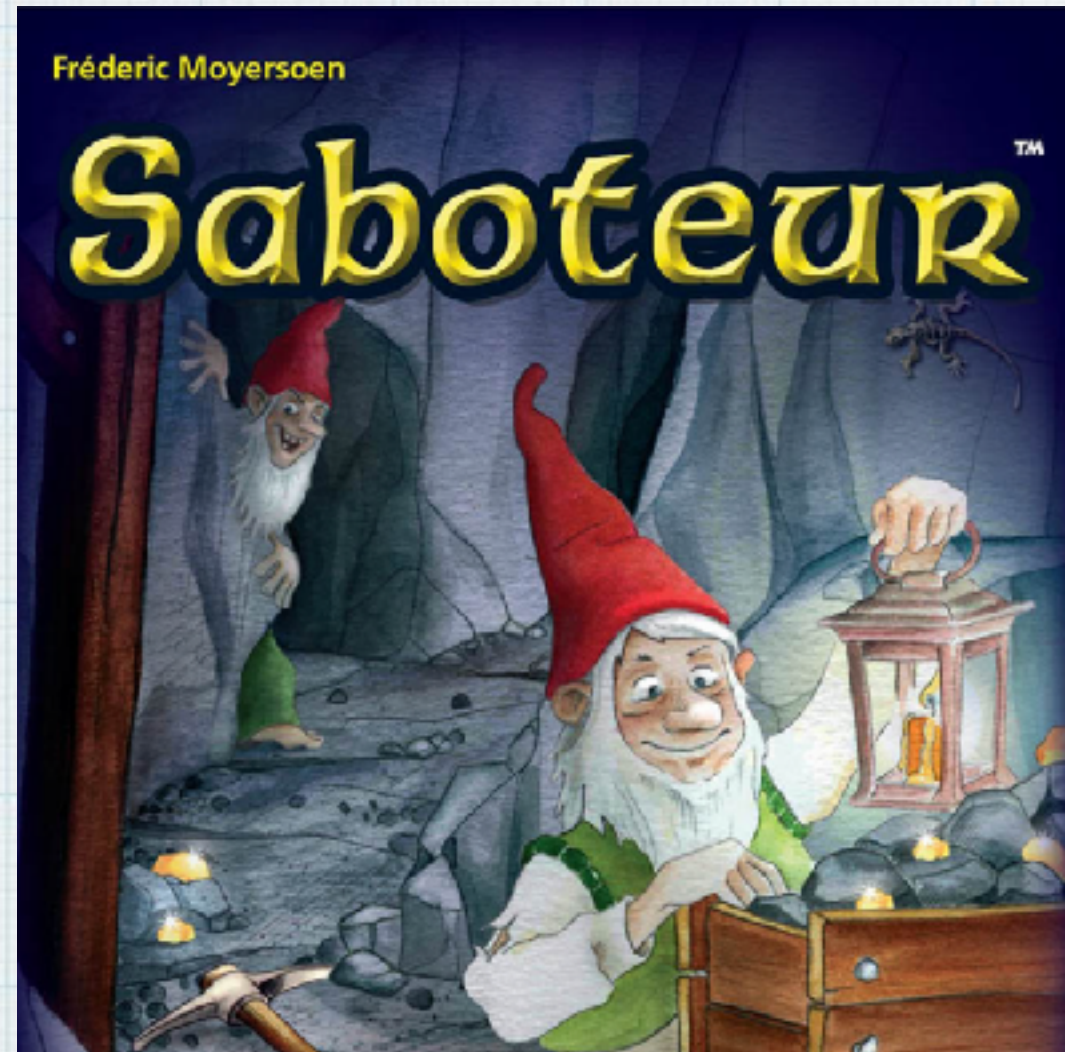
Jeff Sutherland, co-inventor of Scrum, 5Feb 2010 in Scrum Alliance Email.

"Tom Gilb's Planguage referenced and praised at #scrumgathering by Jeff Sutherland. I highly agree" Mike Cohn, Tweet, Oct 19 2009

Taiichi Ohno







Kinser, J. (2008). The top 10 laws of project management. Paper presented at PMI® Global Congress 2008—North America, Denver, CO. Newtown Square, PA: Project Management Institute.

<https://www.pmi.org/learning/library/ten-laws-project-management-literature-6968>

Book, slide and paper References for this 'Project Failure' Part 3, are in the slide Presenter Notes Of this slide  
If you have a pdf slide copy, then get references from the book itself

The 'Value Agile' Book, Free: [tinyurl.com/ValueAgile](http://tinyurl.com/ValueAgile)

And of course also in the the 'Value Agile' book  
See URL at beginning

1. Augustine's Law: "A bad idea executed to perfection is still a bad idea."
2. Lakein's Law: "Failing to plan is planning to fail."
3. Saint Exupéry's Law: "Perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away."
4. Fitzgerald's Law: "There are two states to any large project: Too early to tell and too late to stop."
5. Parkinson's Law: "Work expands to fill the time available."
6. Constantine's Law: "A fool with a tool is still a fool."
7. Graham's Law: "If they know nothing of what you are doing, they suspect you are doing nothing."
8. Murphy's Law: "If anything can go wrong, it will."
9. O'Brochta's Law: "Project management is about applying common sense with uncommon discipline."
10. Kinser's Law: "About the time you finish doing something, you know enough to start."



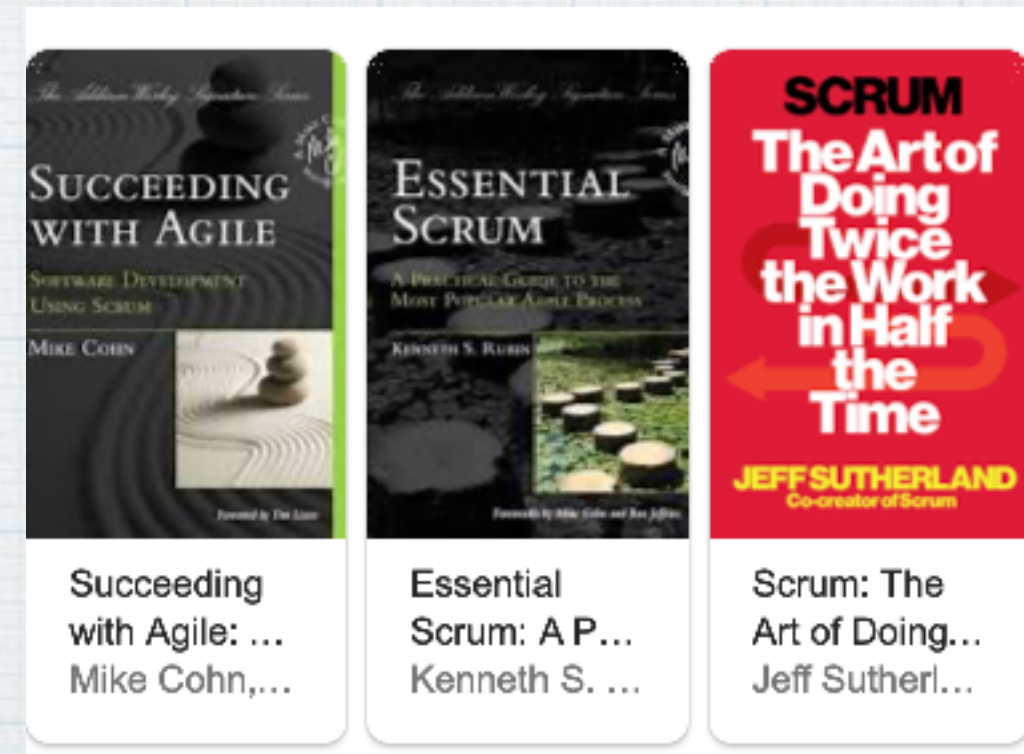
# What is 'Agile As it should be'. ?

[www.gilb.com/dl561](http://www.gilb.com/dl561)

←----- The 'real agile', as it originally was. ----->

## \* Pseudo Agile (Generic Frameworks ?)

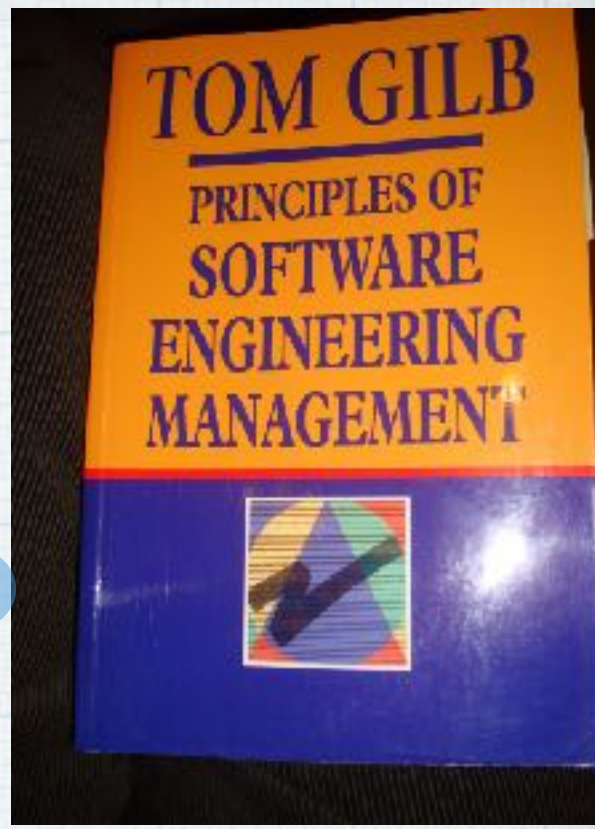
- \* A belief culture
- \* No Quality Measures
- \* No Cost Measures
- \* A Craft culture
- \* Small scale culture
- \* Programming culture
- \* Fails too often



## \* Real Agile (Detailed Engineering Technology)

1988

- \* Stakeholder Value focussed
- \* Cost-Effectiveness (Efficiency)
- \* Systems (not code)
- \* Scale Free
- \* Engineering
- \* Fact Based Incremental Feedback driven
- \* Successful Value Delivery



2005

Gilb: Principles of Software Engineering Management, (1988).  
Chapt 15 Deeper Perspectives on Evolutionary Delivery  
[www.gilb.com/dl561](http://www.gilb.com/dl561) See 15.1.10 Gilb SM 76



# Defining ‘Agile’

- “Any set of tactics that enable a prioritised stream of useful results, *in spite of a changing environment*”

– © Tom Gilb, 7 June 2013, for UK Bank Board (SLC)

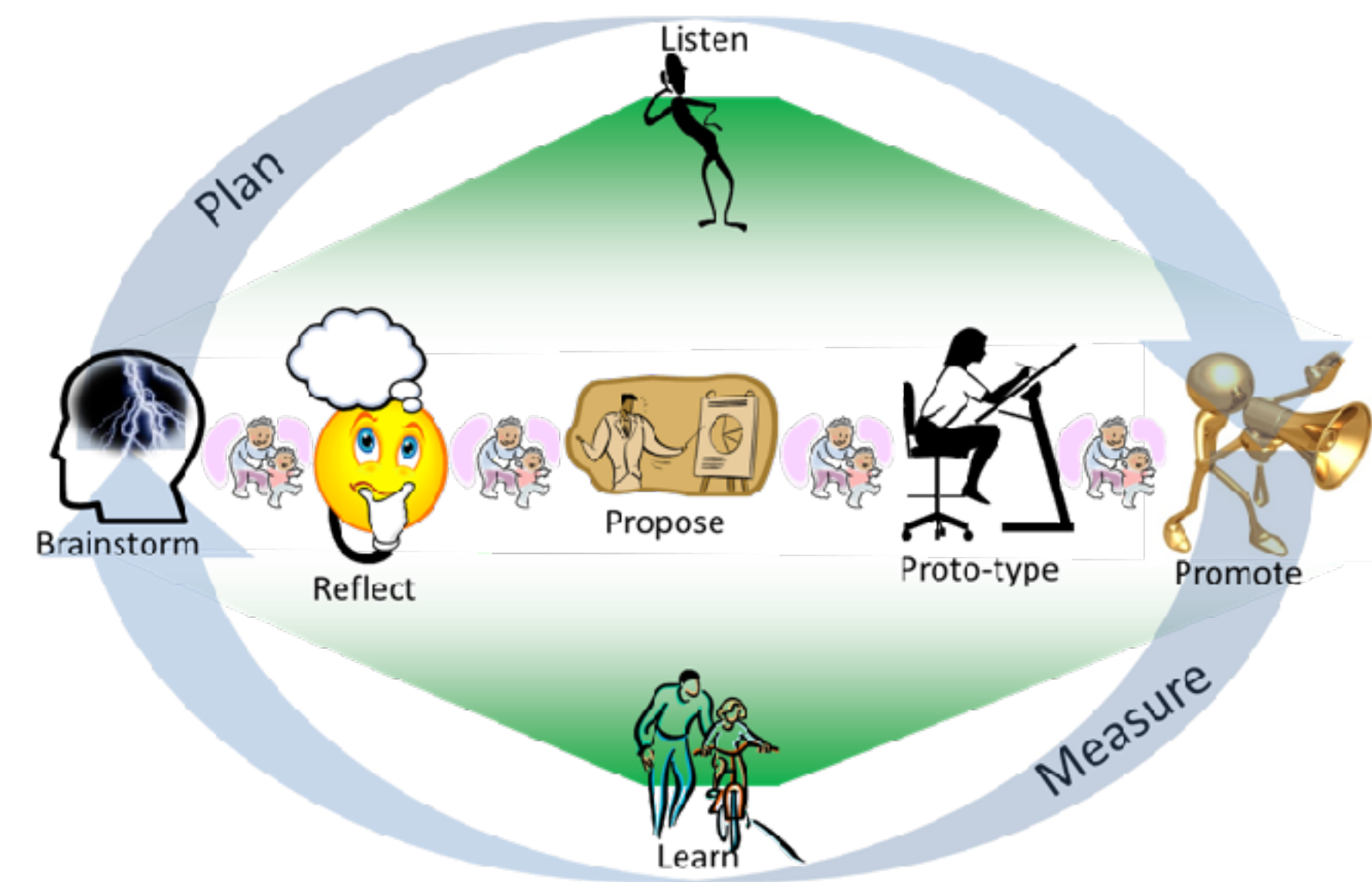
- A focus on doing ‘Agile’, (as a main objective, or culture)
  - is the *wrong* level of focus. A bad idea.

– Using agile tactics that ‘work’, is a good idea.

I think you should

- Focus on results, no matter what
- Agile processes, ie a ‘means’, to improve the ‘results’, ie ‘ends’,
  - are only as good as the improvement
  - in results
    - that are a *consequence* of using those agile processes.

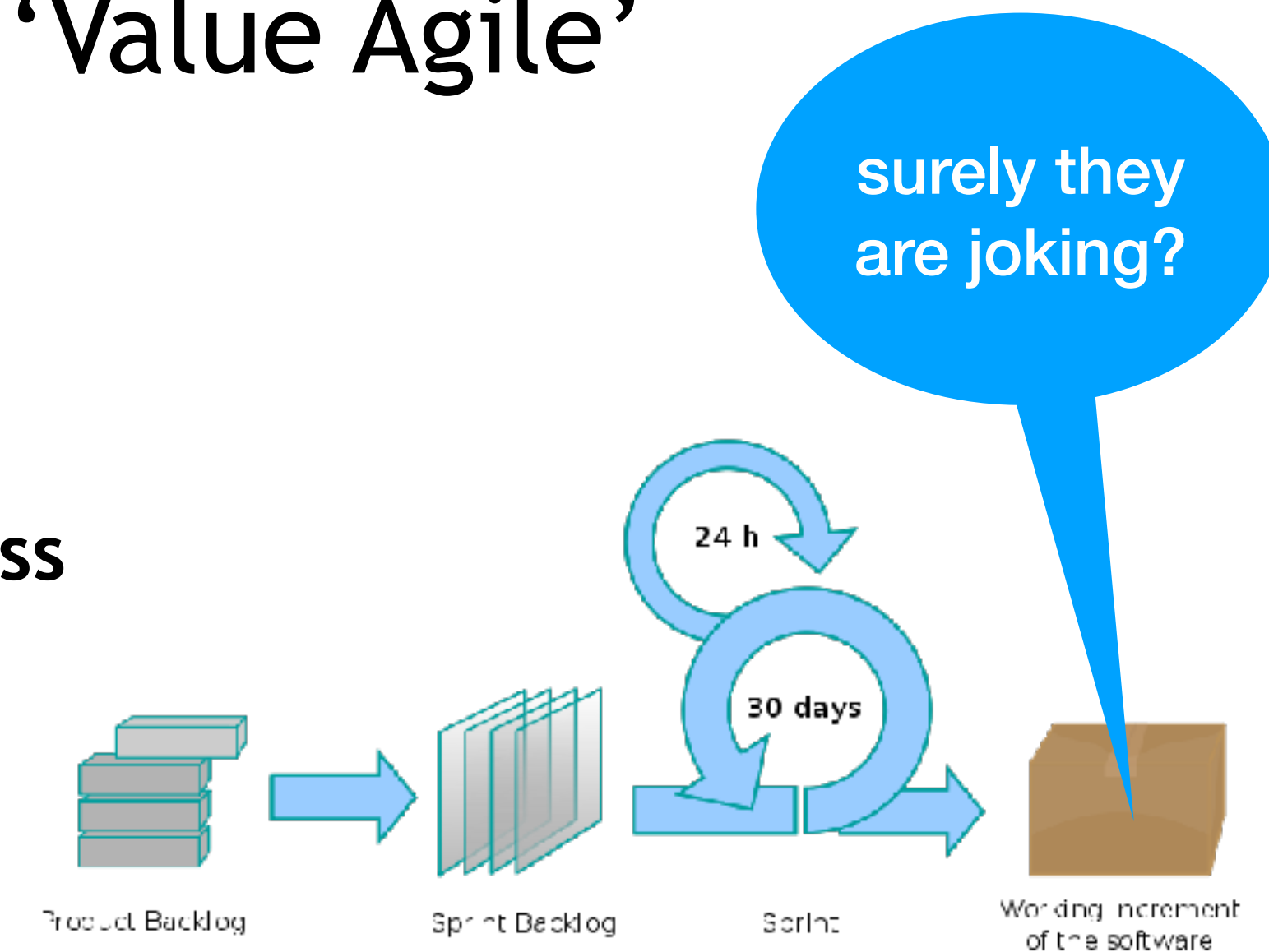
## The Generic Agile Concept





# ‘Traditional Agile’ and ‘Value Agile’

- **Traditional Agile for IT** (Scrum, XP, etc.)
  - Is *unfortunately* not ‘tuned in’ to **delivering business value**
  - It tries to speed up (‘velocity’) code production
  - *As it is now*, ‘traditional Agile’ is *not at all useful* for **business** purposes.
  - They are simply not really managing ‘values’.
  - They ‘talk’ about values, but they do **not quantify** and **manage** them. They do not ‘walk the talk’.



- **The ‘Value Agile’ Model** that we recommend (‘Evo’)
  - Is focussed on **business value** delivery
  - Is used to co-ordinate IT work, to deliver *measurable business value*
    - Deutsche Bank, for example, made ‘Evo’ their *standard* for managing all other Business ‘Agile’ work (Paul Fields, 2013-19)\*
  - Evo ‘connects’ the ‘**business with IT**’ efforts, and all other improvement efforts.
  - Evo is a **systems project management** method: not about code or IT alone. It is about *people, organisation, motivation, data, hardware*, and, ‘sometimes’, about *software*.



\* see presenter note for details on adoption



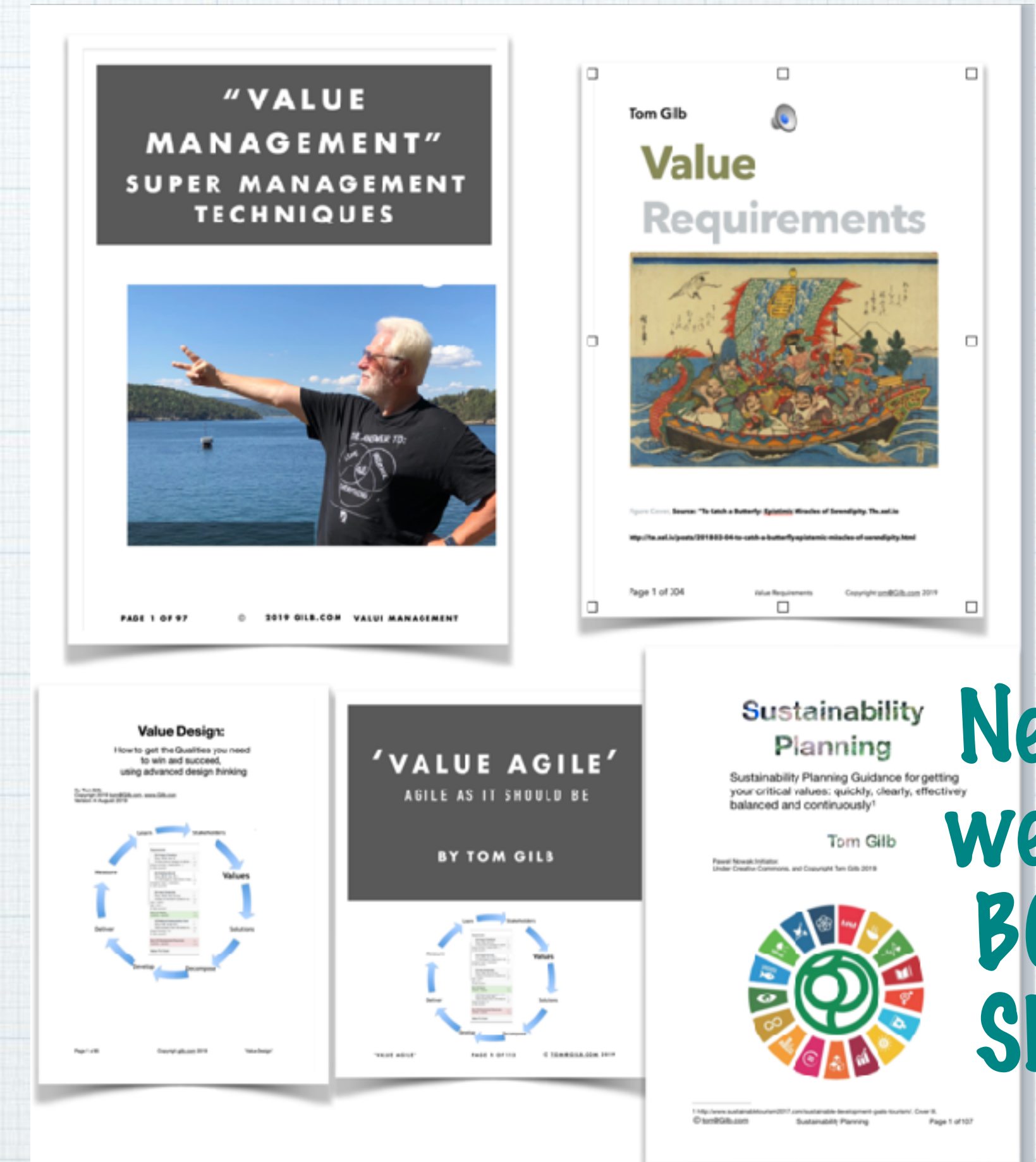
‘Stakeholder value delivery’ is the *real* point,

‘nice’ - *if* ‘agile’ can make  
Stakeholder value delivery  
*better and faster!*





# WE COULD STOP HERE, IF TIME is OUT



Next  
week  
BCS  
SPA

Free for the moment

[gilb.com](http://gilb.com). Paid books.  
Worth at least 1000x more than cost!  
( I think so at least :) )

