Tom Gilb
gilb.com
tom@Gilb.com
@ImTomGilb

Enterprise Architecture,
Conference, **#BCSEASG**
Specialist Interest Group, BCS

ARCHITECTURE ENGINEERING:

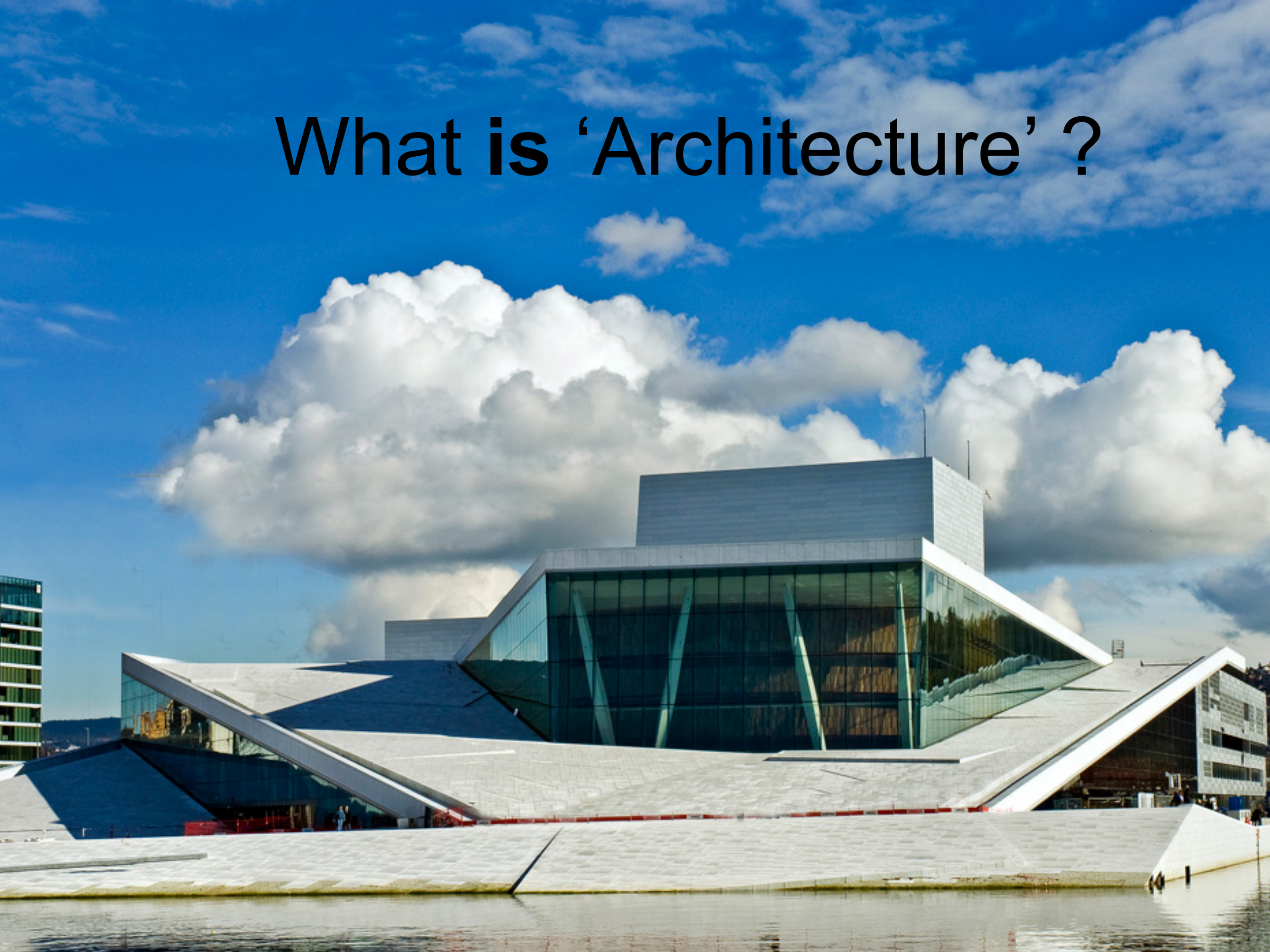
WHY AND HOW, WE HAVE TO 'UP OUR GAME'

AND QUANTIFY DESIGN OBJECTIVES AND COSTS.

BCS London, Covent Garden
17 July 2017
Keynote 10 to 10:30 (30 minutes)

These slides are at
https://tinyurl.com/Gilb17July

What **is** ‘Architecture’ ?



Architecture must try to 'satisfy' Requirements

A —> R

Architecture 'impacts' Requirements

->

http://www.gilb.com/dl857

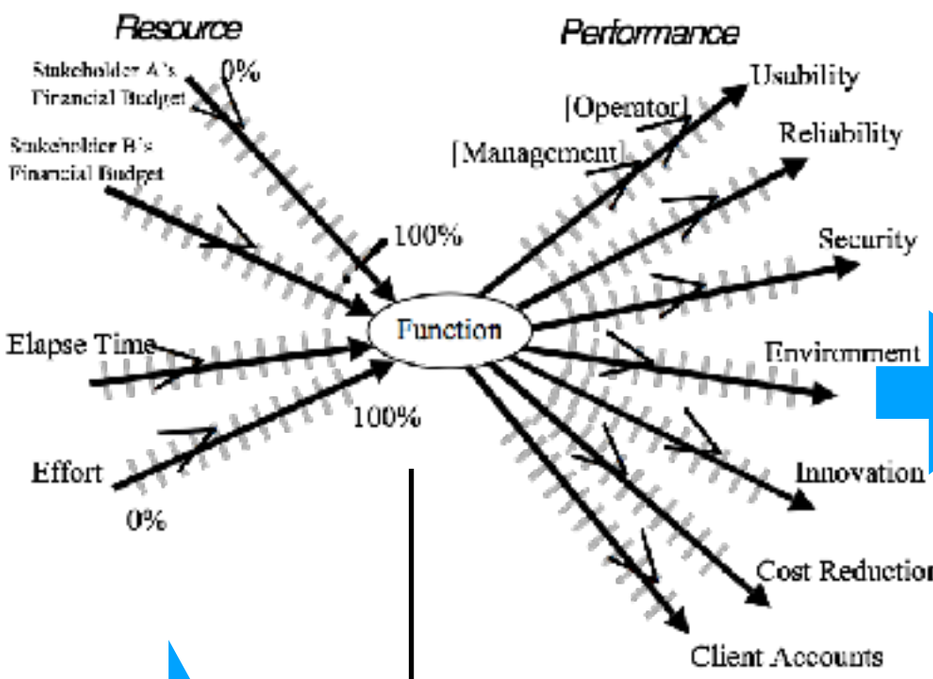
The Basic Design Steps Logic:
a summary

1. **Constraints determine environments.**
2. **Environments determine stakeholders**
3. **Stakeholders have values and priorities**
4. **Values have many dimensions**
5. **Stakeholders determine value levels**
6. **Design hypotheses should be powerful and efficient ideas, for satisfying stakeholder needs**
7. **Design hypotheses can be evaluated quantitatively, with respect to all quantified objectives and resources**
8. **Designs can be decomposed, to find more efficient design subsets, that can be implemented early**
9. **Designs can be implemented sequentially, and their value-delivery, and resource costs, measured**
10. **Designs that unexpectedly threaten achievement of objectives, or excessive use of resources, can be removed or modified.**
11. **Designs that have the best set of effects on objectives, for the least consumption of limited resources, should generally be selected for early implementation.**
12. **A design increment can have unacceptable results, in combination with previous increments, and they, or it, might need removal or modification**
13. **When all objectives are reached, the process of design is complete: except for possible optimization of operational resources, by even-better design.**
14. **When deadlined and budgeted implementation-resources are used up, it might be reasonable to negotiate additional resources; especially if the incremental values are worth the additional resources.**