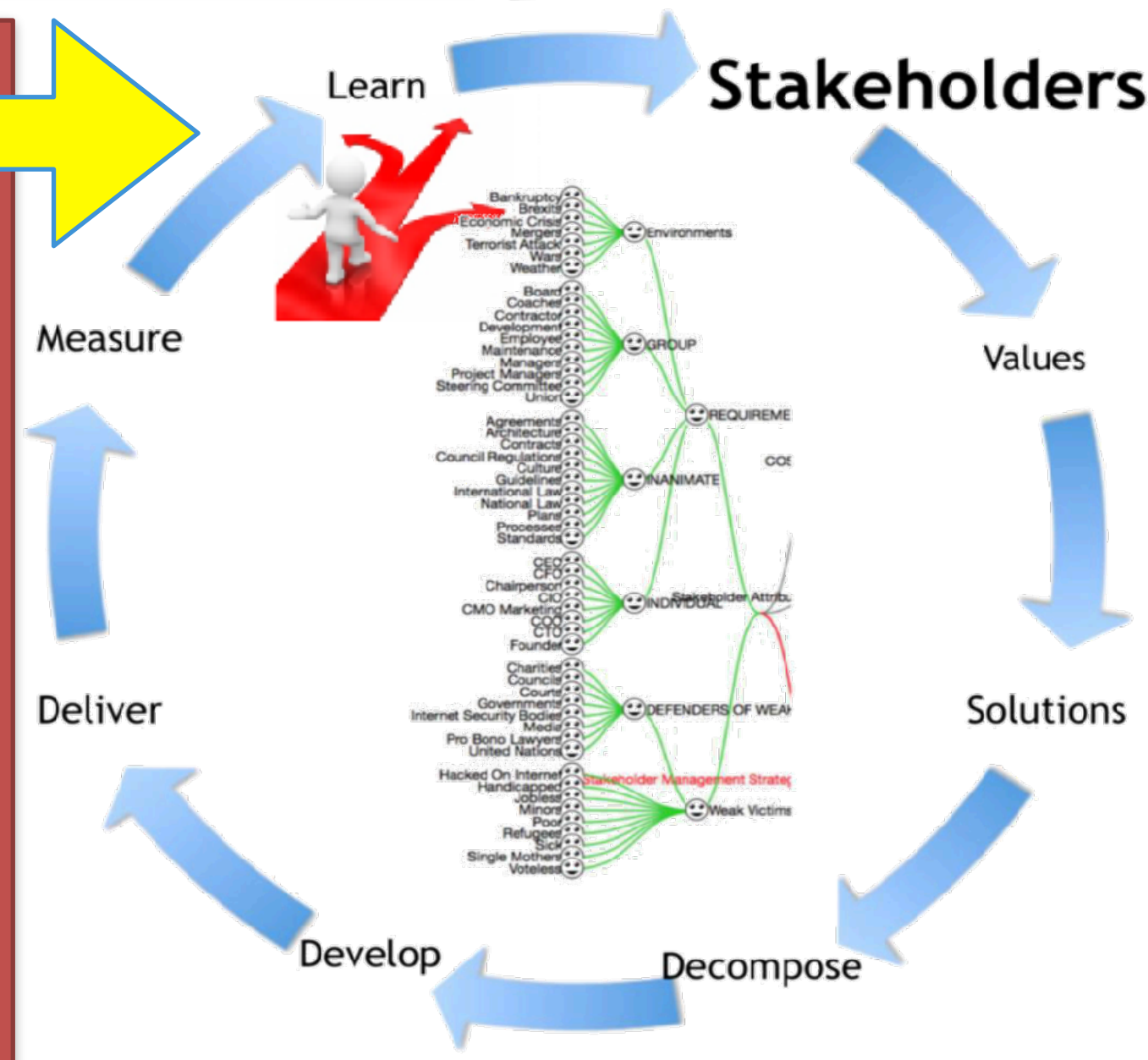
All 2019 Books PDF

[https://www.dropbox.com/sh/adcrki52xo5zb36/AABMD\\_2GOX4rT6c-HRCmT-Qua?dl=0](https://www.dropbox.com/sh/adcrki52xo5zb36/AABMD_2GOX4rT6c-HRCmT-Qua?dl=0)



# Gilb's 'Value-Driven Planning' Principles: 'Prioritize Value'

- 1. 'Critical' Stakeholders determine the values you must manage
- 2. 'Critical' Stakeholder Values *can* and *must* be quantified
- 3. Values are supported by their Value-impacting Architecture (you get the values you *design*, not just the ones you 'require')
- 4. Value 'Goal' levels are determined by timing (*when* you need a level),
  - architecture effect (how *good* your design is),
  - and resources (money, time, people you can afford, or which pay off)
- 5. Value levels can differ for different 'scopes' and conditions (where, who, activity, environment)
- 6. Prioritised Values can, and should, be delivered extremely *early* (this month).
- 7. Value-level delivery levels can be 'locked in' incrementally, ratcheting. Fail-Safe.
- 8. New high-priority Values, and value levels, can be *discovered* (external news, experience) later; anytime, late. React agile.
- 9. You can *estimate* the impacts on all critical values (your 'ends'), of all proposed 'means' (designs, strategies, architectures, solutions) .
- 10. Value delivery will *attract* resources. (money seeks profit)



Requirements	🔦 S1 End Poverty	🔦 S2 End Hunger Str...	🔦 S3 Healthy Lives	🔦 S4 Quality Educatio
➡ G1. Poverty (Decomposed)	95	????	????	????
Status: 0 ➡ Goal: 100 % of sub-goal	95 %	0 %	0 %	0 %
	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>
➡ G2 End Hunger	42	96	????	Δ ????
Status: 0 ➡ Goal: 100 % of sub-goal	42 %	96 %	0 %	Δ %: 0 %
	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>
➡ G3 Healthy Lives	-42	23	????	????
Status: 0 ➡ Goal: 100 % of sub-goal	-42 %	23 %	0 %	0 %
	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>
➡ G4 Quality Education	5	-12	????	????
Status: 0 ➡ Goal: 100 % of sub-goal	5 %	-12 %	0 %	0 %
	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>
➡ G5 Gender Equality	-5	0	????	????
Status: 0 ➡ Wish: 100 % of sub-goal	-5 %	0 %	0 %	0 %
	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>
➡ G6 Water And Sanitation	????	42	????	????
Status: 0 ➡ Wish: 100 % of sub-goal	0 %	42 %	0 %	0 %
	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>
➡ G7 Energy Access	0	3	????	????
Status: 0 ➡ Goal: 100 % of sub-goal	0 %	3 %	0 %	0 %
	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>
➡ G8 Employment And Growth	33	-12	????	????
Status: 0 ➡ Goal: 100 % of sub-goal	33 %	-12 %	0 %	0 %
	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>
Sum Of Values:	Σ %: 128 %	140 %	0 %	0 %
➡ Annual Cost Per Dwelling	-1	-0.5	????	????
Status: 5 ➡ Budget: 1 Cost for 1...Δ %:	25 %	13 %	0 %	0 %
	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>
➡ Years To Do	Δ: 3	4	????	????

Principles updated 180520 tg



# Beyond Scaling: Scale-free Principles for Agile Value Delivery - Agile Engineering.

© tom@Gilb.com 2016, Posted at [gilb.com resources/downloads/papers](http://gilb.com/resources/downloads/papers)

<http://www.gilb.com//dl865>

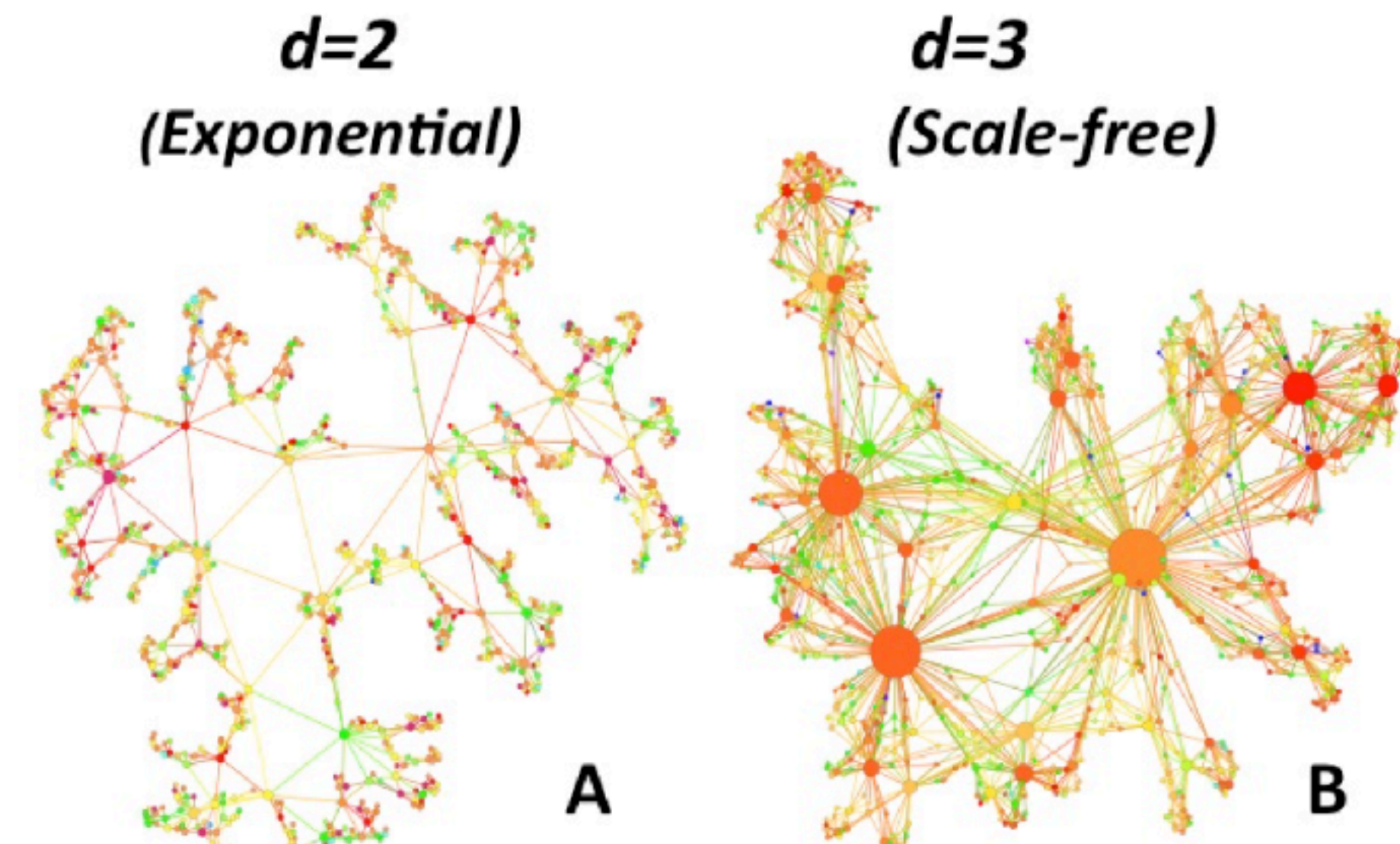
Version March 14 2016, Modified April 11 2016 (XP)

## Summary

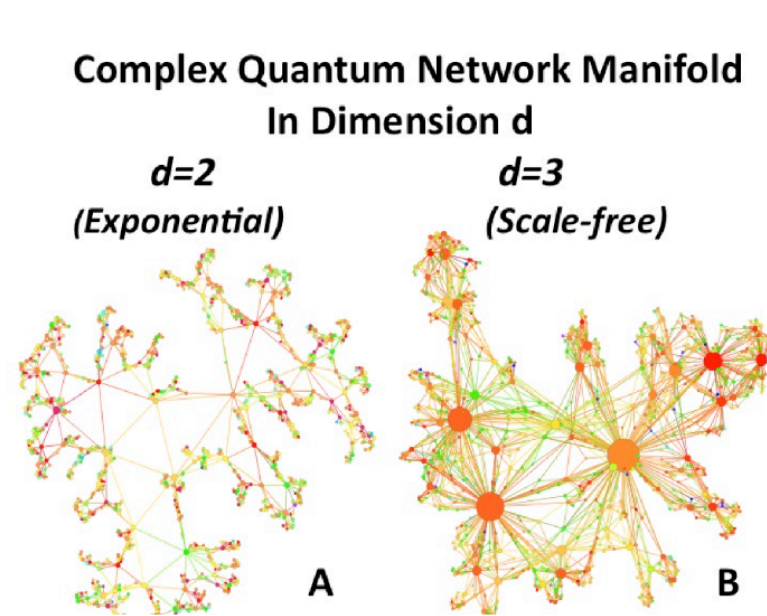
There is widespread interest in how to make Agile (including Scrum) methods, work better, on a large scale.

Mike Beedle's paper [1] gives a good *overview (references to much of the agile scaling literature) of many different proposed methods*. I am not going to argue whether these methods are good or bad. No doubt most of the techniques have some value in some circumstances. My concern is *not* this set of 'conventional agile scaling' ideas'. My concern is the large collection of

## Complex Quantum Network Manifold In Dimension $d$







# Scale-free Agile Principles



Value Planning <https://www.gilb.com/offers/SN2UR7vu/checkout>  
FREE GIFT REVIEW COPY FOR YOU  
ALONE. NO COUPON CODE  
REQUIRED.

the VP ref. below

1. Keep focus on measurable delivery of critical values and their costs. [3, 4, 5, 6, 9, 10, 12, VP (20) Part 1, VP 10.6 ]
2. Deliver value early, quickly and regularly: in roughly 2% increments. [14, 11, VP Ch.4, 2, 5 ]
3. Do NOT focus on code delivery; focus on overall system value and costs. [ VP Ch.4, 10D, 10F, 13, VP 3.4, VP 2.10, VP 9.8, 4, 12]
4. Focus on quantified *critical stakeholder* values. [19, VP 3.4, VP 3.7, VP 3.9, VP 3.10 VP 4.2, 10 ]
5. Synchronize all teams in terms of measurable value delivery. [VP 3.3, VP 3.4, VP Part 1, VP 3.6, VP 3.8, VP 8.4 , 11, 12, 13 ]
6. Solve big problems through ingenious architecture; not through coding faster. [VP 4.5, VP 5.1, VP 5.3, VP 7.2, 15 ]
7. Decompose the large problems by incremental value deliveries: not code deliveries. [7, VP Ch. 5, VP 5.1, VP 5.6 , 10, 11, 13, 15]
8. The software component needs to be integrated into the total system of hardware, data, people, culture. [ VP 5.2, 10 ]
9. If your team cannot deliver small increments of real value early, frequently, and predictably; they are incompetent and need to be abandoned for those who can deliver. [7, VP 2.8, 10]
10. Never commit to contracts for *work done* or *code delivered* alone: there must always be a sufficiently large contractual protection, of paying for measurable value delivered. [12, 15 ].



# Why do these Planguage Evo Scale-free ideas work?

## ● Value quantification

- allows us to focus on the stakeholder results, the main objectives of any project.
- All other activity, below this level should be contributing to delivery of the planned values.
- This means we can delegate the activity to any combination of specialist teams of any size and complexity: yet we can judge whether things are ‘working’.
- We keep our eyes on measured value delivery. We can judge whether both our organization and our architecture are delivering as expected and needed.
- If not we can adjust (**dynamic design to cost**) and go with things that are actually delivering necessary value.

## ● Contracting for value

- relates to the above explanation,
- with the added benefit that outside contractors are now motivated to focus on value delivery, not just ‘doing work’, or ‘programming’.
- It does not matter so much about the underlying complexity.
- That underlying complexity either works (delivers contracted value measurably) or not.
- If not, we change it until it does, or give up if we cannot change to satisfy value delivery needs.

## ● Decomposition by small 2% deliverable value architecture components:

- this is a very basic attack on large size and consequent complexity.
- We can see the incremental impact of each step on the whole system, regarding both value delivery and costs.
- If it is not good enough we try new ideas.
- If we run out of ideas that work, we need to stop.

## ● Risk Management:

- our methods, including 1-3 above, are really all about managing the risk of failing to deliver value for money, on time.
- In addition we have suggested a number of additional risk management ideas.
  - For example estimating the  $\pm$  uncertainty of a design impact on values and costs [9].
  - For example asking for specific evidence [9] that any given design, or strategy will deliver the values and costs we need.
- The more engineering effort we put in to planning for risk up front, the less likely we are to get nasty surprises later
  - (and then blame them on ‘project size and complexity’; rather than our own lack of decent engineering planning).

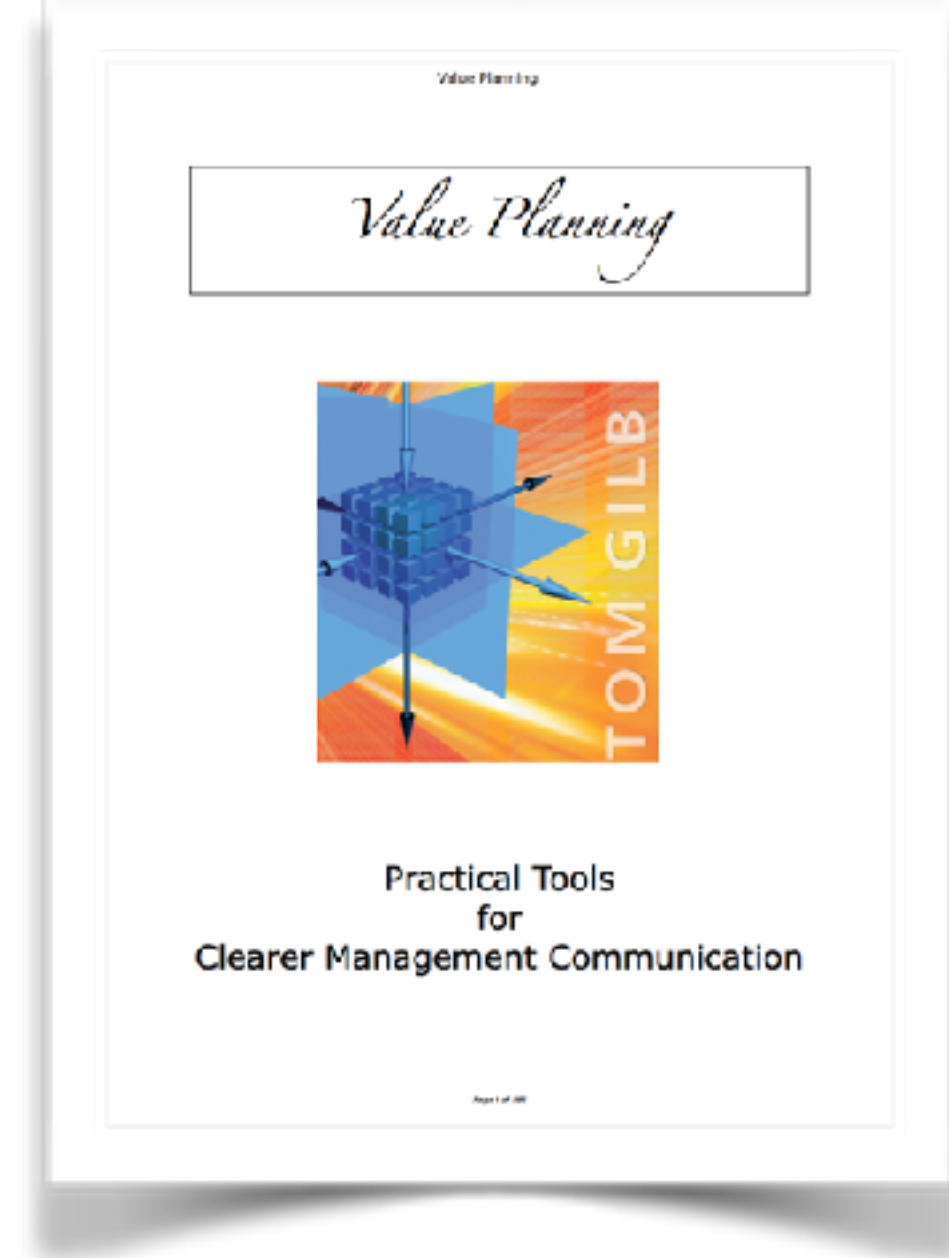
## ● Delegation of decision-making [23].

- **Delegating the power to make decisions to a grass roots level,**
- **and in addition to do so incrementally**
- **while keeping any eye of their level of concern (in terms of value and costs),**
- **should obviously help us make better decisions, in an evidence-based situation.**

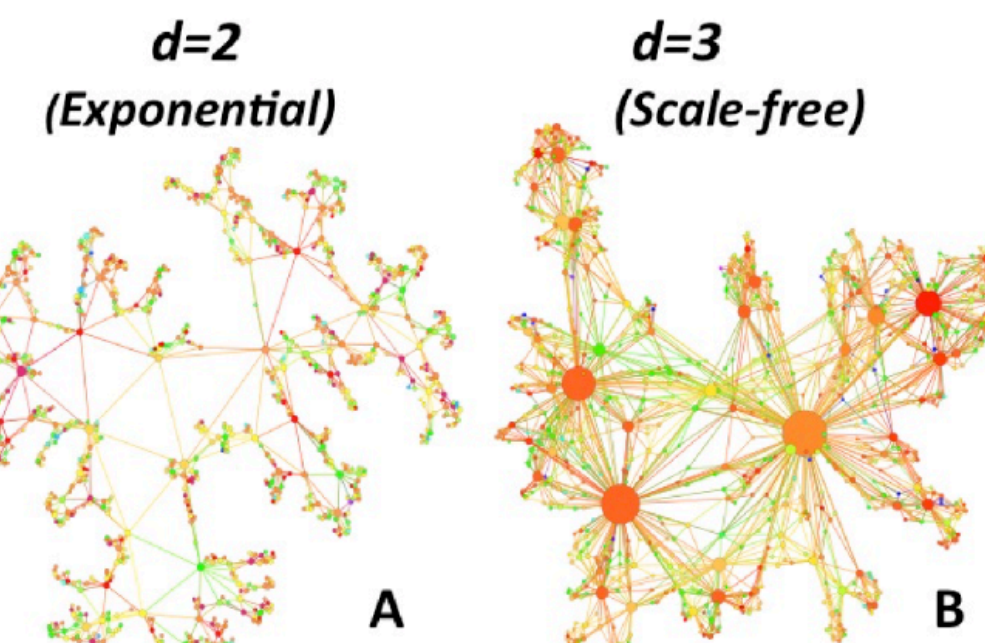
- I have personally used these methods, with remarkable success, on projects involving for example 1,000 programmers and 1,000 hardware engineers (example HICOM (which was in total failure mode after 2 years, at Siemens. Boeing Aircraft projects [thousands of employees involved. To mention just a couple of many). There is no doubt for me that they work, and why they work.

Top Level Results

Any level of solution complexity



## Complex Quantum Network Manifold In Dimension d

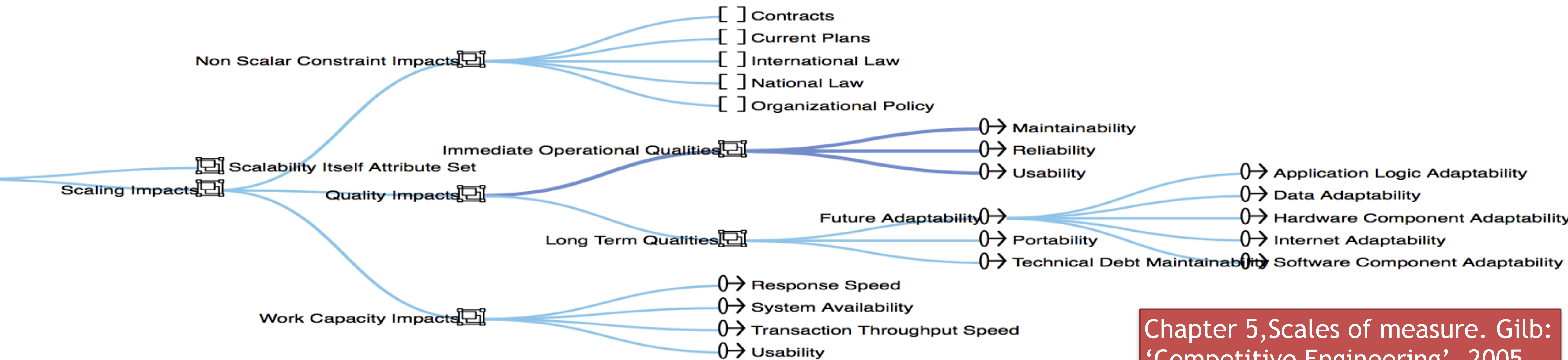


“SCALE-FREE:  
Practical Scaling Methods for Industrial Systems Engineering”  
lecture slides, <http://concepts.gilb.com/dl892>



# Scale-Free:

a set of tailored system properties, defined and measurable



Chapter 5, Scales of measure. Gilb:  
‘Competitive Engineering’, 2005  
<http://www.gilb.com/DL26>

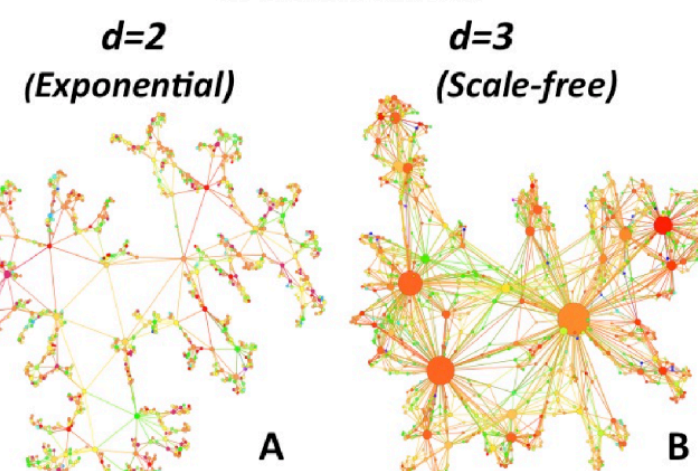
A systematic generic structure  
of some of the quantifiable quality variables  
we might consider  
when modelling a scalability problem

<http://concepts.gilb.com/dl930>

Scalability Metrics:

An Engineering Structure, and Principles, for an Agile World

Complex Quantum Network Manifold  
In Dimension d





## Erik Simmons, Intel, 20 years Experience Scaling with Gilb Methods

- “ Instead, I believe that the **majority of what you have** included for ideas, principles, etc. from CE and VP are in fact **scale-free**.
- They are **not dependent on *project* or *organization* size**.
- They are **good heuristics for almost any project**,
- and **nearly universally applicable**
  - (nearly universal because I hear Koen in my head, and all is heuristic).
- So, CE and VP are not *about* scaling
  - so much as they should be taught and understood as **scale-free**.
- Size is not a reason to choose (or not choose) to use Competitive Engineering, Evo, Planguage, etc.
- As you quoted me in the paper – **this stuff works**.
  - It works on **small** projects. It works on **large** projects.
- Evo on a 5-person team is not really much different than Evo on a 100-person team, except there are more people.
- The principles apply **without alteration** (or “scaling”).
- Anyone who sees a random page of your new paper would probably not guess the topic is scaling (unless you happen to mention that in the text on that particular page).
- ‘Competitive Engineering’ does not scale. It doesn’t need to.”