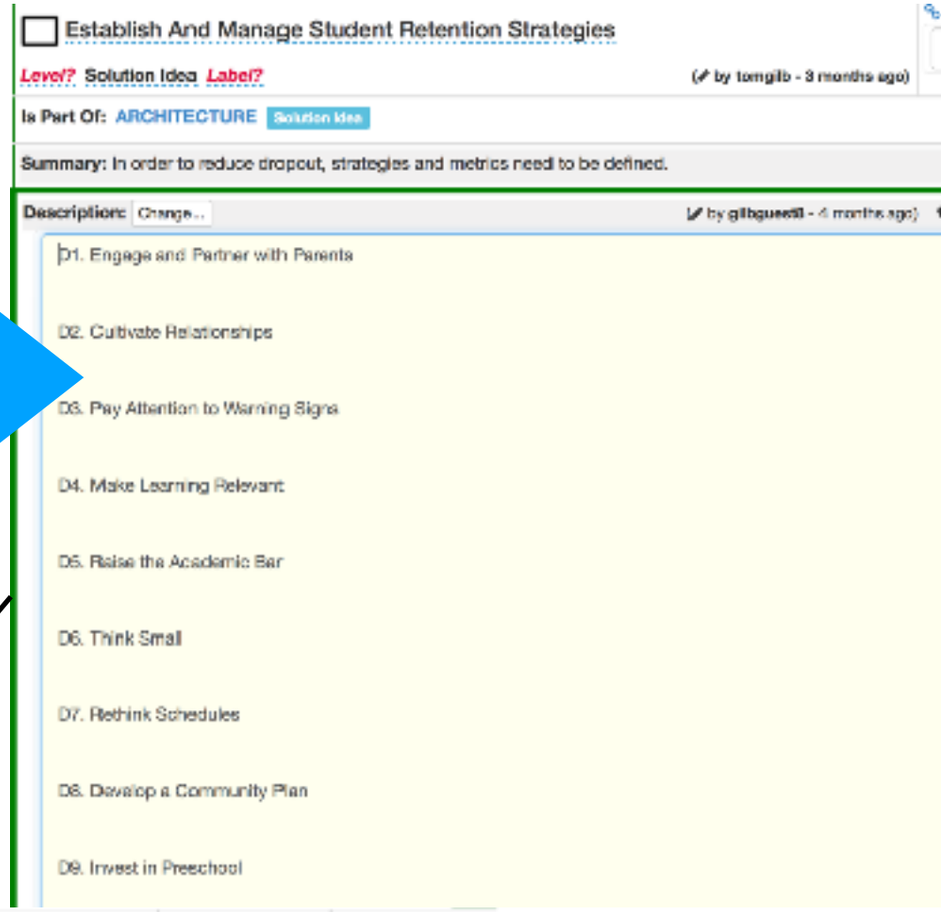
The Logic of Design:
Design Process Principles.
Tom Gilb, 2016, Paper.
http://www.gilb.com/dl857

The Architecture process
is
driven
by
requirements

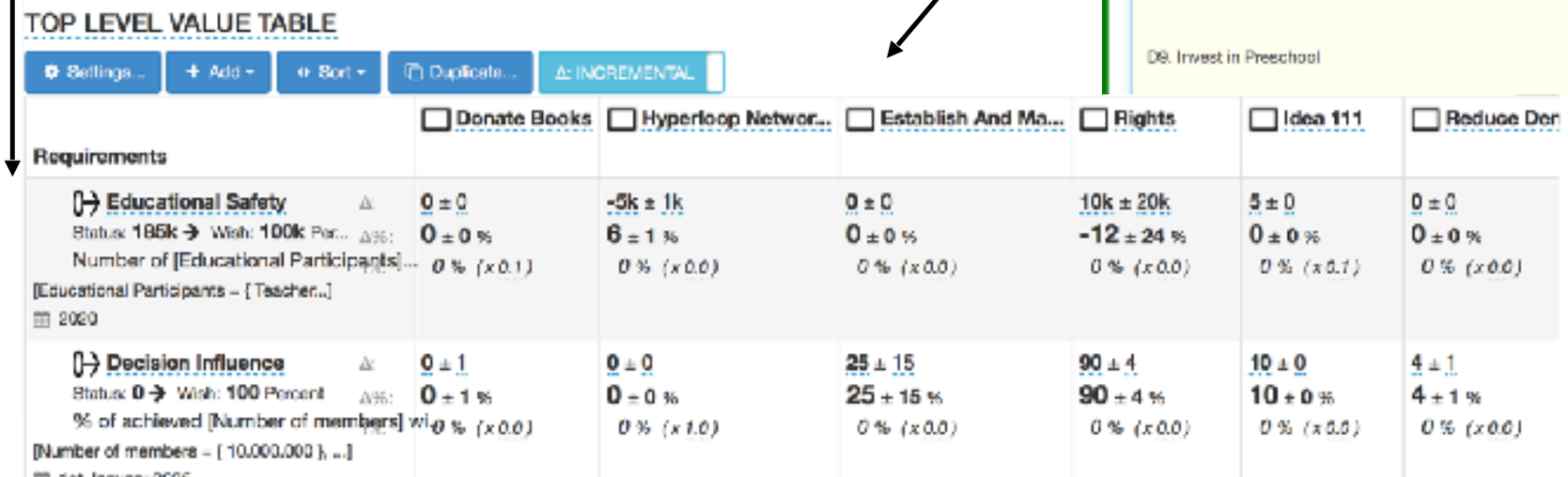
Requirements -> Architecture Process -> Architecture Specs