[erik.simmons@construx.com](mailto:erik.simmons@construx.com)

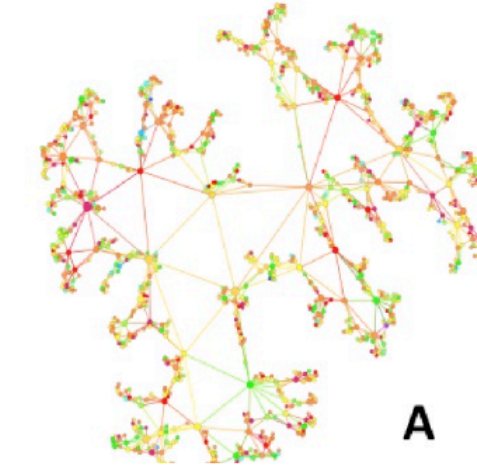
Get a free e-copy of ‘Competitive Engineering’ book.  
<https://www.gilb.com/p/competitive-engineering>



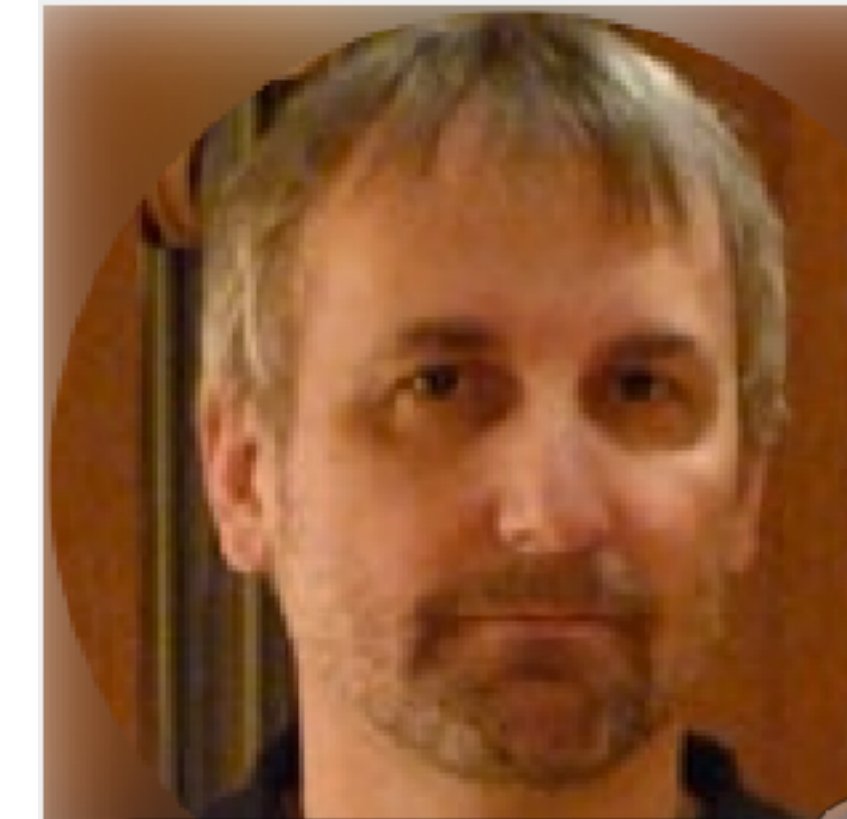
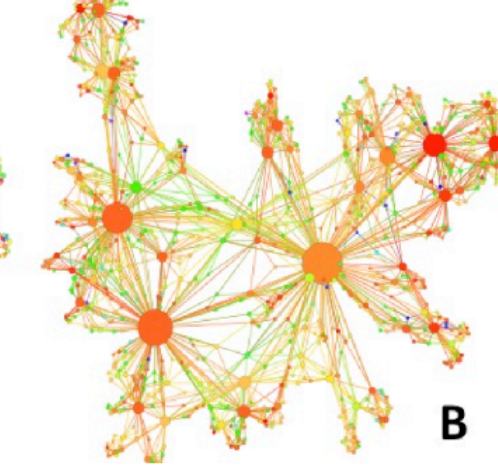
58

Complex Quantum Network Manifold  
In Dimension  $d$

$d=2$   
(Exponential)

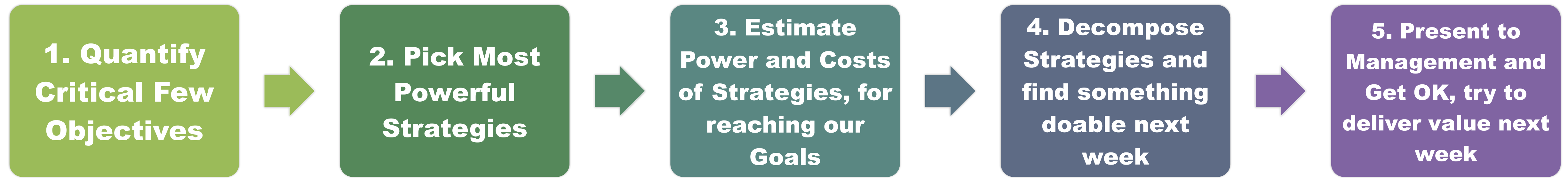


$d=3$   
(Scale-free)



SOURCE: **SCALE-FREE:  
Practical Scaling Methods  
for Industrial Systems Engineering**  
lecture slides  
<http://concepts.gilb.com/dl892>





## **The Evo Agile Startup Week**



Part 4: 'Agile as it should be'

# Starting a larger agile project, Planning Optimisation Week

## DAILY AGENDA

1. Quantify the critical values
2. Draft the best designs to reach the values
3. Build a Table to see if you have pretty good design for the values.
4. And - next part of this book - decompose the designs into weekly do-able value increments
5. Get approval from the 'Powers That Be', to start rolling out results, for r4eal, next week.



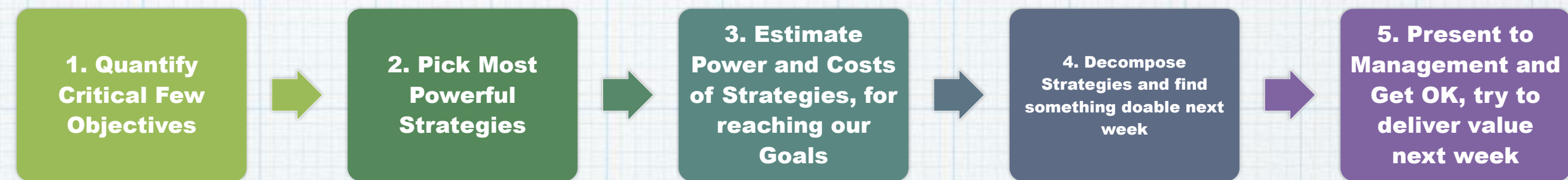


# Top-Level Planning Week

The big idea:  
Plan for a week, then  
start delivering real  
value  
In a prioritised  
Stream

- \* This process can be shortened to 2 days and even 1 day if you need to.

- \* But my experience is that it is then too hectic.
- \* You get what you pay for here.
- \* The full week gives people time to learn, buy in, discuss, argue, and feel pretty good about the proposals.
- \* A week is a small investment to get a big project started better.



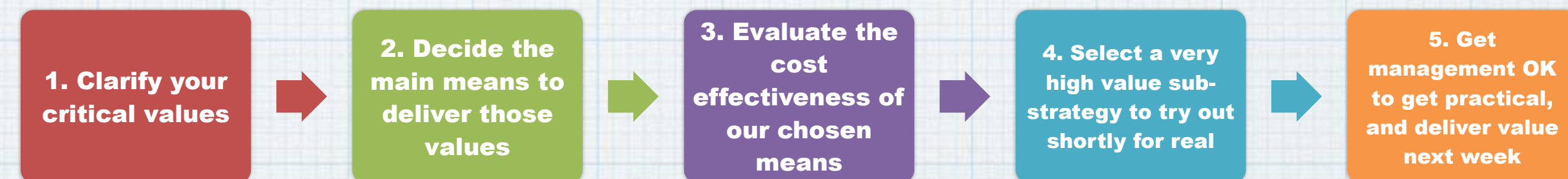
- \* We build a top-level critical model of our project.
  - \* We get a balanced idea of the key values to aim for, and the key constraints to respect.
- \* This top level model, with updates, will become the primary control center for the project.
  - \* It is for the project management level, and all levels they report to.
  - \* The essentials of project control on a one page control panel.

- \* There is only one essential question: are we delivering values as planned, for budgets and deadlines we planned?

- \* We do not use 'yellow stickies': we digitize the planning,
  - \* even just in spreadsheets,
  - \* so we can build on it,
  - \* as we detail the planning,
  - \* and progress the value delivery and learning process.

- \* No 'Infra-structure only',
  - \* just incremental improvements to previous incremental status
    - \* in the plans
    - \* and in the real systems.

## The Planning Week Schedule



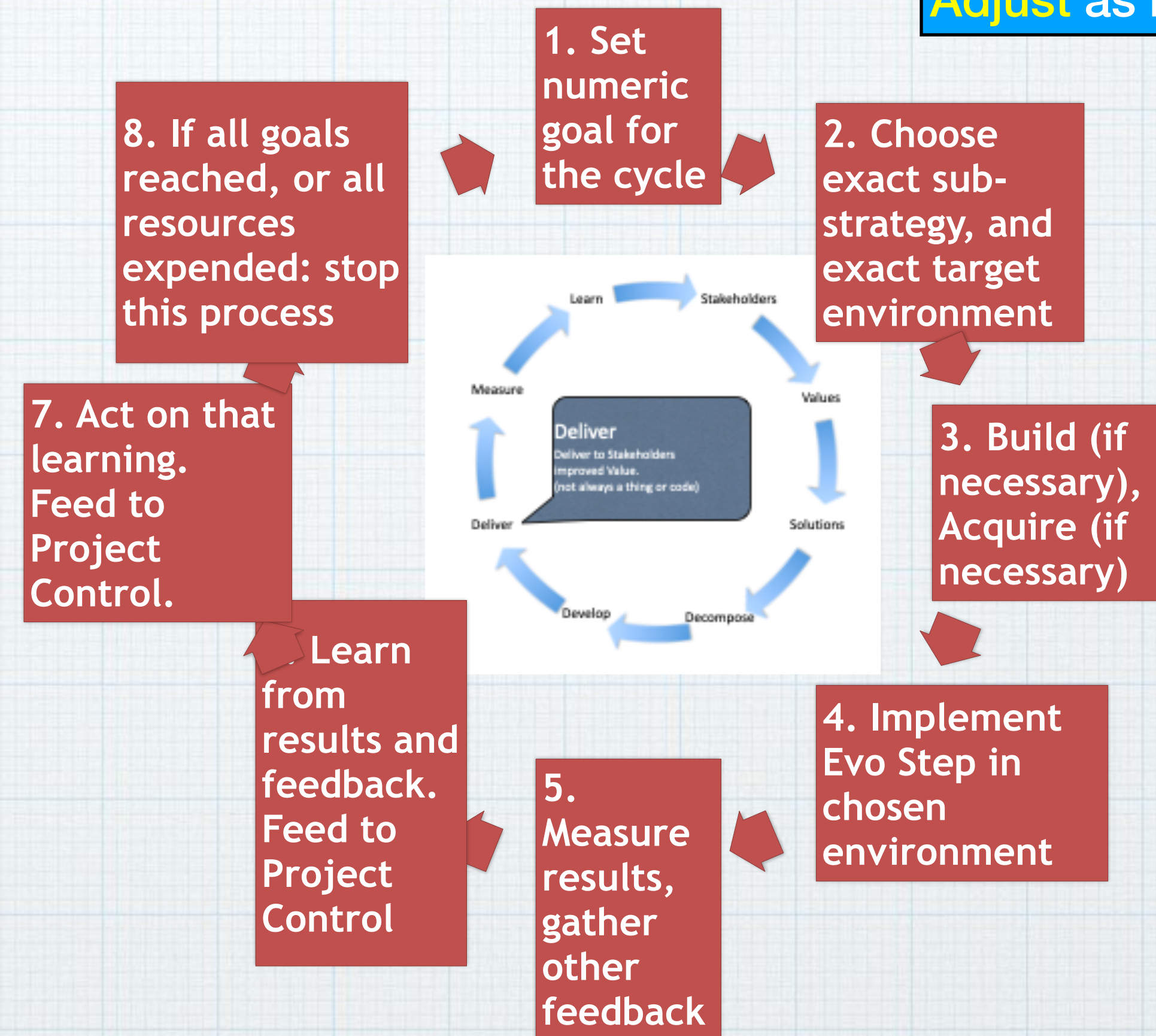
## Purposes of each days tasks



# Learning by Doing

Try new ideas,  
measure their real effects  
In a real system,  
Adjust as needed to succeed.

- \*Participants learn the Planguage methods, on the fly, by doing it.
  - \*No other training necessary.
  - \*But a competent coach is necessary,
    - \*someone who knows what is in this book! You.
- \*I have personally coached 5 real project teams at once in the same week, and repeated the feat 5 different weeks (= 25 projects) at McDonnell-Douglas Aircraft (now part of Boeing), for aircraft design projects.
  - \*They liked the results so much they commissioned me to train their coaches and certify them as competent.
  - \*We always got approval to deliver measurable results from the next week and onwards.
  - \*What manager could resist?
  - \*There are many more case studies of the 'startup week' method.
    - \* (like Ericsson, HP, JP Morgan Bank, DoD)
- \*We do not build prototypes or mockups of our design.
  - \*We test our design ideas by *implementing them*
    - \*on real existing systems:
    - \*but usually on a small scale,
    - \*a week's work,
    - \*before we scale up.
- \*I personally do not trust mockups and prototypes at all.
  - \*Not for large projects.
  - \* I do not believe they give us credible enough information.
  - \*They certainly do not deliver any real value to stakeholders.
- \* Real and small increments cost roughly the same as prototypes and mockups:
  - \*but they deliver much more credible feedback from the real world,
  - \*and above all, they deliver real and measurable value.



'Weekly' 'sprints': Deliver Value and Learn



The big trick in being able to use  
real, small, value-delivery steps,  
on large projects  
is knowing:

- \*How to *decompose design* into small implementable delivery steps (architecture -> sub-designs)
- \*How to safely deliver these small steps to real live existing systems, products, services.
- \*Before you 'get skeptical on me', let me inform you that
  - \*Elon Musk increments real assembly-line production of Tesla cars,
  - \* with average 20 incremental changes (half hardware, half software) weekly.
  - \*And he makes a damned fine vehicle for me, too.
  - \*This is the same method I am talking about. Here
- \* Safest car in the world, one of my 'Very personal' values!

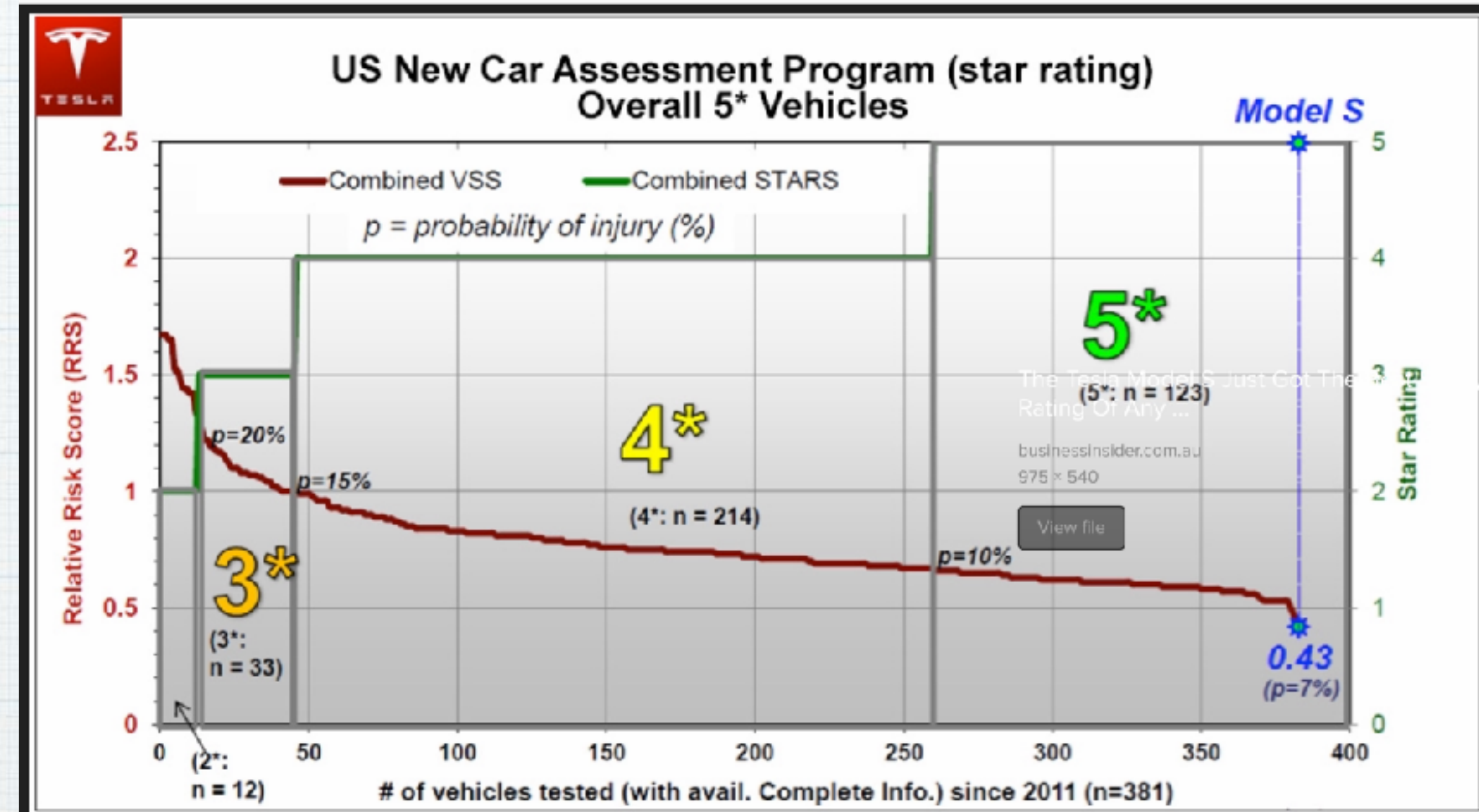


FIGURE: TESLA S SAFETY, DESIGNED IN INCREMENTALLY,  
AND MODEL 3 GOT BETTER.  
NOT BAD FOR A BEGINNER IN CAR INDUSTRY.

The Value Planning Decomposition Chapter  
Here's a link to "Ch 5 Decomposition by Value" in my Dropbox  
<https://tinyurl.com/VPDecomposition>



# WE should NEVER REALLY 'BUILD A NEW SYSTEM' FROM SCRATCH (no matter how radical the vision and architecture)

- \*It is worth mentioning that this (Evo, with POW start) is not a process which always assumes we are starting from scratch.
- \*I have often used it for a major upgrade of *existing* systems, several years old.
  - \*For example the 8 years old US DOD Persinscom system.
- \*I normally can assume that the previous system/product/service is out there, right now, in the field, being used by real people.
- \*I can also assume that the old system badly needs value improvements now, and that is why we are 'starting this project'.
- \*Your project is not, *ever*, to 'create a new system/product/service'.
  - \*The *real* project is always, without exception, to *improve the critical values*, of the 'old' system.
  - \*But this will be a cultural shift for many, and require leadership.
  - \*"Building and spending are not the game, real value delivery is the scoring mechanism!"
- \*We can therefore exploit this reality (of existing systems) for *these* purposes:
- \*As a realistic playground for experiments in design: see how well things *really* work.
- \*As a possibility to actually improve the 'old' system *immediately*, in critical priority areas. Put design to immediately-useful value improvement. Prove you know how to design usefully.
- \*As a major risk management strategy, where we do things in small steps, and get feedback before committing more resources. Big failure is impossible with this method.
- \*Of course there are all kinds of things that are bad and not cost-effective, with the old system.
- \*And there are all kinds of new improved designs that need to be put in place. But these can both be done, in their own time. Perhaps as an increment, and hopefully a cost-effective increment.
- \*But there is no need to do major investments in system *replication*, before proving that you can design for real value quickly, when and where it counts.

US DoD. Persinscom **Impact Estimation Table:**

Design Ideas ->	Designs						Sum Requireme
	Technology Investment	Business Practices	People	Empowerment	Principles of IMA Management	Business Process Re-engineering	
<b>Requirements</b>	50%	100%	5%	5%	5%	60%	185%
Availability	50%		5-10%	0%	0%	200%	265%
90% <-> 99.5% Up time							
Usability			5-10%	50%	0%	10%	130%
200 <-> 60 Requests by Users							
Responsiveness	50%	10%	90%	25%	5%	50%	180%
70% <-> ECP's on time							
Productivity	45%						303%
3:1 Return on Investment	50%						251%
Morale							
72 <-> 60 per month on Sick Leave							
Data Integrity	42%						177%
88% <-> 97% Data Error %							
Technology Adaptability	5%						160%
75% Adapt Technology							
Requirement Adaptability	80%						260%
? <-> 2.6% Adapt to Change							
Resource Adaptability	10%	80%	5%	50%	50%	75%	270%
2.1M <-> ? Resource Change							
Cost Reduction	50%	40%	10%	40%	50%	50%	240%
FADS <-> 30% Total Funding							
Sum of Performance	482%	280%	305%	390%	315%	649%	
Money % of total budget	15%	4%	3%	4%	6%	4%	36%
Time % total work months/year	15%	15%	20%	10%	20%	18%	98%
Sum of Costs	30	19	23	14	26	22	
Performance to Cost Ratio	16:1	14:7	13:3	27:9	12:1	29.5 :1	

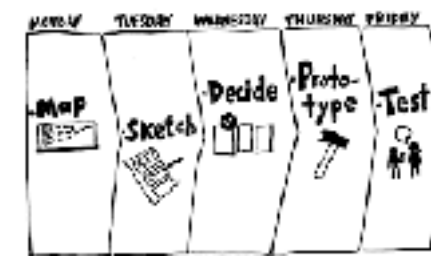
The Unity Method 111111

for decomposition into iterative value delivery steps

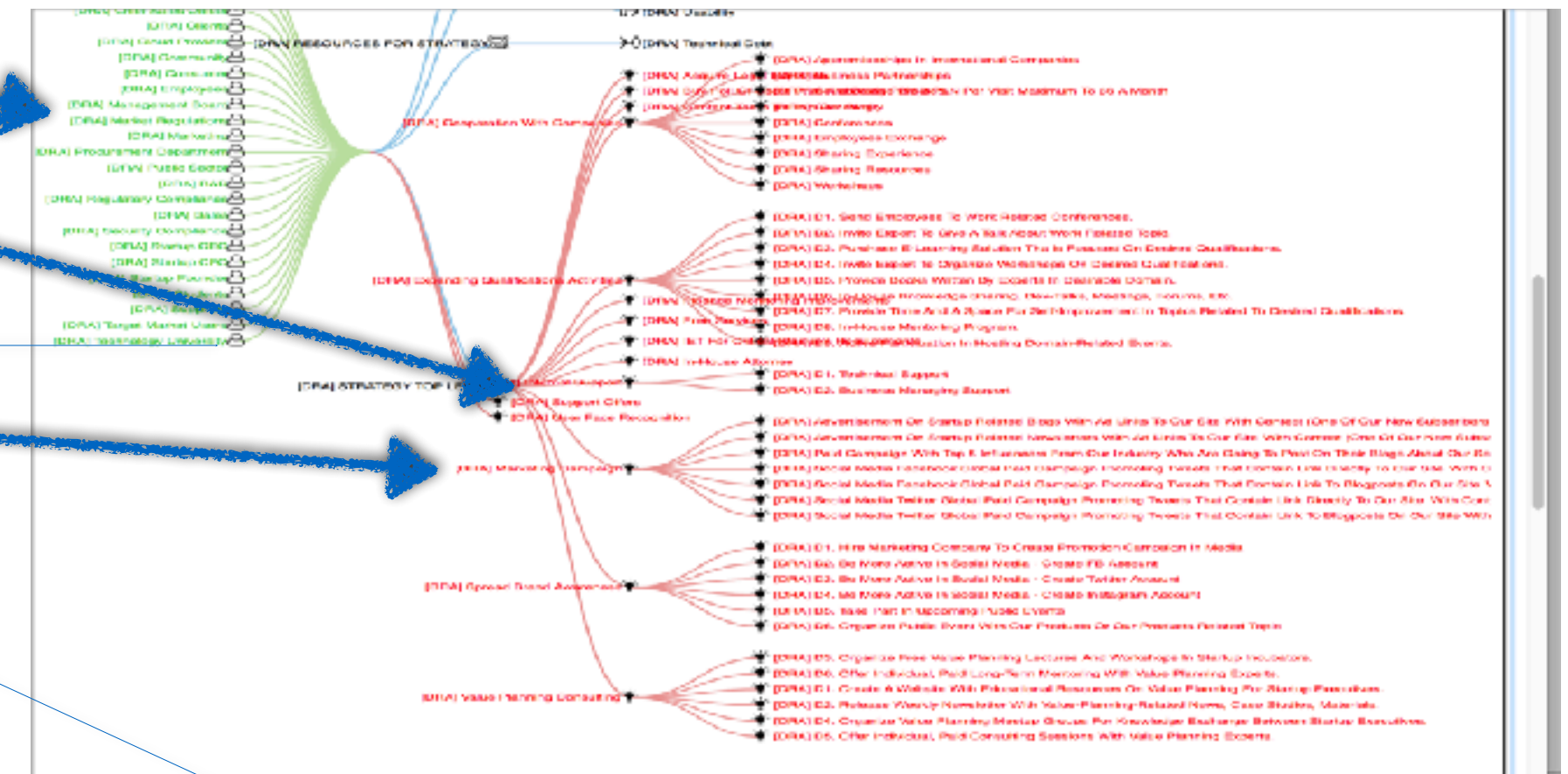
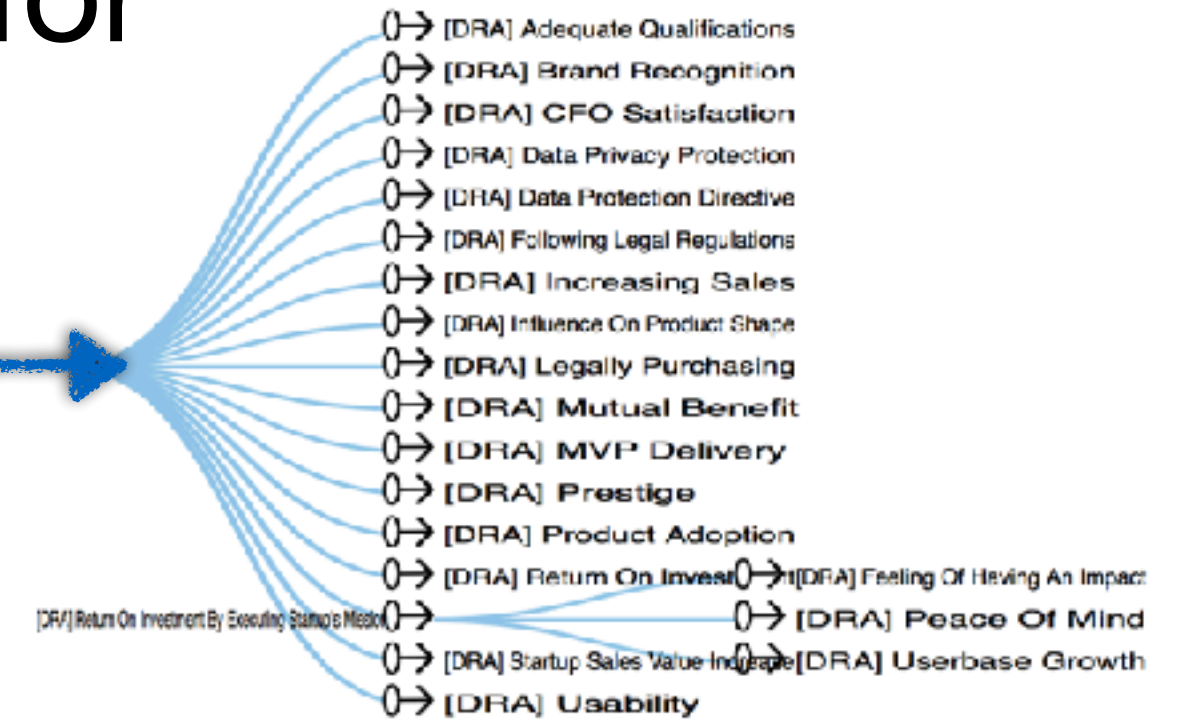
<http://www.gilb.com/DL451>



# An advanced 'Design grownups.



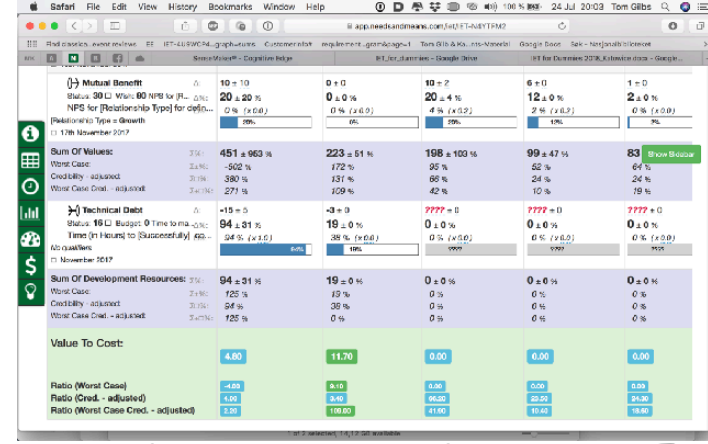
## Sprint' for



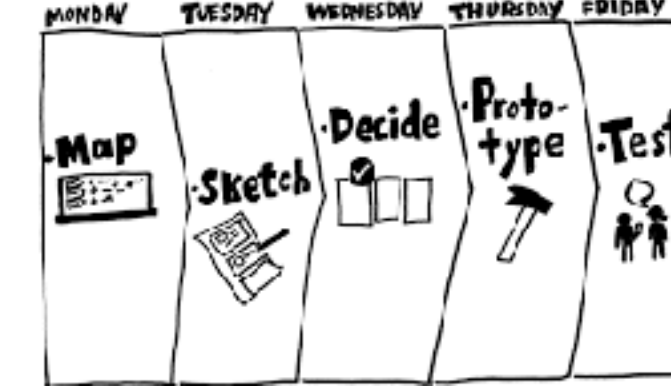
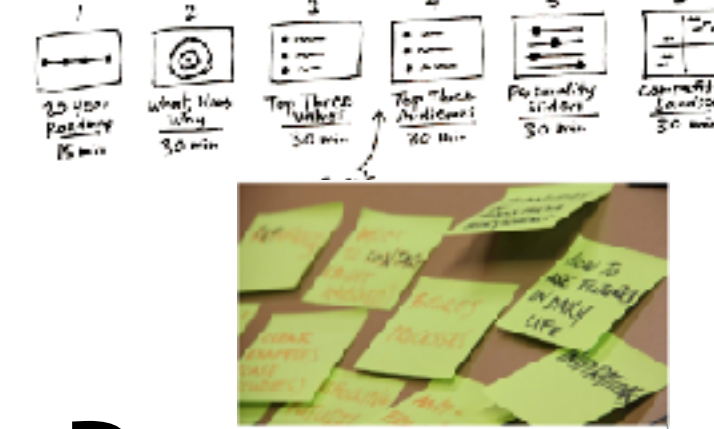
<b>Mutual Benefit</b> Status: 30 <input type="checkbox"/> Worst: 60 NPS for [R...] NPS for [Relationship Type] for defn... Relationship Type = Growth 11/11/2017		10 ± 10	0 ± 0	10 ± 2	6 ± 0	1 ± 0
		20 ± 20 %	0 ± 0 %	20 ± 4 %	12 ± 0 %	2 ± 0 %
		0 % (x 0.0)	0 % (x 0.0)	4 % (x 0.2)	2 % (x 0.2)	0 % (x 0.0)
		99%	0%	99%	10%	3%
Sum Of Values:		451 ± 963 %	223 ± 51 %	198 ± 103 %	99 ± 47 %	83
Worst Case:		-502 %	172 %	95 %	52 %	64 %
Credibility - adjusted:		31 %	131 %	66 %	24 %	24 %
Worst Case Cred. - adjusted:		271 %	109 %	42 %	10 %	19 %
<b>Technical Debt</b> Status: 16 <input type="checkbox"/> Budget: 0 Time to ma... Time (in Hours) to [Successfully] ... No qualWers 11/11/2017		1 ± 5	-3 ± 0	???? ± 0	???? ± 0	???? ± 0
		31 %	19 ± 0 %	0 ± 0 %	0 ± 0 %	0 ± 0 %
		94 % (x 1.0)	38 % (x 0.0)	0 % (x 0.0)	0 % (x 0.0)	0 % (x 0.0)
		60%	100%	99%	99%	99%
Sum Of Development Resources:		94 ± 31 %	19 ± 0 %	0 ± 0 %	0 ± 0 %	0 ± 0 %
Worst Case:		125 %	19 %	0 %	0 %	0 %
Credibility - adjusted:		31 %	39 %	0 %	0 %	0 %
Worst Case Cred. - adjusted:		125 %	0 %	0 %	0 %	0 %
Value To Cost:		4.60	11.70	0.00	0.00	0.00
Ratio (Worst Case)		-4.09	9.10	0.00	0.00	0.00
Ratio (Cred. - adjusted)		1.00	3.40	0.00	0.00	0.00



# Project Startup versus Design Sprint



Planguage  
Evo



- Engineering Based
- Systems Applicable (UX)
- All Values Quantified
- Risk Mgt ( $\pm$ .Cred, Prty)
- Scale-Free
- Decades of Experience
- Research Published: HP
- Many publ.Case Studies
- AI Prioritization Val/€
- Design estimates V&€
- Actual incr. measures
- Digital Planning Long Term

- Programming Craft
- Software and UI Limited
- Values Not Quantified
- No Explicit Risk Mgt.
- Not proven large scale
- Hot new idea
- No known research
- Can't find cases, yet
- Role player decides priority
- No estimates
- Dodgy Prototype
- Yellow Sticky Culture





# Design Sprint ‘Claimed Benefits’ <-Jake (of course YOU are skeptical, and know this.)



## “8 incredible Design Sprint benefits for your business”

“Here are the 8 amazing Design Sprint benefits you get in your business by employing this methodology of [Google](#):

### 1. Design Sprint helps you save time and money

Design Sprint is designed to work quickly and intensely to get a solution to a business problem through design.

By using Design Sprint you reduce the time you spend on the design process and the process of defining your product, going from months to days

This is a great benefit because you save a lot of time and money and allows you to define a validation plan based on the feedback from your users.

### 2. Design Sprint Quickly Reduces Product Development Cycles

Derived from the above, development times are dramatically reduced, as Design Sprint work on a connecting problem with the solution. This helps you to test whether an idea works or not, without developing products with very long production cycles (Idea, Design, Approve, Develop, Launch and Validate).

With the Design Sprint you become a more agile organization

Before investing in the development of your product or a new functionality that requires an expensive process you can dedicate 5 days so that the team understands the problem that your company is facing, designing the solutions, creating a functional prototype and validating your ideas in a matter of hours. Becoming a more agile organization.

### 3. Real feedback with Design Sprint

Knowing the feedback of your product is fundamental to developing successful products. Many times when we get this information is when we have finished the project.

With the Design Sprint, you know firsthand and quickly the real feedback from your customers. This feedback is crucial because it helps you improve your product or service at the same time you design it

On the other hand, your team is actively working on the process, as the production cycle involves different sources of information within your organization.

### 4. Validate your business ideas with Design Sprint

Without validation, it is difficult for ideas and products to work. That is precisely what you will do on the last day of the Sprint in a very concrete way.

Through Design Sprint you can design the validation plan of the business idea or functionality of your product

Being clear how the process will be, the time you are going to invest and the type of results with which we can continue the process of transferring your product to the market.

### 5. Generates business and innovation.

Design Sprint gives your team a way of working to solve complex problems in a week.

So you can achieve a new approach to the project that would have taken months, even years

### 6. Align expectations with your team

Making all departments share knowledge, needs, and strategy so that the result is a solution that satisfies and meets needs.

Being able to make your step to deploy is a cycle of continuous product integration

### 7. Help you measure

The sprint design uses measurement processes in the different phases that the methodology uses.

What allows you to measure the results obtained at the end of the process, as well as the impact of the same on your business and on the equipment and surplus generated during the process

### 8. An agile and fast methodology that you can apply to your business

Once you internalize the Design Sprint methodology you can use it and coordinate it with other processes that you already have established in your project or business.

Typically, the first time you make a Sprint Design is tiring and difficult.

We recommend that you count with the help of a Sprint Master Certified to achieve these incredible results”

## Skeptical Observations <-TSG

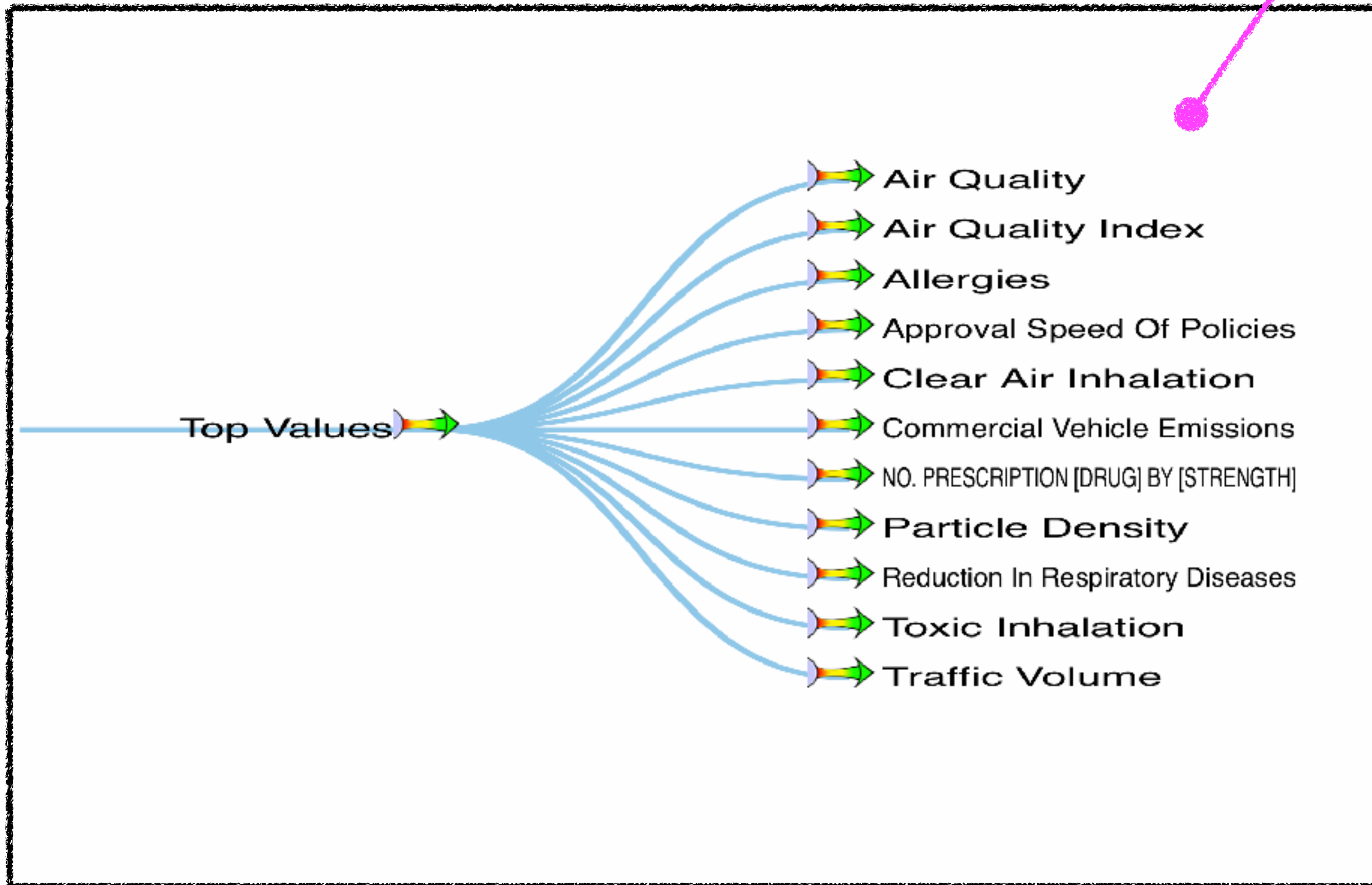
- These claims are made by a **seller** of ‘Design Sprint’ training and certification service ([letshackity.com](https://letshackity.com))
- Most of the terms and concepts have **poor definition**, and are *highly ambiguous* (examples)
  - Design, Align Expectations, Investing (Product Dev), Complex Problems, measure the results, agile methodology, validation, and many more.
- Not one single number is offered to indicate the magnitude of improvements
- No clear baseline (who is going to get improved) is indicated
- No references to real case studies with results, costs, problems
- No comparison with any other known methods
- No links or references to anything
- Lots of causal assertions, none proven
- “This feedback is crucial because it helps you improve your product or service at the same time you design it”
- No indication or example of the types and magnitude of the costs for the individual, the project, and the organization for learning and maintaining the Design Sprint method
- No glowing references from real people or customers
- No information about how things went after the first week, to tell us how good or bad the week was.
- Constant implication: Google is successful, therefore this method is good



# Detailed Examples Of Evo Agile Startup Week

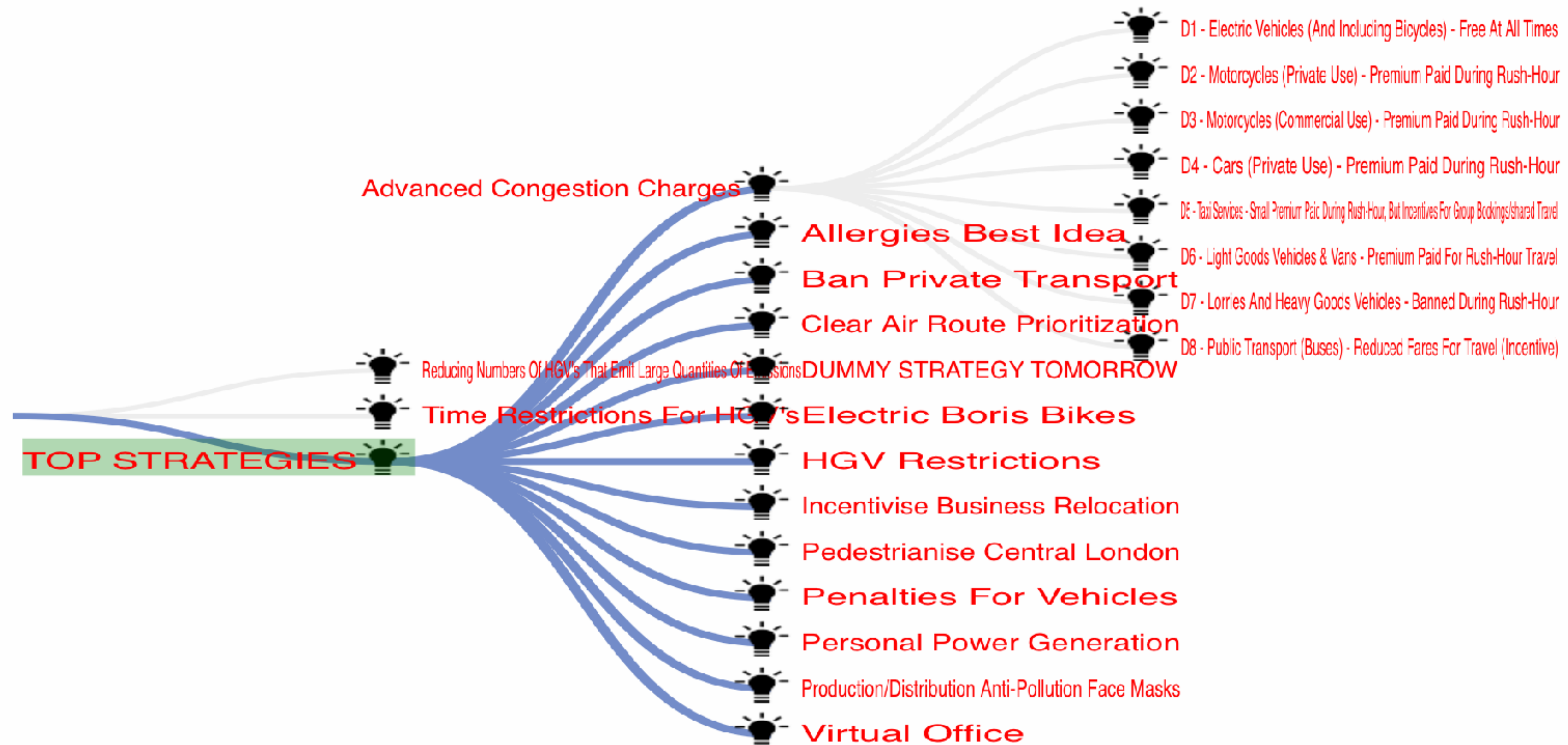


# Day 1: The Top Ten (or 11) Critical Stakeholder Values Quantified on a Page





## Day 2: The Top-Ten Best Designs: the architecture to deliver the values





Day 3:  
Value Table:  
estimate how  
cost-effective  
your designs  
are

- See next slide
- For
- Simplification
- Priority Design
- Bar Chart

		💡 HGV Restrictions	💡 Clear Air Route P...	💡 Advanced Congesti...	💡 Penalties For V...
<b>Requirements</b>					
<b>Air Quality Index</b> =; Past: 135 → Wish: 67 µg/m³ Δ%:		-7%	0%	74%	88%
<b>Air Quality</b> =; Status: 9.5k → Goal: 150 People Δ%:		0%	0%	37%	43%
<b>Allergies</b> =; Status: 10 → Wish: 1 number of ... Δ%:		-56%	33%	44%	22%
<b>Approval Speed Of Policies:</b> Status: 6 → Goal: 3 Mnths Δ%:		-33%	0%	0%	0%
<b>NO. PRESCRIPTION [DRUG] BY ...</b> Status: 1k → Wish: 100 NUMBER Δ%:		0%	11%	39%	50%
<b>Clear Air Inhalation</b> =; Status: 20 → Wish: 70 % Δ%:		4%	40%	36%	50%
<b>Particle Density</b> =; Status: 1k → Wish: 300 Number of Δ%:		0%	0%	54%	50%
<b>Reduction In Respiratory Di...</b> Status: 1k → Wish: 100 PATIENTS Δ%:		0%	17%	17%	78%
<b>Toxic Inhalation</b> =; Status: 100 → Wish: 10 Max Mg Po Δ%:		-1%	0%	61%	50%
<b>Sum Of Values:</b> Σ%:		-93 %	101 %	362 %	431 %
<b>LABOUR EFFORT</b> =; Status: 0 → Budget: 1k WORK MONTHS Δ%:		10%	10%	30%	30%
<b>£ CAPITAL COSTS</b> =; Status: 0 → Budget: 1m Δ%:		3%	30%	30%	

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

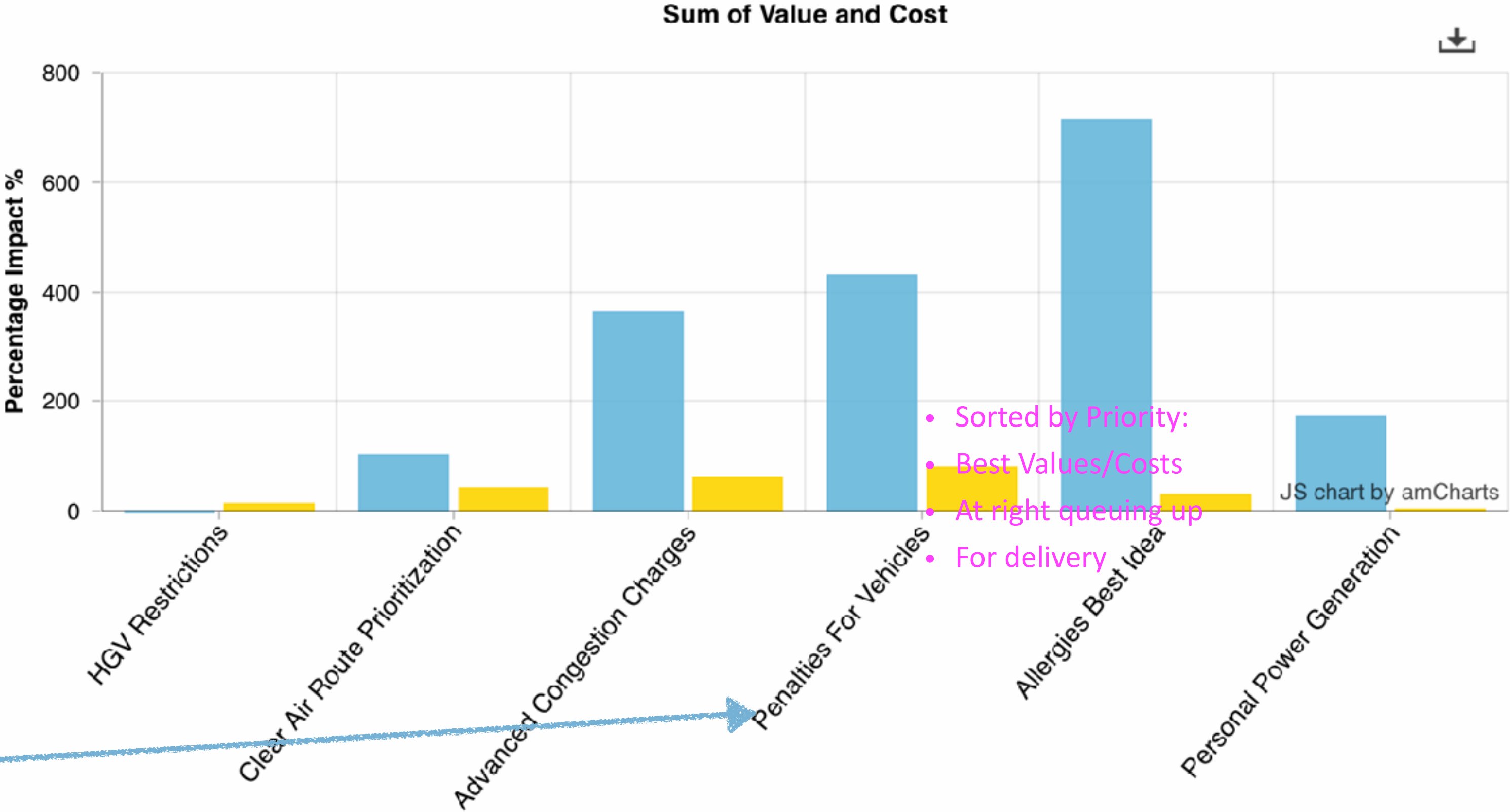




Day 3:

# Value Table: estimate how cost-effective your designs are

Requirements	HGV Restrictions	Clear Air Route P...	Advanced Congesti...	Penalties For Vi
<div><div><div></div><div>Air Quality Index</div></div><div>Past: 135 → Wish: 67 µg/m³</div><div>Δ%:</div></div>	<div><div></div><div>-7%</div></div>	<div><div></div><div>0%</div></div>	<div><div></div><div>74%</div></div>	<div><div></div><div>68%</div></div>
<div><div><div></div><div>Air Quality</div></div><div>Status: 9.5k → Goal: 150 People</div><div>Δ%:</div></div>	<div><div></div><div>0%</div></div>	<div><div></div><div>0%</div></div>	<div><div></div><div>37%</div></div>	<div><div></div><div>43%</div></div>
<div><div><div></div><div>Allergies</div></div><div>Status: 10 → Wish: 1 number of ...</div><div>Δ%:</div></div>	<div><div></div><div>-56%</div></div>	<div><div></div><div>33%</div></div>	<div><div></div><div>44%</div></div>	<div><div></div><div>22%</div></div>
<div><div><div></div><div>Approval Speed Of Policies:</div></div><div>Status: 6 → Goal: 3 Mnths</div><div>Δ%:</div></div>	<div><div></div><div>-33%</div></div>	<div><div></div><div>0%</div></div>	<div><div></div><div>0%</div></div>	<div><div></div><div>0%</div></div>
<div><div><div></div><div>NO. PRESCRIPTION [DRUG] BY ...</div></div><div>Status: 1k → Wish: 100 NUMBER</div><div>Δ%:</div></div>	<div><div></div><div>0%</div></div>	<div><div></div><div>11%</div></div>	<div><div></div><div>39%</div></div>	<div><div></div><div>50%</div></div>
<div><div><div></div><div>Clear Air Inhalation</div></div><div>Status: 20 → Wish: 70 %</div><div>Δ%:</div></div>	<div><div></div><div>4%</div></div>	<div><div></div><div>40%</div></div>	<div><div></div><div>36%</div></div>	<div><div></div><div>50%</div></div>
<div><div><div></div><div>Particle Density</div></div><div>Status: 1k → Wish: 300 Number of ...</div><div>Δ%:</div></div>	<div><div></div><div>0%</div></div>	<div><div></div><div>0%</div></div>	<div><div></div><div>54%</div></div>	<div><div></div><div>50%</div></div>
<div><div><div></div><div>Reduction In Respiratory Di...</div></div><div>Status: 1k → Wish: 100 PATIENTS</div><div>Δ%:</div></div>	<div><div></div><div>0%</div></div>	<div><div></div><div>17%</div></div>	<div><div></div><div>17%</div></div>	<div><div></div><div>78%</div></div>
<div><div><div></div><div>Toxic Inhalation</div></div><div>Status: 100 → Wish: 10 Max Mg Pol...</div><div>Δ%:</div></div>	<div><div></div><div>-1%</div></div>	<div><div></div><div>0%</div></div>	<div><div></div><div>61%</div></div>	<div><div></div><div>50%</div></div>
Sum Of Values:	Σ%: -93 %	101 %	362 %	431 %
<div><div><div></div><div>LABOUR EFFORT</div></div><div>Status: 0 → Budget: 1k WORK MONTHS</div><div>Δ%:</div></div>	<div><div></div><div>10%</div></div>	<div><div></div><div>10%</div></div>	<div><div></div><div>30%</div></div>	<div><div></div><div>30%</div></div>
<div><div><div></div><div>£ CAPITAL COSTS</div></div><div>Status: 0 → Budget: 1m</div><div>Δ%:</div></div>	<div><div></div><div>3%</div></div>	<div><div></div><div>30%</div></div>	<div><div></div><div>30%</div></div>	<div><div></div><div>30%</div></div>



Sum Of Value (Estimated)

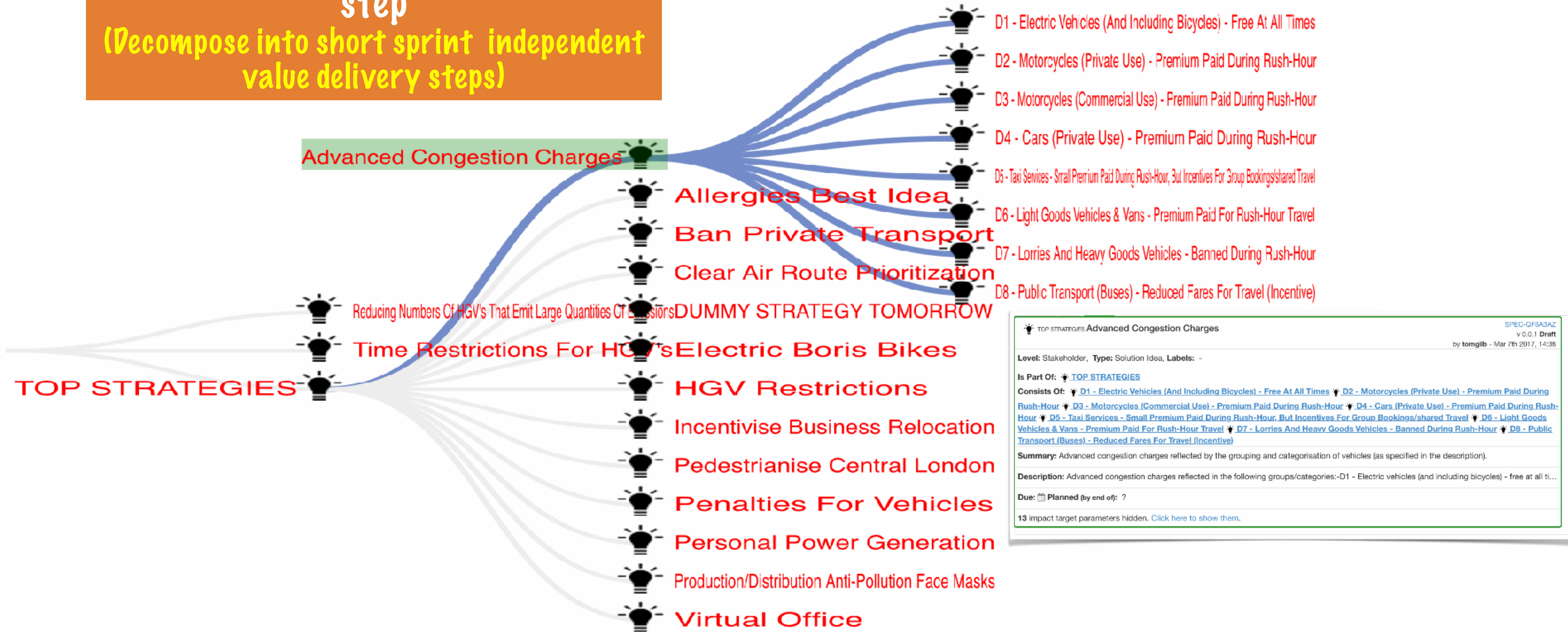
Sum Of Cost (Estimated)

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





Day 4:  
Identify next weeks value-delivery  
step  
(Decompose into short sprint independent  
value delivery steps)





Day 4:

Identify next weeks value-delivery step.

Sort the 'sprint sized' value delivery designs by values/costs delivery priority

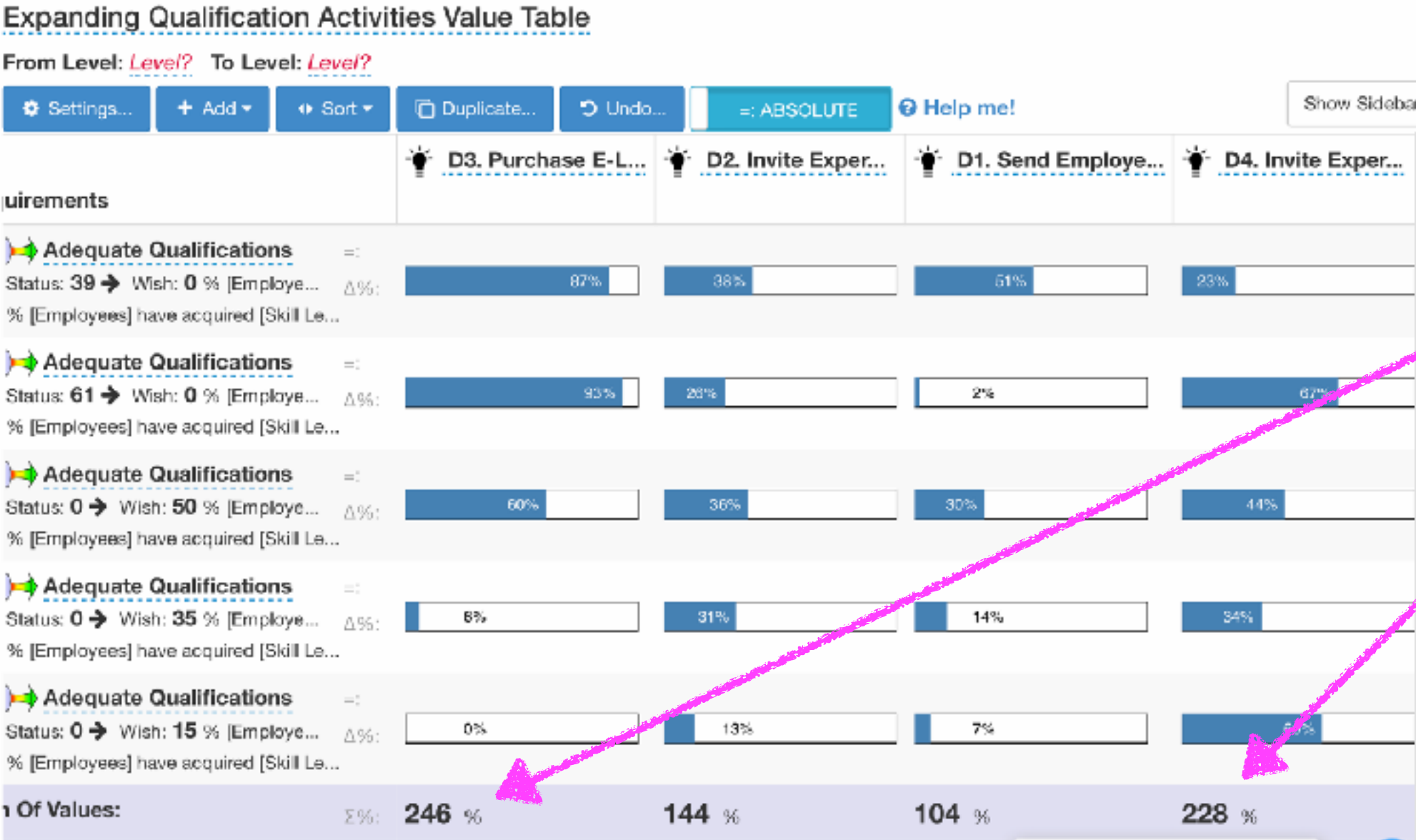


FIGURE: HERE, FROM ANOTHER PLAN, IS A VALUE TABLE FOR DECIDING WHICH ONES OF THE SUB-DESIGNS ARE TO BE PRIORITIZED NEAR TERM (SOURCE POLISH EXPOR PLAN)

- 

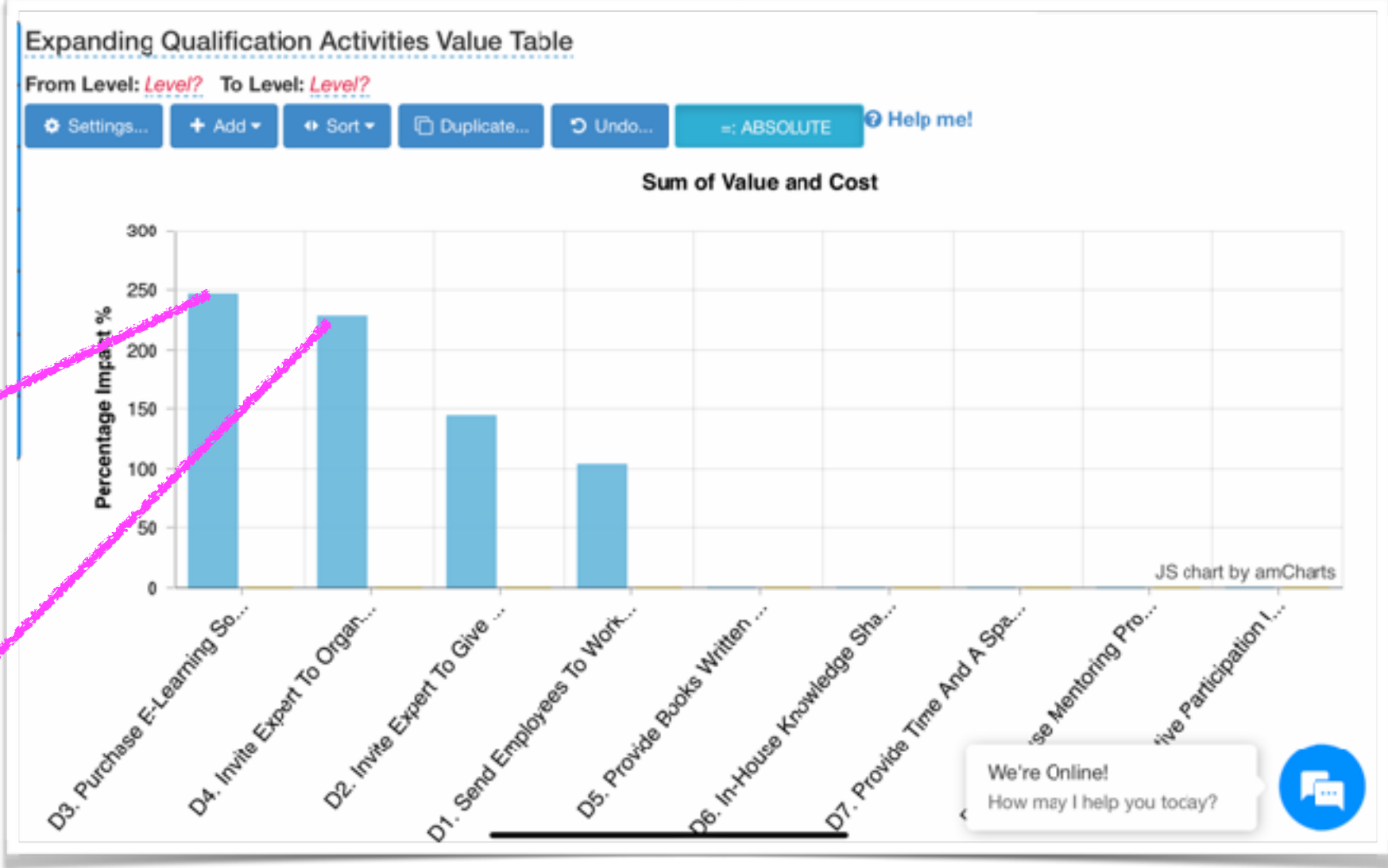


FIGURE: THIS BAR CHART IS EXTRACTED FROM THE TABLE AT LEFT, WE ASKED VALPLAN.NET TO SORT BY IMPACT TOTAL ON ALL VALUE REQUIREMENTS. LEFT-SIDE IS HEAD OF VALUE DELIVERY QUEUE THIS IS 'AUTOMATIC PRIORITIZATION OF DESIGN'. (SOURCE POLISH EXPOR PLAN)



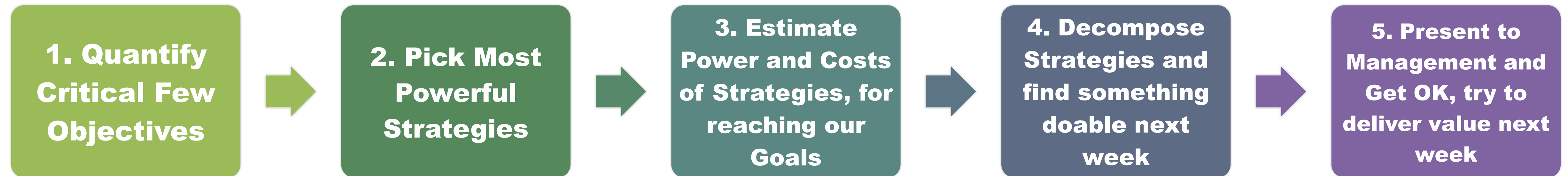
## Day 5: Present Plans to Management, ask for approval to deliver the value.

- “Sub-Design D3 gives best overall stakeholder value delivery
- And takes 1 sprint week
- Shall we follow this value-delivery process?
  - Weekly ?
- Would you like a weekly report on incremental value delivery?
- Or would you prefer to look at costs and risks too?”





# Evo Startup Week: Formal Process



**VALUES?**

**SOLUTIONS !**

**ESTIMATES**

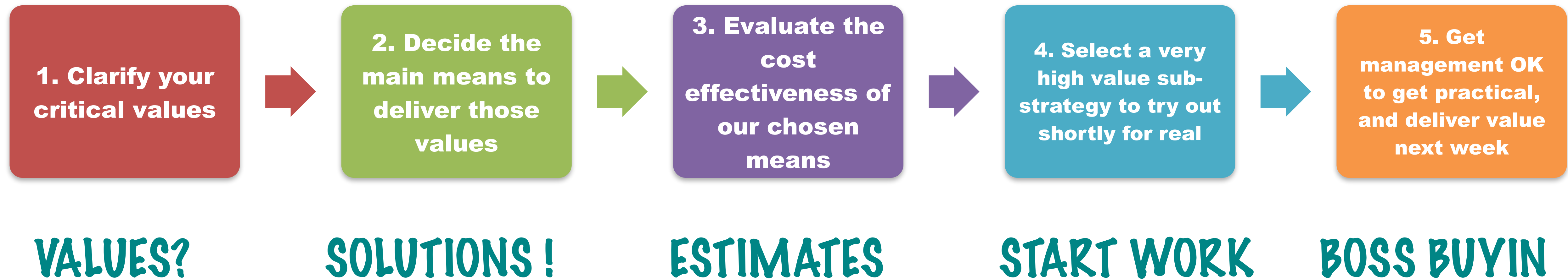
**START WORK**

**BOSS BUYIN**



# Evo Startup Week:

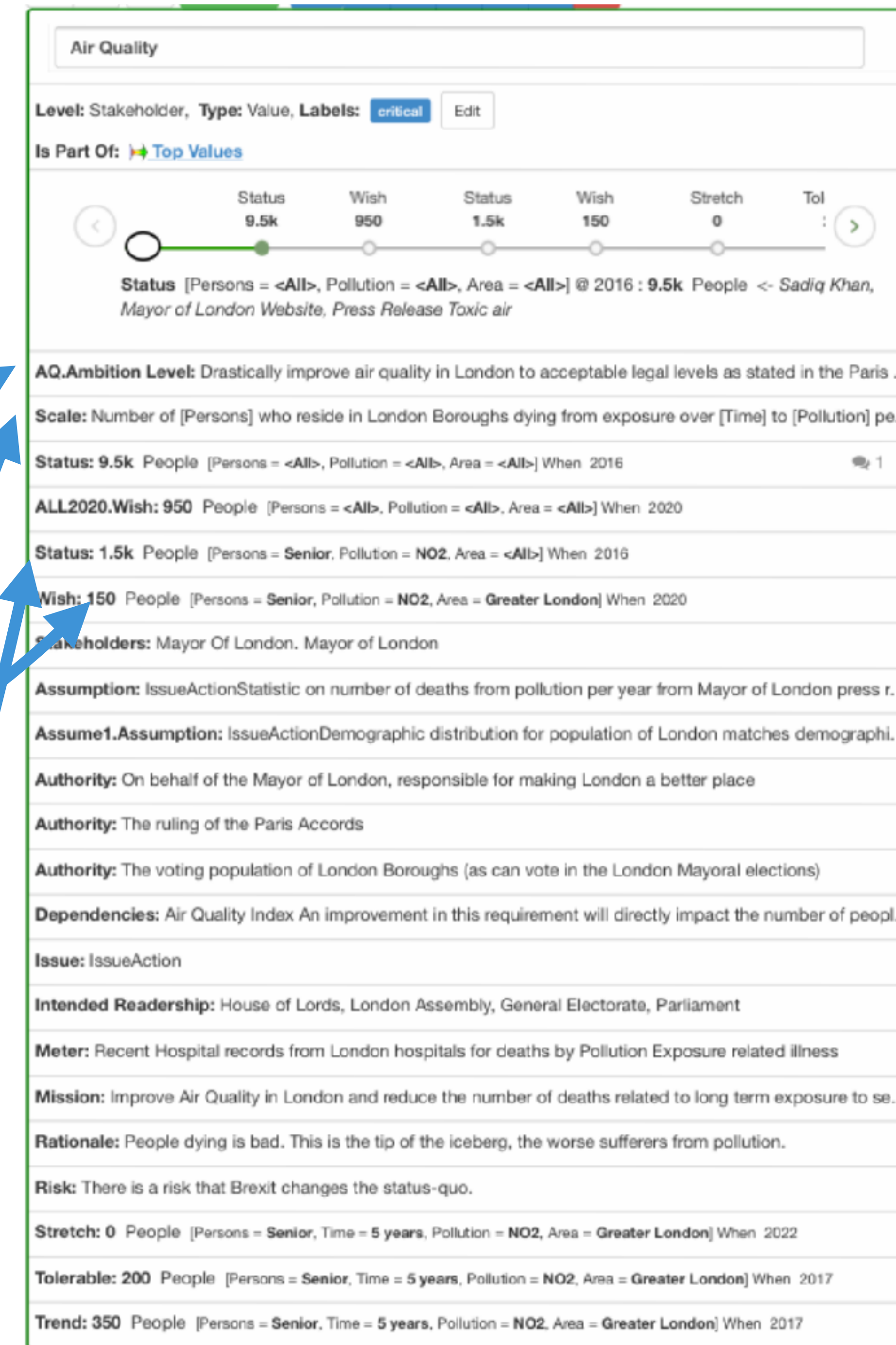
What is behind the process steps?  
Why are we doing this set of steps ?





# Every Monday: REQUIREMENTS

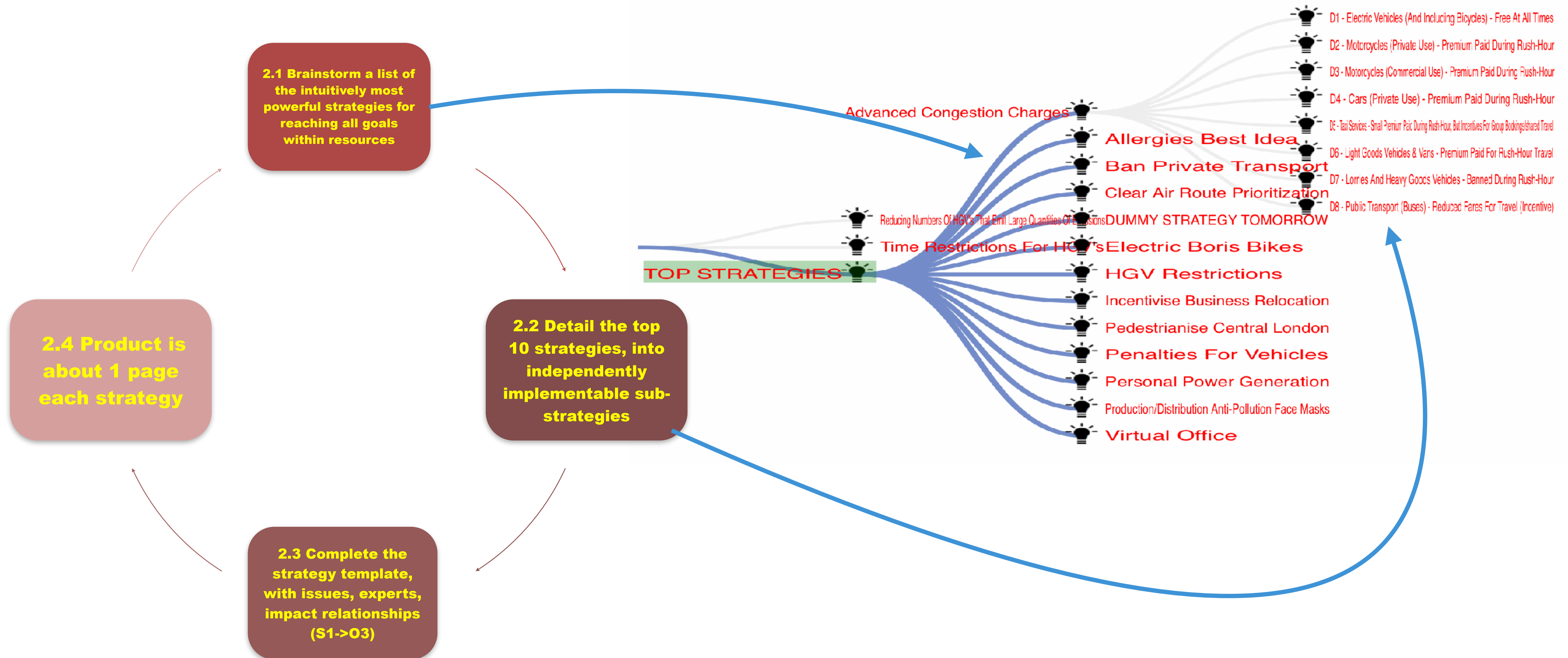
## Set this cycle's Goals





# Tuesday: ARCHITECTURE

## Identify Most-Effective Strategies





# Wednesday: Sanity Check

## Build 'Impact Estimation Table'



Requirements	<input type="checkbox"/> Incentivise	<input type="checkbox"/> Tea Kiosk	<input type="checkbox"/> Daily Danger Checks	Su
<b>(→) Project Timeliness</b> Status: 10 → Wish: 5 % % time overrun necessary to deliver [Project Cost Size = { Medium (\$10k - ... ) }] 30th June 2017	8 ± 0 -2 % 40 ± 0 % 32 % (x 0.8) 40%	5 ± 1 -5 % 100 ± 20 % 50 % (x 0.5) 100%	15 ± 8 5 % -100 ± 160 % -80 % (x 0.8) -100%	
<b>(→) Building Security</b> Status: 50 → Wish: 10 % I... % of [Emergency Types] which in fac... [Emergency Types = { Earthquake }, ...] 30th June 2018	50 ± 0 0 % Injury 0 ± 0 % 0 % (x 0.0) 0%	50 ± 0 0 % Injury 0 ± NaN % 0 % (x 0.6) 0%	30 ± 10 -20 % Injury 50 ± 25 % 15 % (x 0.3) 50%	
<b>(→) User Productivity</b> Status: 15 → Wish: 5 minutes number of minutes for a [user] to co... [user = { adult }, ...] 30th June 2017	10 ± 0 -5 minutes 50 ± 0 % 0 % (x 0.0) 50%	8 ± 3 -7 minutes 70 ± 30 % 56 % (x 0.8) 70%	15 ± 0 0 minutes 0 ± 0 % 0 % (x 0.0) 0%	
<b>Sum Of Values:</b> Credibility - adjusted:	Σ%: 90 ± 0 % Σ?%: 32 %	170 ± 50 % 106 %	-50 ± 185 % -65 %	
<b>(→) Method Implementation Cost</b> Status: 0 → Budget: 3m \$ Total monetary cost in US Dollars, fo... [Project Cost Size = { }] 30th June 2017	500k ± 0 500k \$ 17 ± 0 % 34 % (x 0.0) 17%	2m ± 0 2m \$ 67 ± 0 % 134 % (x 0.0) 67%	1m ± 0 1m \$ 33 ± 0 % 66 % (x 0.0) 33%	
<b>Sum Of Development Resources</b> Credibility - adjusted:	Σ%: 17 ± 0 % Σ?%: 34 %	67 ± 0 % 134 %	33 ± 0 % 66 %	
<b>Value To Cost:</b>				

**Selected Impact Target**

Row: **User Productivity**  
Col: **Tea Kiosk**

Scale: number of minutes for a [user] to complete a [task]

**Value Impact:** Change...

Estimate: minutes  
Δ -7 ± 3

Actual: minutes  
Δ scale val ± 0

Credibility: 0.8

In-house measurements of design / strategy correlate to external sources

**Evidence:**  
we have used tea kiosks and several competitors have which save about seven minutes for users

**Source:**  
[https://www.tripadvisor.com/ShowUserReviews-g154995-d4871495-r475327934-McDonald\\_s-London\\_Ontario.html](https://www.tripadvisor.com/ShowUserReviews-g154995-d4871495-r475327934-McDonald_s-London_Ontario.html)

Add Comment...



# Thursday: MAKE VALUE HAPPEN NEXT WEEK

Find what we can deliver next week

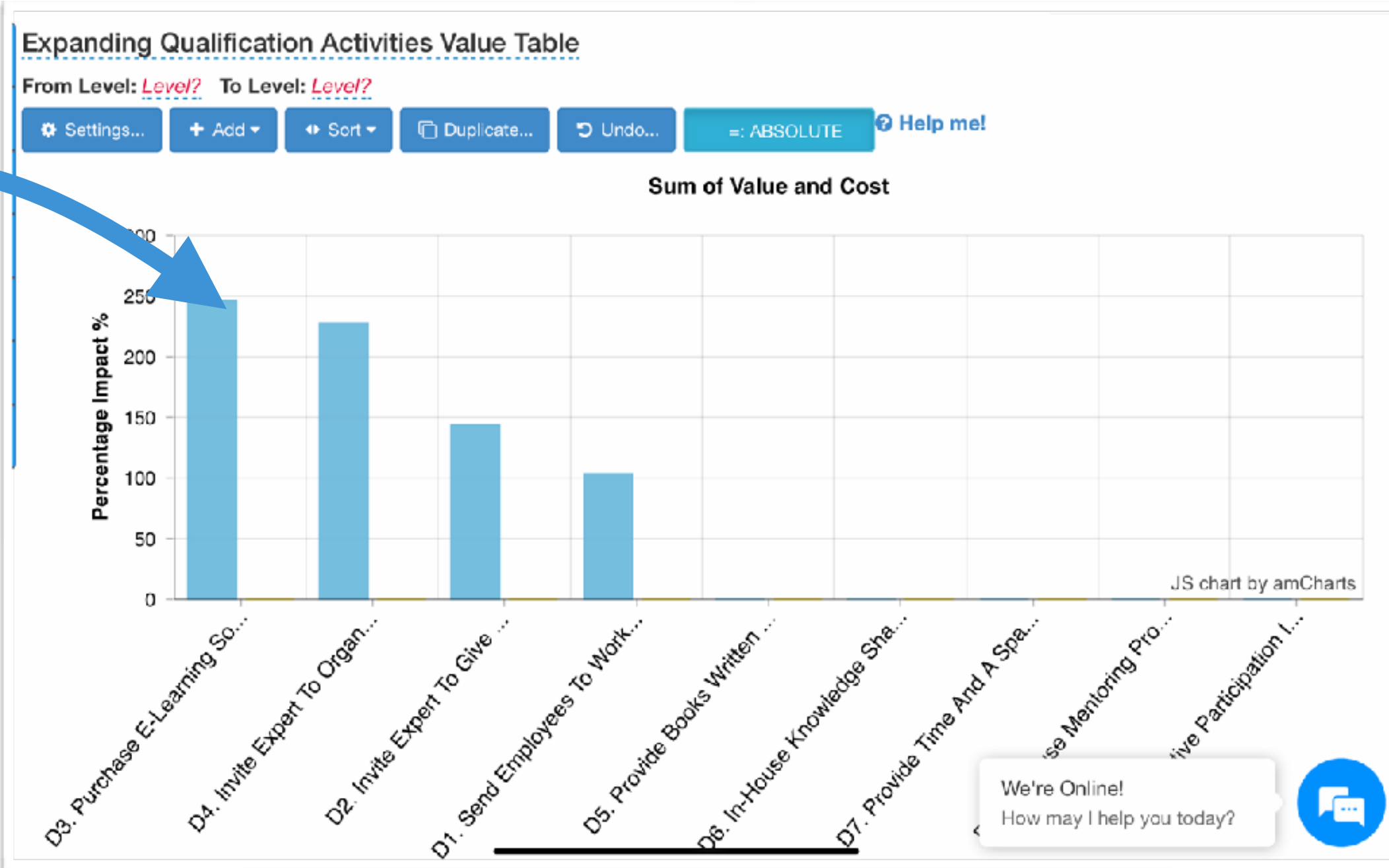
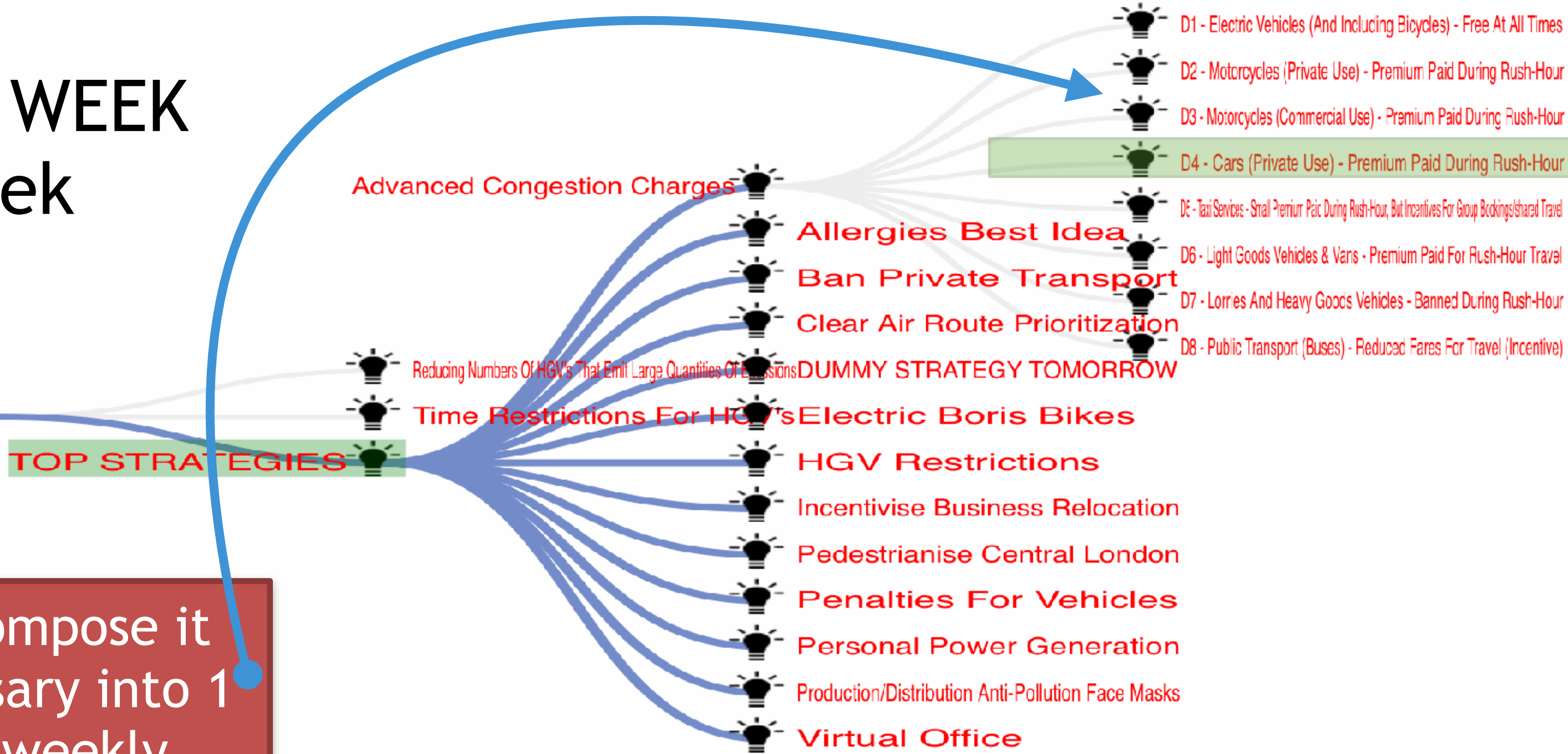
4.5 Option:  
several parallel  
deliveries,  
parallel teams

4.4 Agree to one  
value delivery  
next week

4.1 Look at  
most values/  
costs strategy

4.2 Decompose it  
if necessary into 1  
or more weekly  
implementations

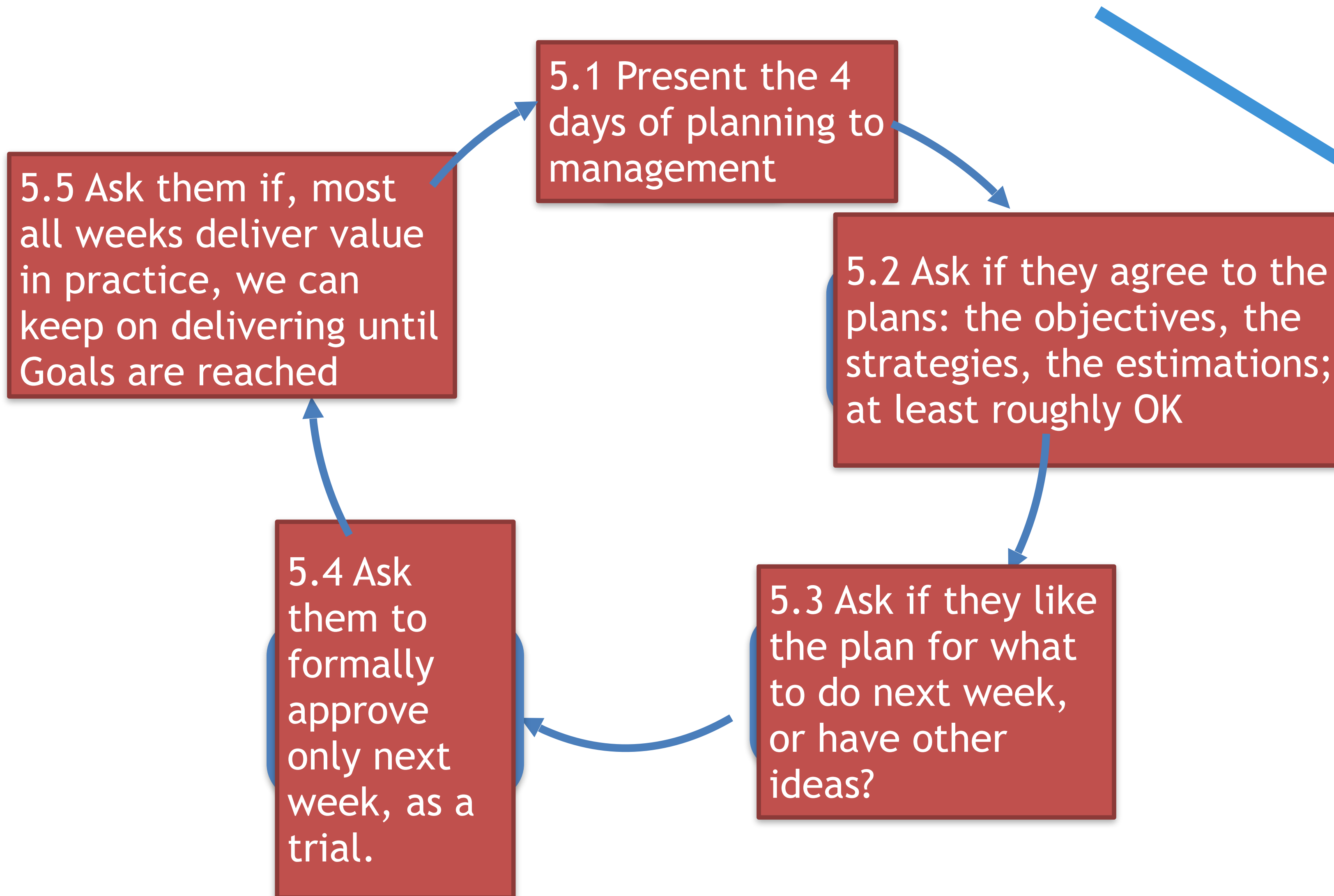
4.3 Estimate which  
one of several  
options would give  
best effect





# Friday : GET THE BOSS ON BOARD

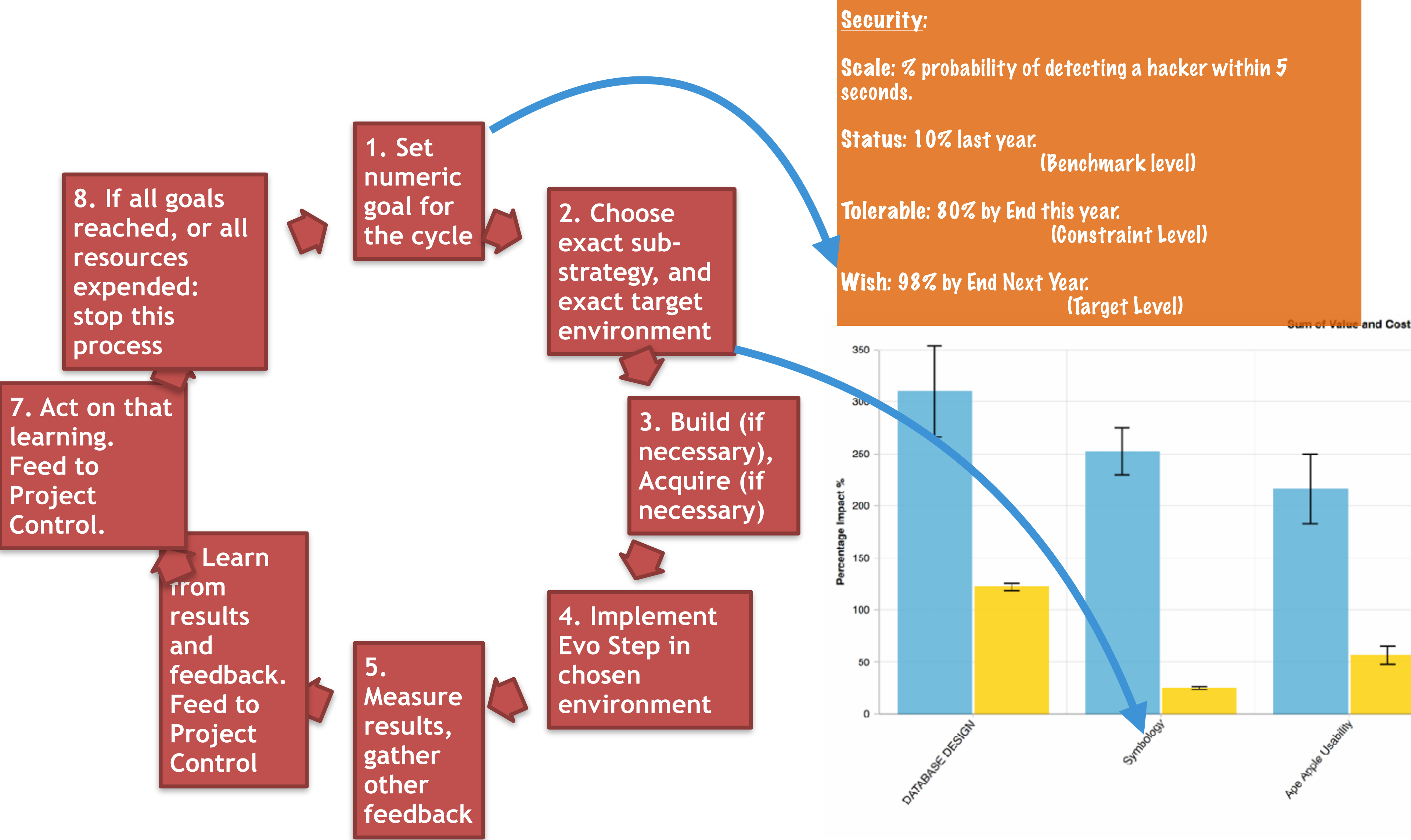
Get Management Approval to try to deliver real measurable value next week



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 second		
	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		



# Evo Weekly Cycle after Startup Week (week 2, 3, 4, ... n)

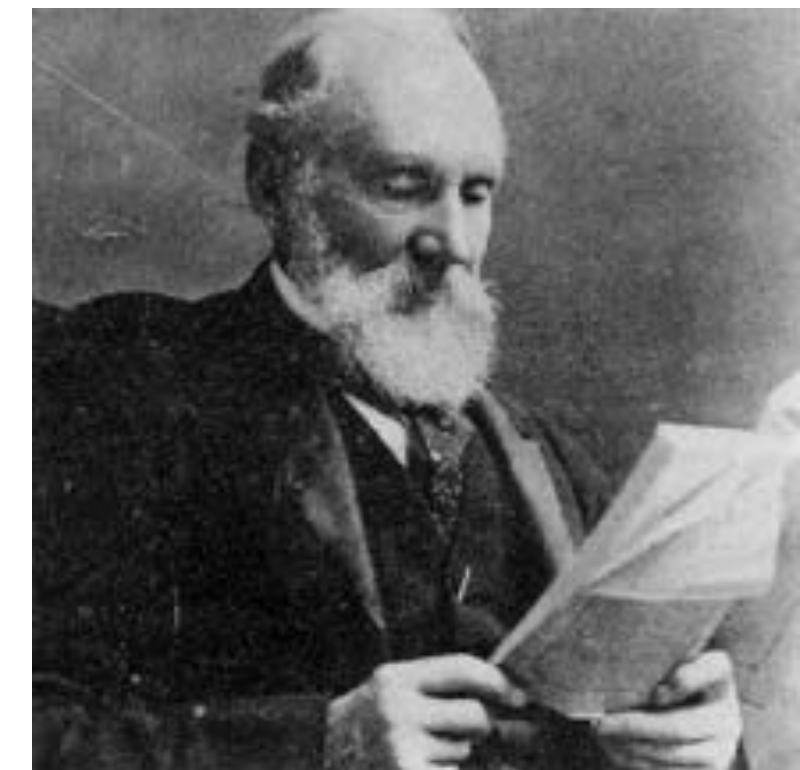
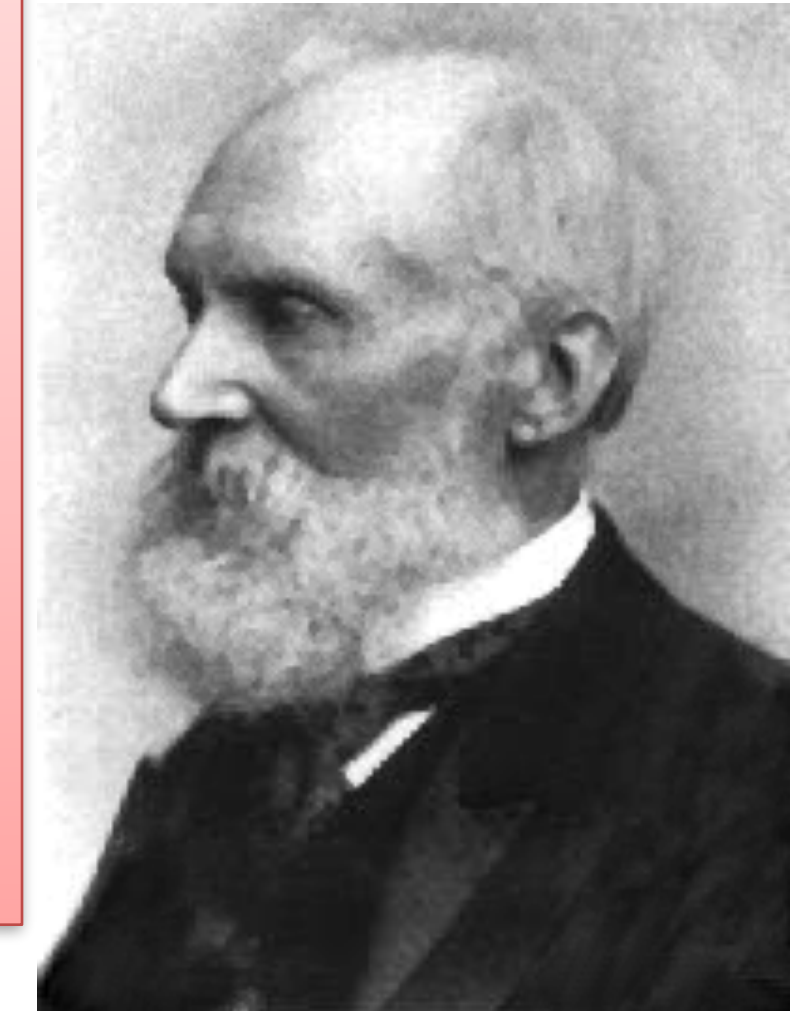




” I often say that when you can measure **what** you are speaking about, and express it in numbers, you know something about it;”

***Lord Kelvin, 1893***

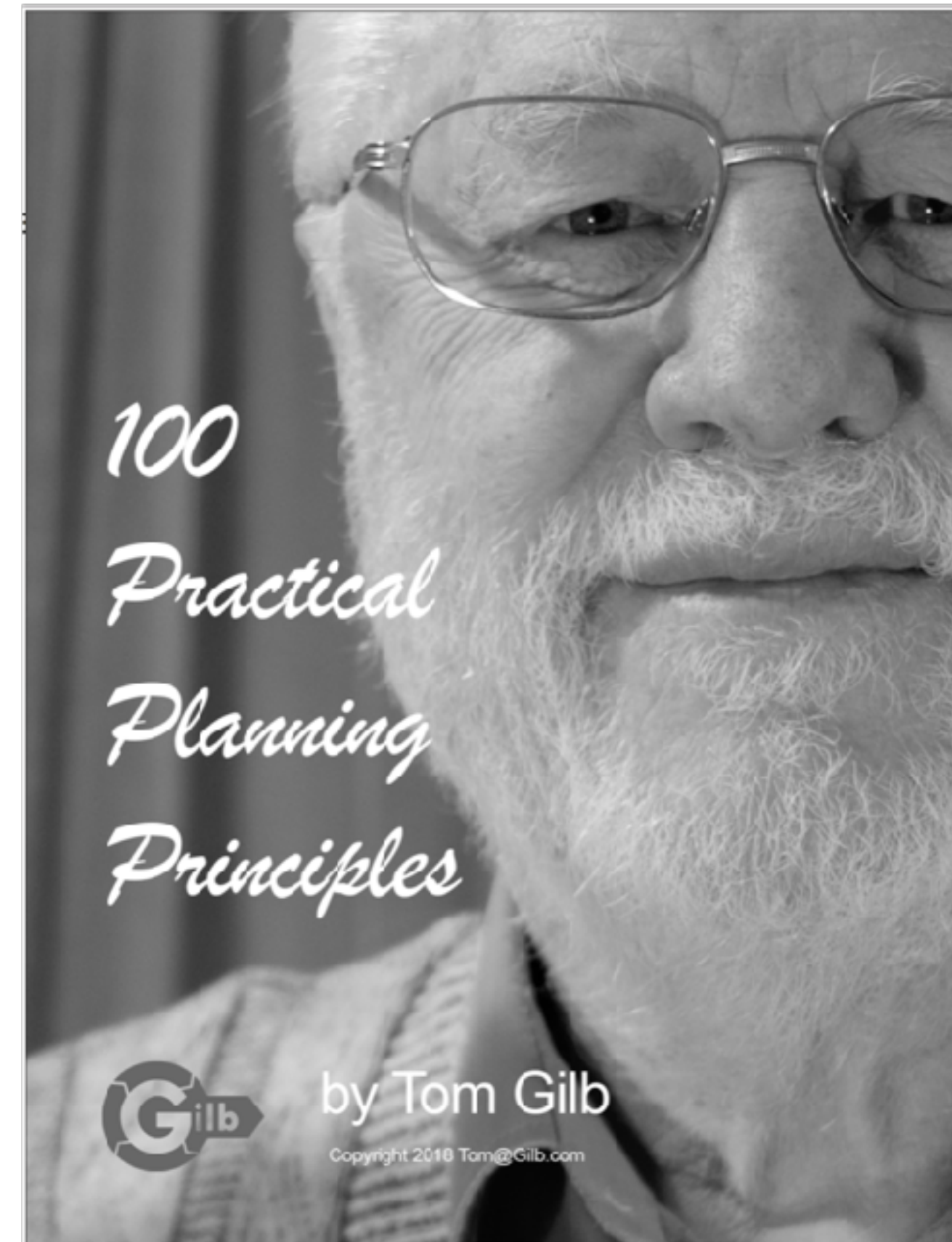
From <http://zapatopi.net/kelvin/quotes.html>





# We have written down the details for our ‘Value Agile’

- **100 Practical Planning Principles.**
- <https://www.gilb.com/offers/Shju4Zqn/checkout>
- FREE GIFT REVIEW COPY FOR YOU ALONE. NO COUPON CODE REQUIRED.
- Be my guest
  - But it demands **hard work** of **smart** people
  - But ‘**This Stuff Works!**’ (Erik Simmons, Intel. CE book foreword.





# 'Value Agile' Course, end slide

BCS SPA 2 Hours Digital Course  
Co-sponsor Specialist Group Quality

For initial presentation  
Wednesday 20th May 2020, 18:00 to 20:00 UK +

Video URL= [https://www.youtube.com/playlist?list=PLK8hokJ0qd3\\_wlvv0j85YhmNfNj8ZJ8M-](https://www.youtube.com/playlist?list=PLK8hokJ0qd3_wlvv0j85YhmNfNj8ZJ8M-)  
(General site of videos, SPA and my courses and talks)

Slide Location Pdf : = <http://concepts.gilb.com/dl974>

Slide Folder (PPTX or Keynote slide copy)  
<https://www.dropbox.com/sh/qfkgv4slajv3s0m/AAAHAS-w7AV5lxTzDbwa7k6na?dl=0>

The 'Value Agile' Book, Free: [tinyurl.com/ValueAgile](http://tinyurl.com/ValueAgile)

By Tom Gilb, in Norway (Kolbotn, near Oslo)

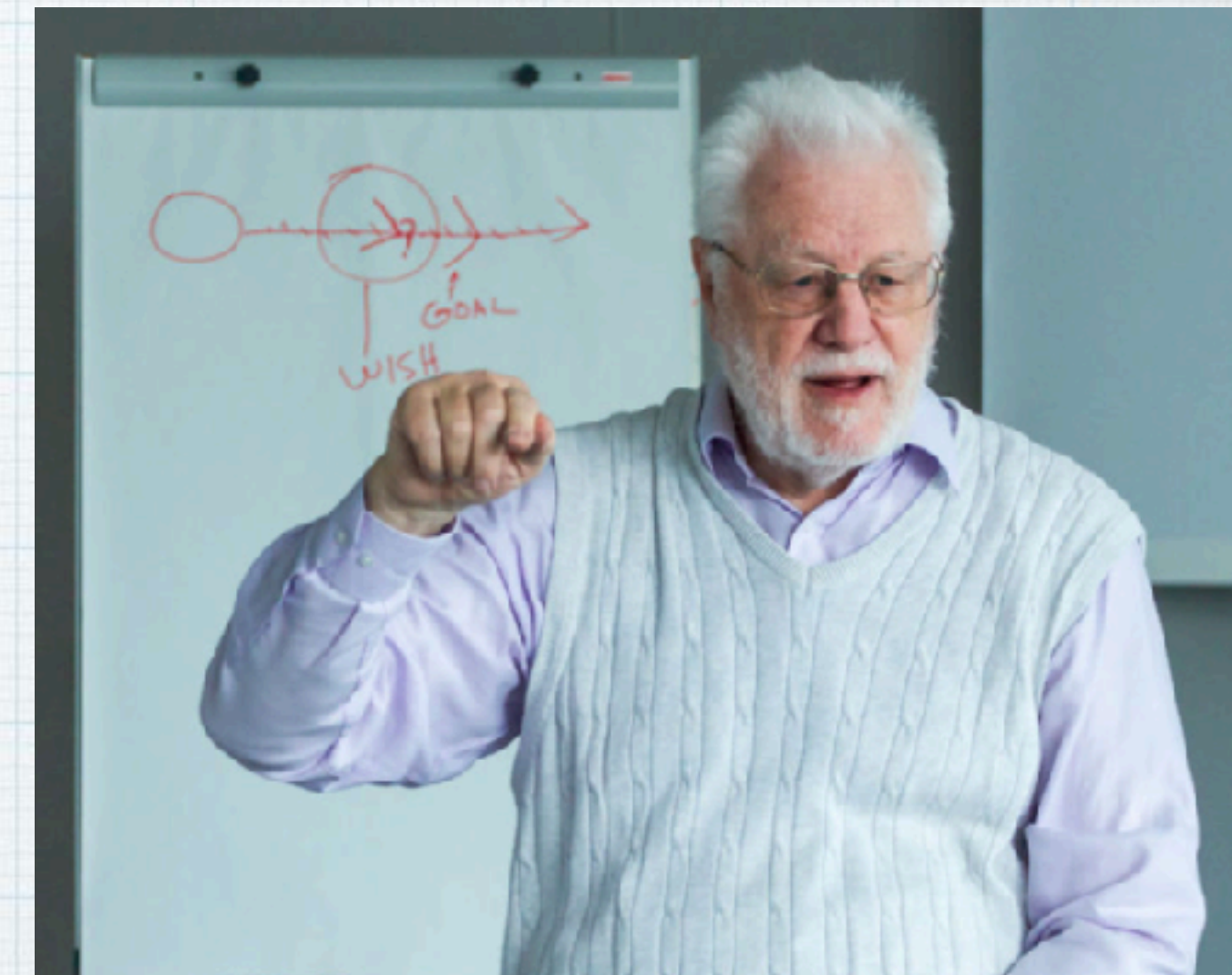
[tom@Gilb.com](mailto:tom@Gilb.com)

[www.Gilb.com](http://www.Gilb.com)

@ImTomGilb (Twitter)

[www.linkedin.com/in/tomgilb](http://www.linkedin.com/in/tomgilb)

Co-sponsored by BCS Specialist Group on Quality



In Poland, 5 Day Masterclass,  
<http://nowy.me/gilb/>



**Last slide**  
**Backups behind this**



# Anti-Financial Crime (AFC) Department Mission Statement:

Be a trusted and respected independent control function that aims to protect the bank from financial crime risk.  
Establish a proactive framework to prevent, detect, and report financial crime risk events.

<- PV, Head of Anti-Financial Crime

Our Vision:

## **Anti-Financial Crime Technology:**

“To provide XXXX Bank the best possible capability to prevent, detect, and report possible financial crime, in-line with the expectations of our global regulators.” <- SC



# Report of AFC Project Results Jan 2019, 3 Sub-projects Using Gilb's Value Driven Methods

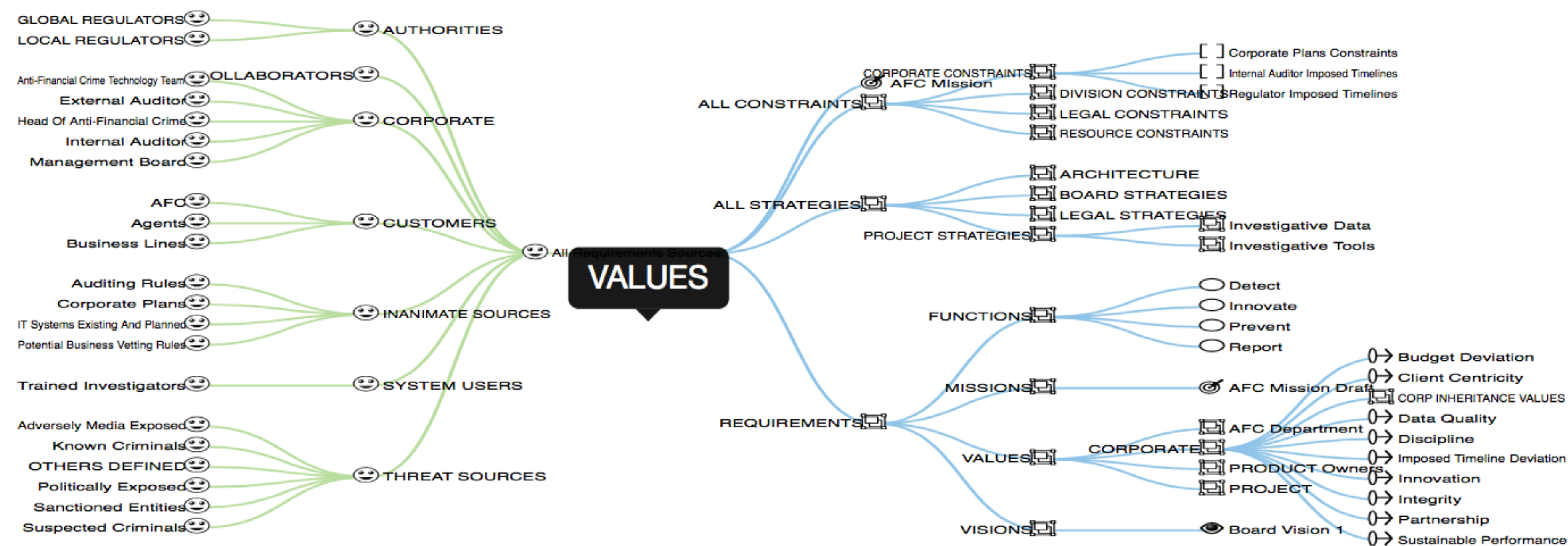
Within a few weeks it enabled us to agree with business stakeholders to **quickly eliminate non-viable costly solution design options**, and justify deferral of lower priority functionalities. This was without jeopardising commitments made to the regulator regarding AFC Transaction Monitoring, Preventative Client Screening and Global Risk Analysis.

Within a few weeks it enabled us to justify the spend on essential business capabilities and align to commitments made to the regulator, counter-act claims against the non-viability of the programme from disgruntled stakeholders, **and saved the programme from being incorrectly cancelled** at the planning stage.

The result has enabled to exit the ideation process with a strategy approach that will enable us to be more effective in adopting the Agile Initiative, consisting of proposed quantifiable measurement definitions. That can now be applied during the subsequent rollout across the department. It has **equipped us with a strategy for delivering organisational change through the initiative on programmes that are already under significant cost pressures.**

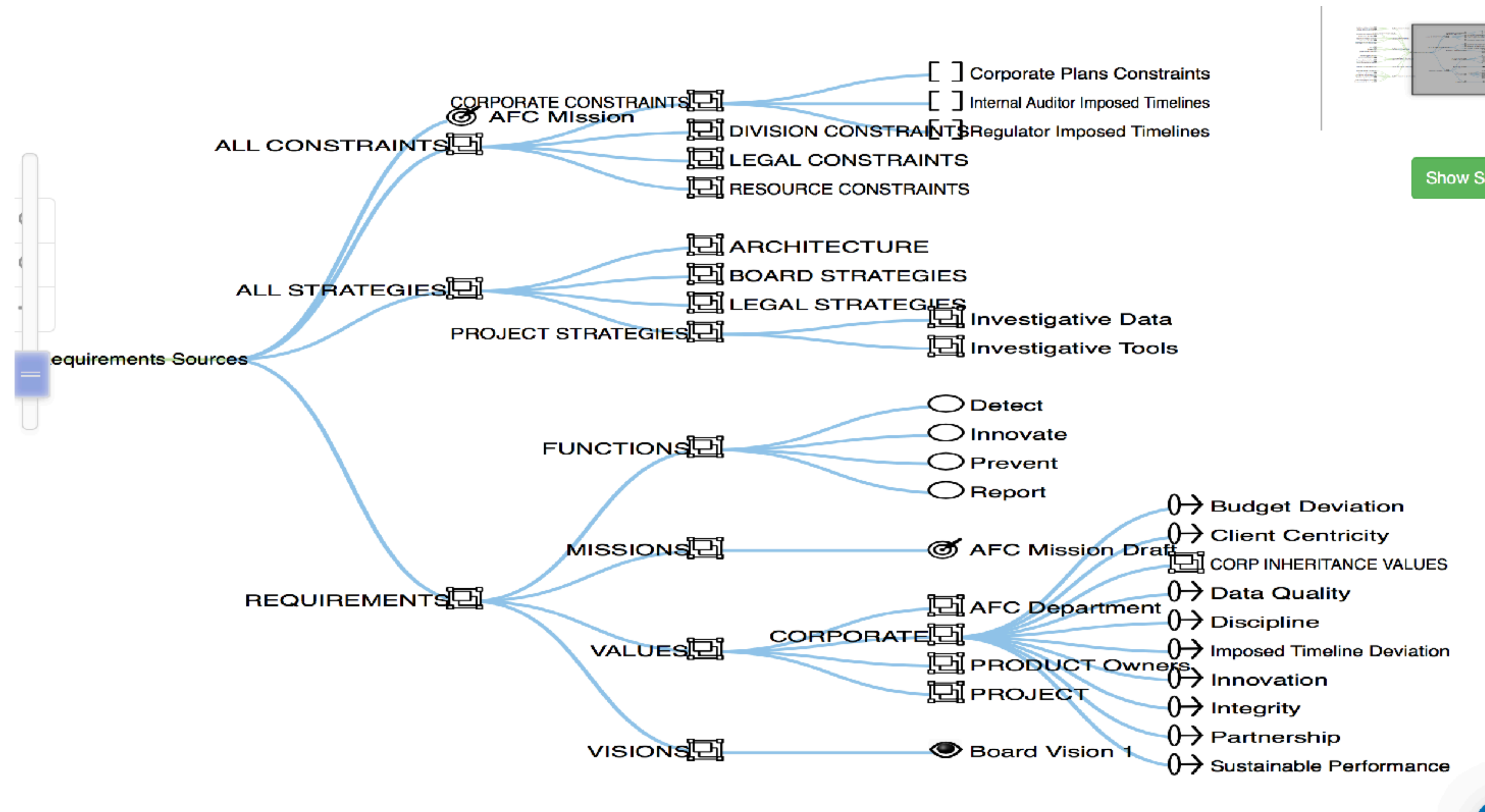


# Diagram over AFC Planning





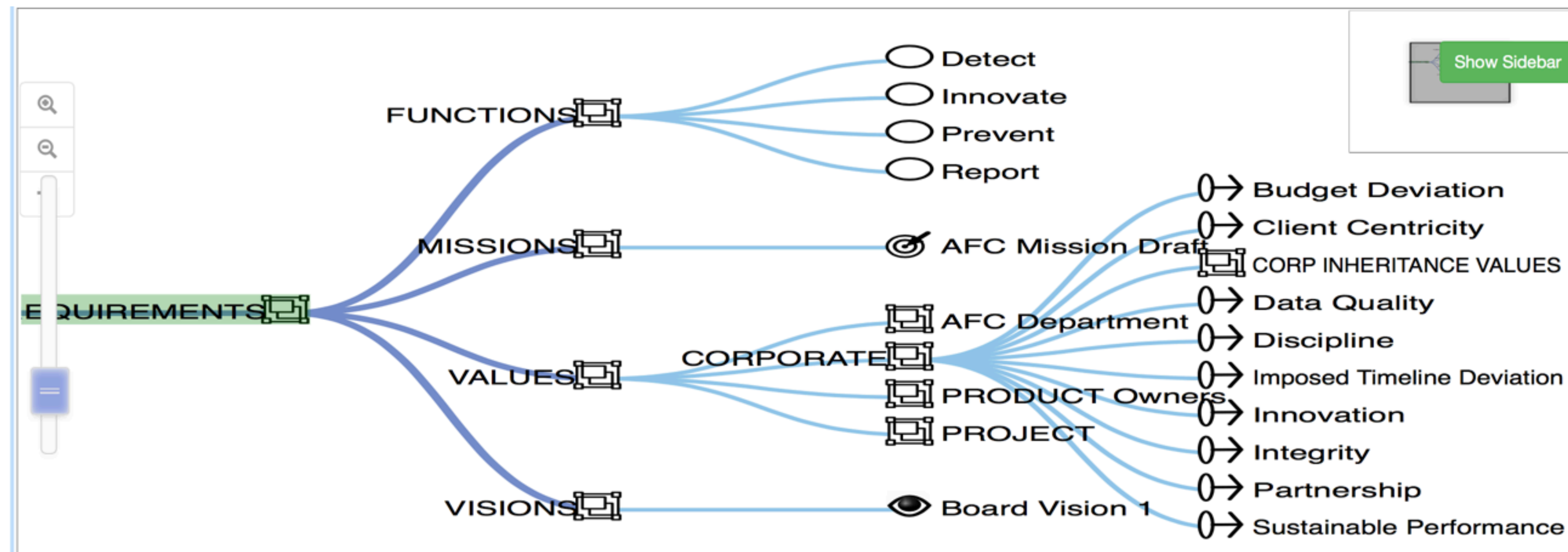
# AFC Requirements Constraints and 'Architecture' Overview





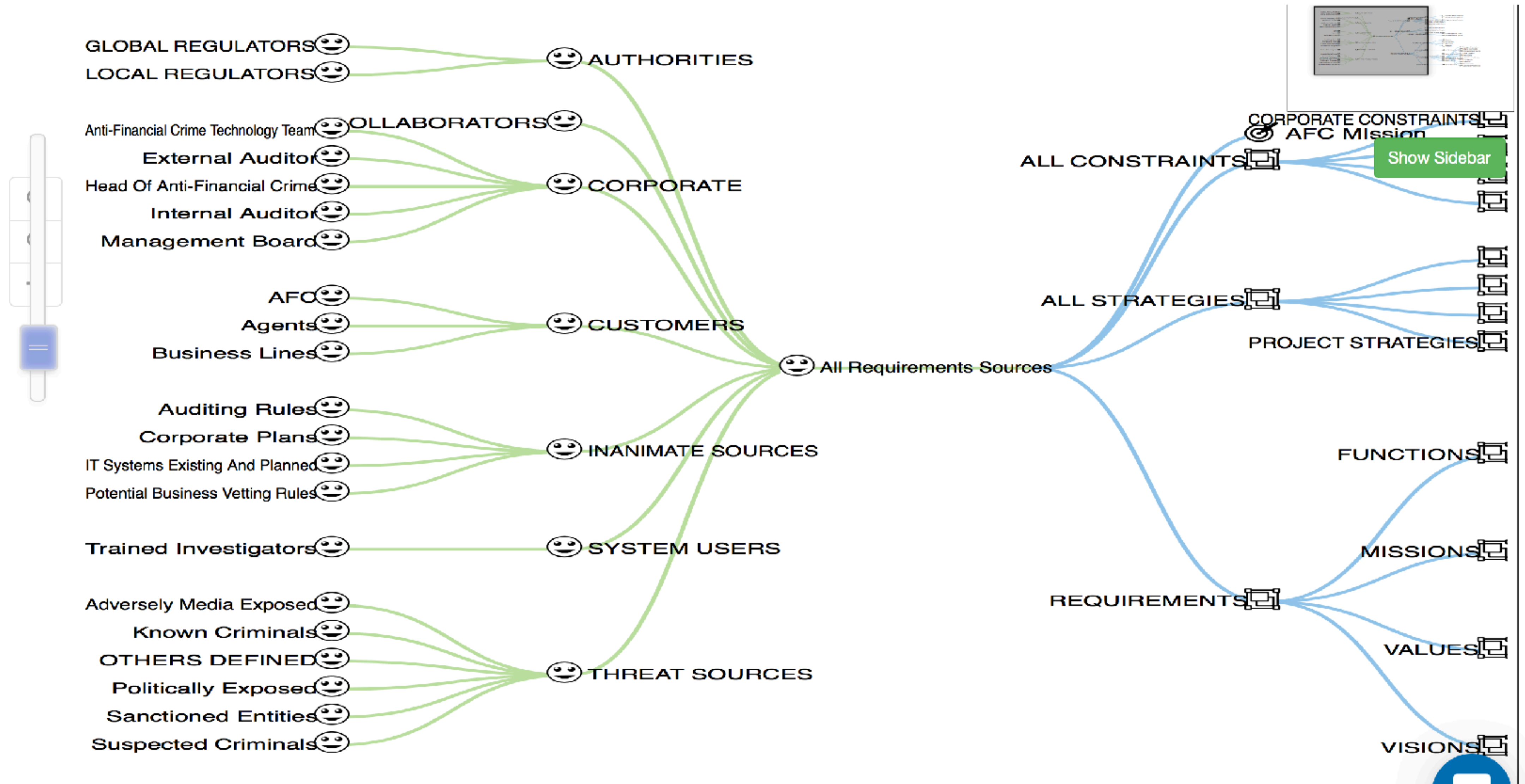
# AFC Requirements

(focus on 4 types, detail for Values))





# Stakeholders AFC





# Critical set of AFC Objectives

## Critical Project Objectives FINANCE IT PROCESS

Tag: (→) [Budget Deviation](#)  
Scale:  
Status : 0  
Wish : 0

Tag: (→) [Client Centricity](#)  
Scale:  
Status : 0  
Wish : 0

Tag: (→) [Data Quality](#)  
**Initial Draft Example.**Scale: % of #Perfect Data# for defined #AFC# [Purposes] [Data] [Qualities] in [Business Areas] as defined by [Quality Instances] for given [Sources]  
**Status** [Purposes = **Detect**, Data = **All**, Qualities = **All**, Business Areas = **All**, Quality Instances = **All**, ]: **90** % Data Level Required  
**Short Term.Wish** [Purposes = **Detect**, Data = **Financial Amounts,Dates**, Qualities = **Complete**, Business Areas = **Corporate banking**, Quality Instances = **Regulations**, ]: **95** % Data Level Required  
**Medium Term.Wish** [Purposes = **Detect**, Data = **All**, Qualities = **All**, Business Areas = **All**, Quality Instances = **All**, ]: **97** % Data Level Required  
**Pushing Envelope.Stretch** [Purposes = **Detect**, Data = **All**, Qualities = **All**, Business Areas = **All**, Quality Instances = **All**, ]: **99.9** % Data Level Required  
**Stakeholders:** Auditing Rules, External Auditor,

Tag: (→) [Discipline](#)  
Scale:  
Status : 0  
Wish : 0

Tag: (→) [Imposed Timeline Deviation](#)  
Scale:  
Status : 0  
Wish : 0

Tag: (→) [Innovation](#)  
Scale:  
Status : 0  
Wish : 0

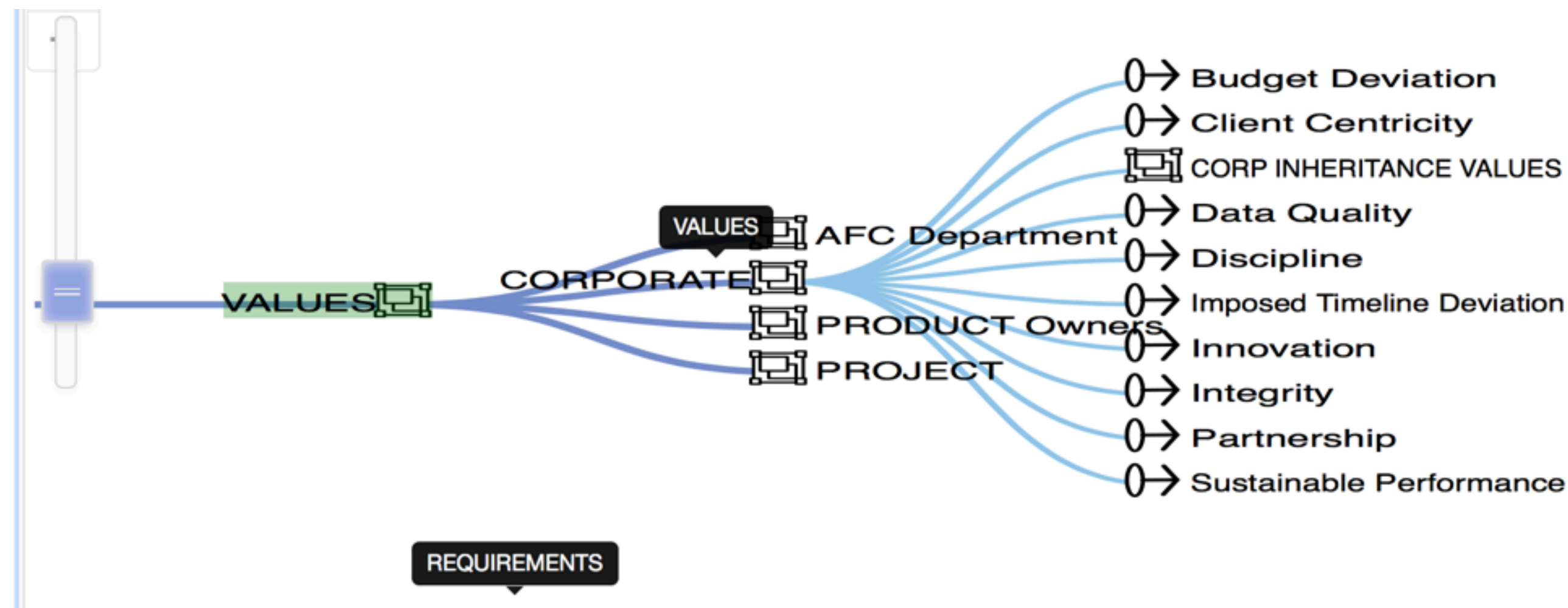
Tag: (→) [Partnership](#)  
Scale:  
Status : 0  
Wish : 0

Tag: (→) [Sustainable Performance](#)  
Scale:  
Status : 0  
Wish : 0





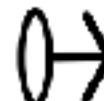




# Corporate Objectives AFC





# Data Quality

## Value Quantified

 <u>Data Quality</u>		0.0.1	
<u>Business Value</u> <span>critical</span>		(✎ by tomgilb - 11 minutes ago)	
Is Part Of:  <u>CORPORATE</u>			
 <b>Ambition Level:</b> more than sufficient data quality to meet present and future needs for [AFC] purposes			
 <b>Initial Draft Example.Scale:</b> % of #Perfect Data# for defined #AFC# [Purposes] [Data] [Qualities] in [Business Areas] as defined by [Quality Instances] for given [Sources]			
 <b>Stakeholders:</b> Auditing Rules, External Auditor.			
© <b>Status:</b> Level: <b>90</b> % Data Level Required [Purposes = <b>Detect</b> , Data = <b>All</b> , Qualities = <b>All</b> , Business Areas = <b>All</b> , Quality Instances = <b>All</b> , ] When January 2018			
© <b>Short Term.Wish:</b> Level: <b>95</b> % Data Level Required [Purposes = <b>Detect</b> , Data = <b>Financial Amounts,Dates</b> , Qualities = <b>Complete</b> , Business Areas = <b>Corporate banking</b> , Quality Instances			
© <b>Medium Term.Wish:</b> Level: <b>97</b> % Data Level Required [Purposes = <b>Detect</b> , Data = <b>All</b> , Qualities = <b>All</b> , Business Areas = <b>All</b> , Quality Instances = <b>All</b> , ] When January 2020			
© <b>Pushing Envelope.Stretch:</b> Level: <b>99.9</b> % Data Level Required [Purposes = <b>Detect</b> , Data = <b>All</b> , Qualities = <b>All</b> , Business Areas = <b>All</b> , Quality Instances = <b>All</b> , ] When January 2023			



# Data Quality (?)

## A 'Wish' requirement detail

© **Status:**

(✎ by tomgilb - 4 days ago) 0

90 % Data Level Required [Purposes = **Detect**, Data = **All**, Qualities = **All**, Business Areas = **All**, Quality Instances = **All**, ] (to be delivered by end of: January 2018 )

**Source:**  
tsg wild guess for illustrative purposes

Show Sidebar

© **Short Term.Wish:** Change...

(✎ by tomgilb - 4 days ago) 0

**Scale Level:** % Data Level Required

95

**By When:**

02/2019

Date format: mm/yyyy (e.g. "02/2017" for February 2017)

**Qualifiers:** Copy from...

[Purposes] =  
\* Detect

[Qualities] =  
\* Complete

[Quality Instances] =  
\* Regulations

+Add additional qualifier

[Data] =  
\* Financial Amounts \* Dates

[Business Areas] =  
\* Corporate banking

[Sources] =  
Click to select the value(s) for this qualifier

**Source:**  
Type something

Add Comment...

© **Medium Term.Wish:** Level: 97 % Data Level Required [Purposes = **Detect**, Data = **All**, Qualities = **All**, Business Areas = **All**, Quality Instances = **All**, ] When January 2020



# A Stretch level requirement detail

© **Short Term.Wish:**

(✎ by tomg)

95 % Data Level Required [Purposes = **Detect**, Data = **Financial Amounts,Dates**, Qualities = **Complete**, Business Areas = **Corporate banking**, Quality Instances = **Regu**  
by end of: February 2019 )

© **Medium Term.Wish:** Level: 97 % Data Level Required [Purposes = **Detect**, Data = **All**, Qualities = **All**, Business Areas = **All**, Quality Instances = **All**, ] When January

© **Pushing Envelope.Stretch:** Change...

(✎ by tomgilb - 4 days a

Scale Level: % Data Level Required

99,9

Qualifiers: 📋 Copy from...

[Purposes] =  
✕ Detect

[Qualities] =  
✕ All

[Quality Instances] =  
✕ All

By When:

01/2023

Date format: mm/yyyy (e.g. "02/2017" for February 2017)

[Data] =  
✕ All


[Business Areas] =  
✕ All

[Sources] =  
Click to select the value(s) for this qualifier





# ‘Data Quality’


## the ‘Scale of Measure’ definition detail


 **Data Quality**


**Business Value** critical (by tomgilb - 11 minutes ago)


Is Part Of:  **CORPORATE**

 **Ambition Level:** more than sufficient data quality to meet present and future needs for [AFC] purposes Show Sidebar

 **Initial Draft Example.** **Scale:**  (by tomgilb - 4 days ago)

**Scale Description:** 

% of #Perfect Data# for defined #AFC# [Purposes] [Data] [Qualities] in [Business Areas] as defined by [Quality Instances] for given [Sources] 

 Press **⌘+e** to show formatting toolbar.

**AFC:** defined as:  
Anti-Financial Crime


**Business Areas:** defined as:  
 ↑ Create in Project...

**Data:** defined as:  
 ↑ Create in Project...

**Perfect Data:** defined as:  
 ↑ Create in Project...

**Purposes:** defined as:  
 ↑ Create in Project...

**Detect:** defined as:  
Provide trained investigators within AFC teams with an alert that a Financial Crime may have occurred.  
- Provide the investigative tools and data to allow the determination of the level of suspicion attached to an alert.





# The 'Status Level' of Data Quality (detail)

© **Tag.Status:** Change...(✎ by tomgilb - 4 days)

**Scale Level:** % Data Level Required

90

**When:**

01/2018

Date format: mm/yyyy (e.g. "02/2017" for February 2017)

**Qualifiers:** ✎ Copy from...

[Purposes] =

✖ Detect

[Qualities] =

✖ All

[Quality Instances] =

✖ All

+Add additional qualifier

**[Data] =**

✖ All

**[Business Areas] =**

✖ All

**[Sources] =**

Click to select the value(s) for this qualifier

**Source:**

tsg wild guess for illustrative purposes

💬 Add Comment...

© **Short Term.Wish:** Level: **95** % Data Level Required [Purposes = **Detect**, Data = **Financial Amounts,Dates**, Qualities = **Complete**, Business Areas = **Corporate**

© **Medium Term.Wish:** Level: **97** % Data Level Required [Purposes = **Detect**, Data = **All**, Qualities = **All**, Business Areas = **All**, Quality Instances = **All**, ] When Janu

© **Pushing Envelope.Stretch:** Level: **99.9** % Data Level Required [Purposes = **Detect**, Data = **All**, Qualities = **All**, Business Areas = **All**, Quality Instances = **All**, ] V



# Stakeholders

## (direct association with Values)

Tag.Stakeholders: Change...

by tomgilb - 4 days ago 0

Link to existing...

Link to new...

Stakeholder

Roles

Notes

Actions

From: Auditing Rules

\* Authority

\* Expert

Enter link description

Show Sidebar

From: External Auditor

\* Authority

Enter link description

Enter additional information

Source:

tsg draft example

Add Comment...

© Status: Level: 90 % Data Level Required [Purposes = Detect, Data = All, Qualities = All, Business Areas = All, Quality Instances = All, ] When January 2018

© Short Term.Wish: Level: 95 % Data Level Required [Purposes = Detect, Data = Financial Amounts,Dates, Qualities = Complete, Business Areas = Corporate banking, Quality Instances ...

© Medium Term.Wish: Level: 97 % Data Level Required [Purposes = Detect, Data = All, Qualities = All, Business Areas = All, Quality Instances = All, ] When January 2020

© Pushing Envelope.Stretch: Level: 99.9 % Data Level Required [Purposes = Detect, Data = All, Qualities = All, Business Areas = All, Quality Instances = All, ] When January 2023



# TWELVE TOUGH QUESTIONS

- 1. Why isn't the improvement quantified?
- 2. What is degree of the risk or uncertainty and why?
- 3. Are you sure? If not, why not?
- 4. Where did you get that from? How can I check it out?
- 5. How does your idea affect my goals, measurably?
- 6. Did we forget anything critical to survival?
- 7. How do you know it works that way? Did it before?
- 8. Have we got a complete solution? Are all objectives satisfied?
- 9. Are we planning to do the 'profitable things' first?
- 10. Who is responsible for failure or success?
- 11. How can we be sure the plan is working, during the project, early?
- 12. Is it 'no cure, no pay' in a contract? Why not?

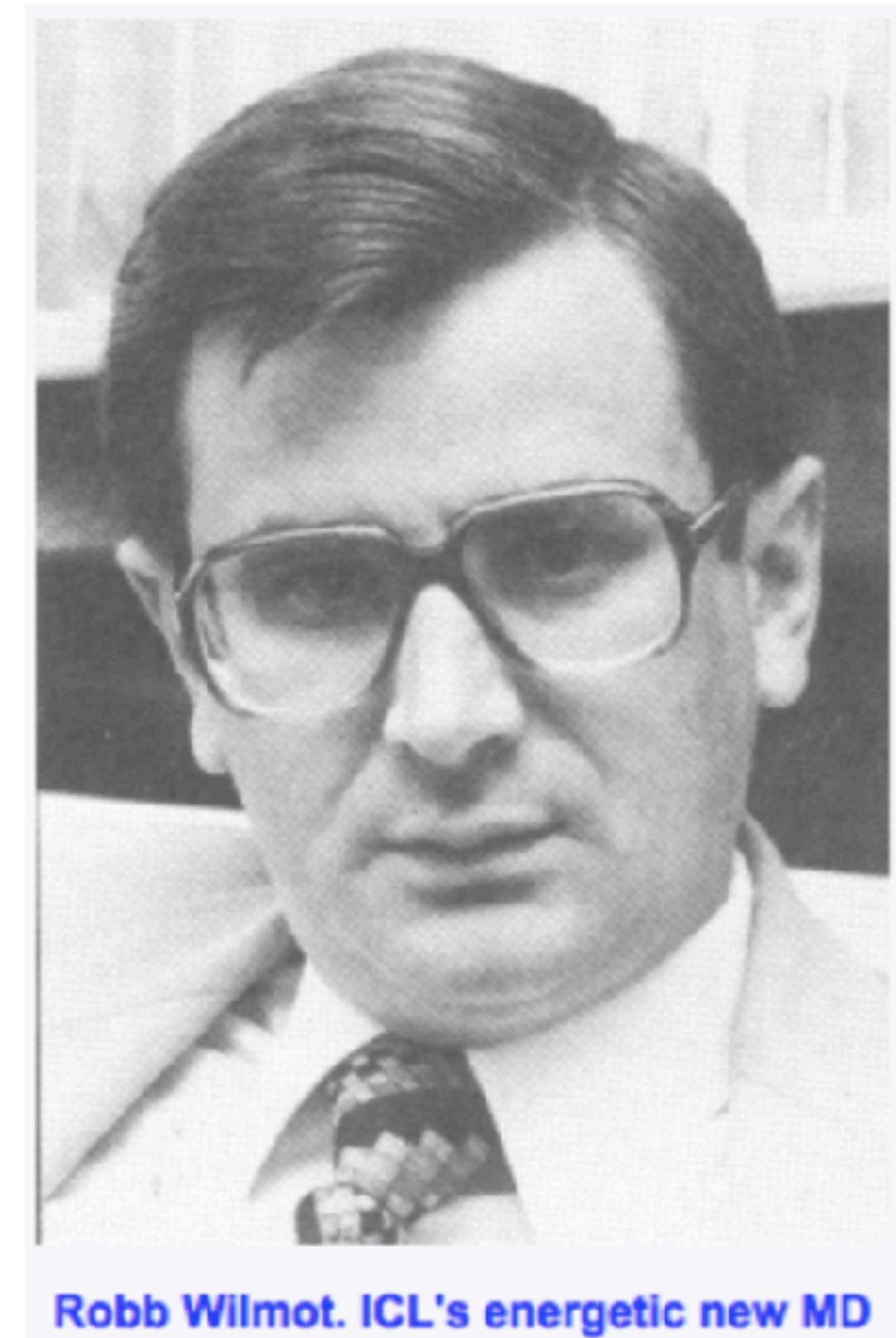
<http://www.gilb.com/dl24> is a paper on 12 tough questions



# **The Bill of Rights for Company Communication**

(written by Tom)

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



**PS ICL went into profit for next 15 years, after 7 years in red**





## 20 Sept, 2011 Report on Gilb Evo method (Richard Smith, Citigroup)



- <http://rsbtechnology.co.uk/blog:8>
- Back in 2004, I was employed by a large investment bank in their FX e-commerce IT department as a business analyst.
- The wider IT organisation used a complex waterfall-based project methodology that required use of an intranet application to manage and report progress.
- However, it's main failings were that it almost totally missed the ability to track delivery of actual value improvements to a project's stakeholders, and the ability to react to changes in requirements and priority for the project's duration.
- The toolset generated lots of charts and stats that provided the illusion of risk control, but actually provided very little help to the analysts, developers and testers actually doing the work at the coal face.
- The proof is in the pudding;
  - I have **used Evo** (albeit in disguise sometimes) on two large, high-risk projects in front-office investment banking businesses, and several smaller tasks.
  - On the largest critical project, the original business functions & performance objective **requirements document, which included no design, essentially remained unchanged** over the 14 months the project took to deliver,
  - but **the detailed designs** (of the GUI, business logic, performance characteristics) **changed many many times**, guided by lessons learnt and **feedback** gained by delivering a succession of early deliveries to real users.
  - In the end, the new system responsible for 10s of USD billions of notional risk, **successfully went live over one weekend for 800 users worldwide**, and **was seen as a big success by the sponsoring stakeholders**.





**Richard Smith**



28 March 2015

“ I attended a 3-day course with you and Kai whilst at Citigroup in 2006”

© Gilb.com





Previous PM Methods:  
No 'Value delivery tracking'.  
No change reaction ability



Richard Smith

- “However, (our old project management methodology) main failings were that
- it almost totally missed the ability to track delivery of actual *value* improvements to a project's stakeholders,
- and the ability to react to changes
  - in requirements and
  - priority
  - for the project's duration”





We only had the illusion of control.  
But little help to testers and analysts



Richard Smith

- “The (old) toolset generated lots of charts and stats
- that provided the illusion of risk control.
- But actually provided very little help to the analysts, developers and testers actually doing the work at the coal face.”





The proof is in the pudding;



Richard Smith

- “The proof is in the pudding;
- I have **used Evo**
  - *(albeit in disguise sometimes)*
  - on two large, high-risk projects in front-office investment banking businesses,
  - and several smaller tasks. “





*Experience: if top level requirements are separated from design, the 'requirements' are **stable!***



Richard Smith

- “On the largest critical project,
- the original ***business functions & performance objective requirements document,***
- ***which included no design,***
- essentially remained ***unchanged***
- over the **14 months** the project took to deliver,....”



# Dynamic (Agile, Evo) design testing: not unlike ‘Lean Startup’



Richard Smith

- “... but **the detailed designs**
  - (of the GUI, business logic, performance characteristics)
- **changed many many times,**
  - guided by lessons learnt
  - and **feedback** gained by
  - delivering a succession of early deliveries
  - to real users”





It looks like the stakeholders liked the top level system qualities, on first try



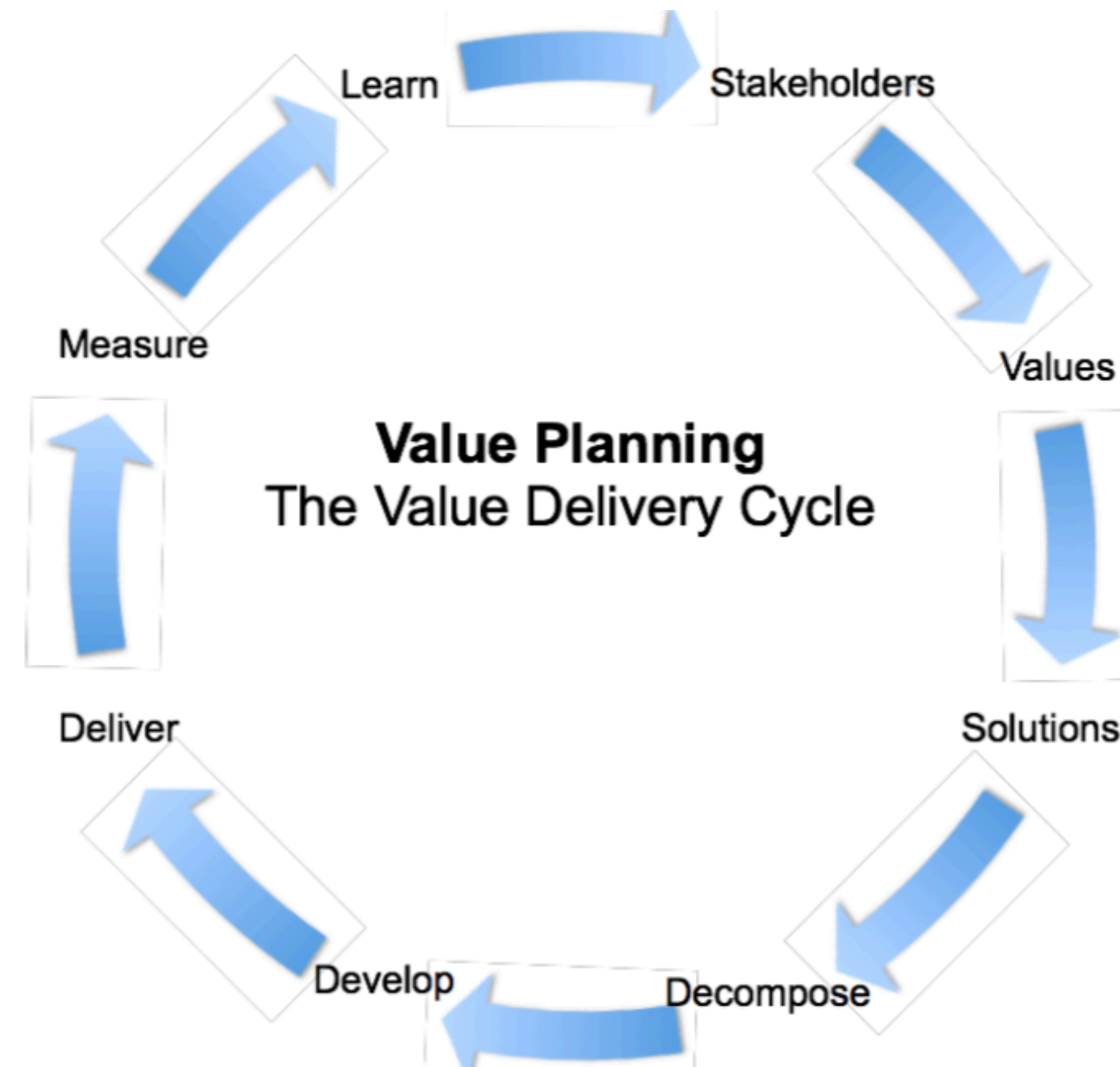
Richard Smith

- “ In the end, the new system responsible for 10s of USD billions of notional risk,
- **successfully went live**
- **over one weekend**
- **for 800 users worldwide,**
- and **was seen as a big success**
- **by the sponsoring stakeholders.”**



$\Sigma$

1. Focus on delivering BANK values, quantified.
2. Plan a week, then start the value delivery stream
3. Resources are given for quantified bank Value improvements
4. Continued resources are dependent on actual measurable delivery levels
5. Shift from 'IT' focus to Bank Systems Focus (IT is a tool, Agile is a tool)
6. Do this at all levels of management, starting starting with this Change project
7. Pilot some 'IT' projects with Value Planning
  - A. SOME OLD PROJECTS. WHICH ARE STUCK
  - B. SOME NEW PROJECTS (like AFC)





Surely we have used our 2 hours  
by here?





My “value Agile” book References for the ‘agile manifesto’ chapter  
Are in the presenter notes of this slide  
You will also find them in the ‘Value Agile’ Book  
Link earlier



# 'Value Agile'

## Book References from it.

- \* In presenter notes this slide and in the book too.