Architecture Process



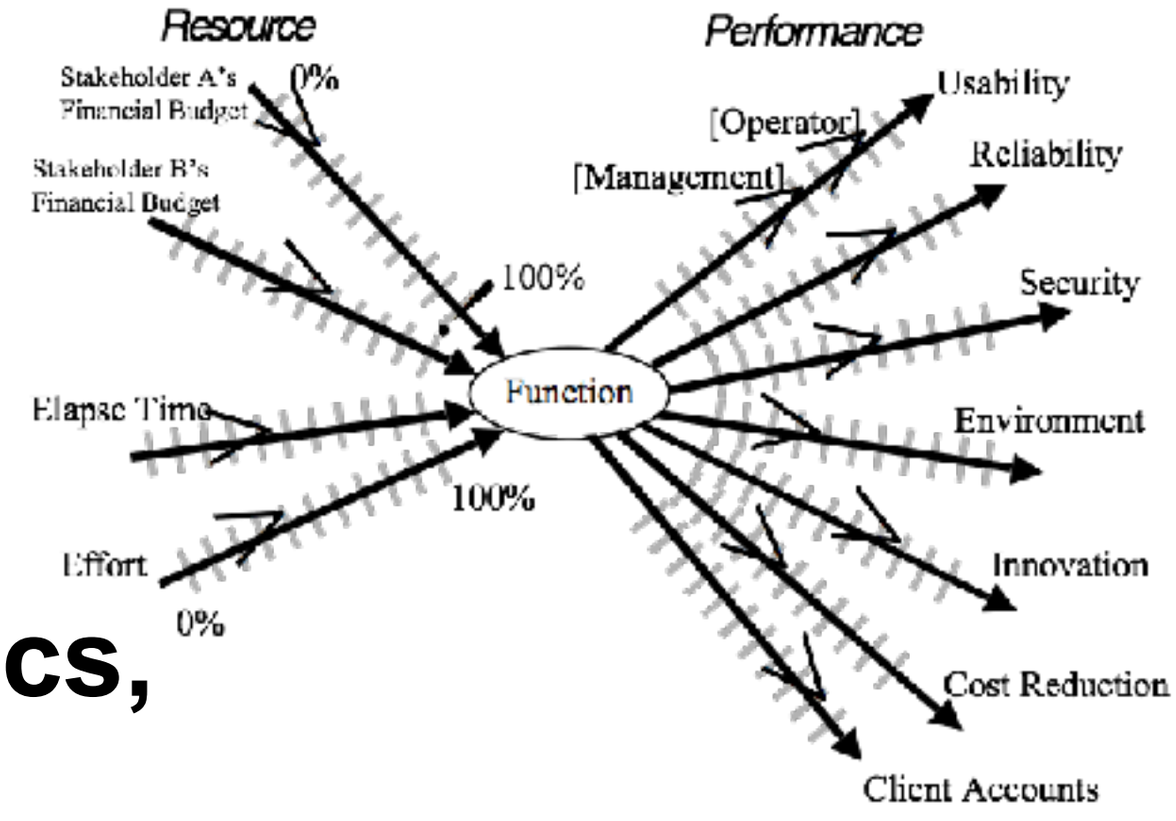
Process Engineering Tool



The architecture is there to satisfy requirements

Architecture
that *never* refers to
critical
qualities,
performance characteristics,
costs,
and **constraints**

Is **not** really architecture
Of any kind



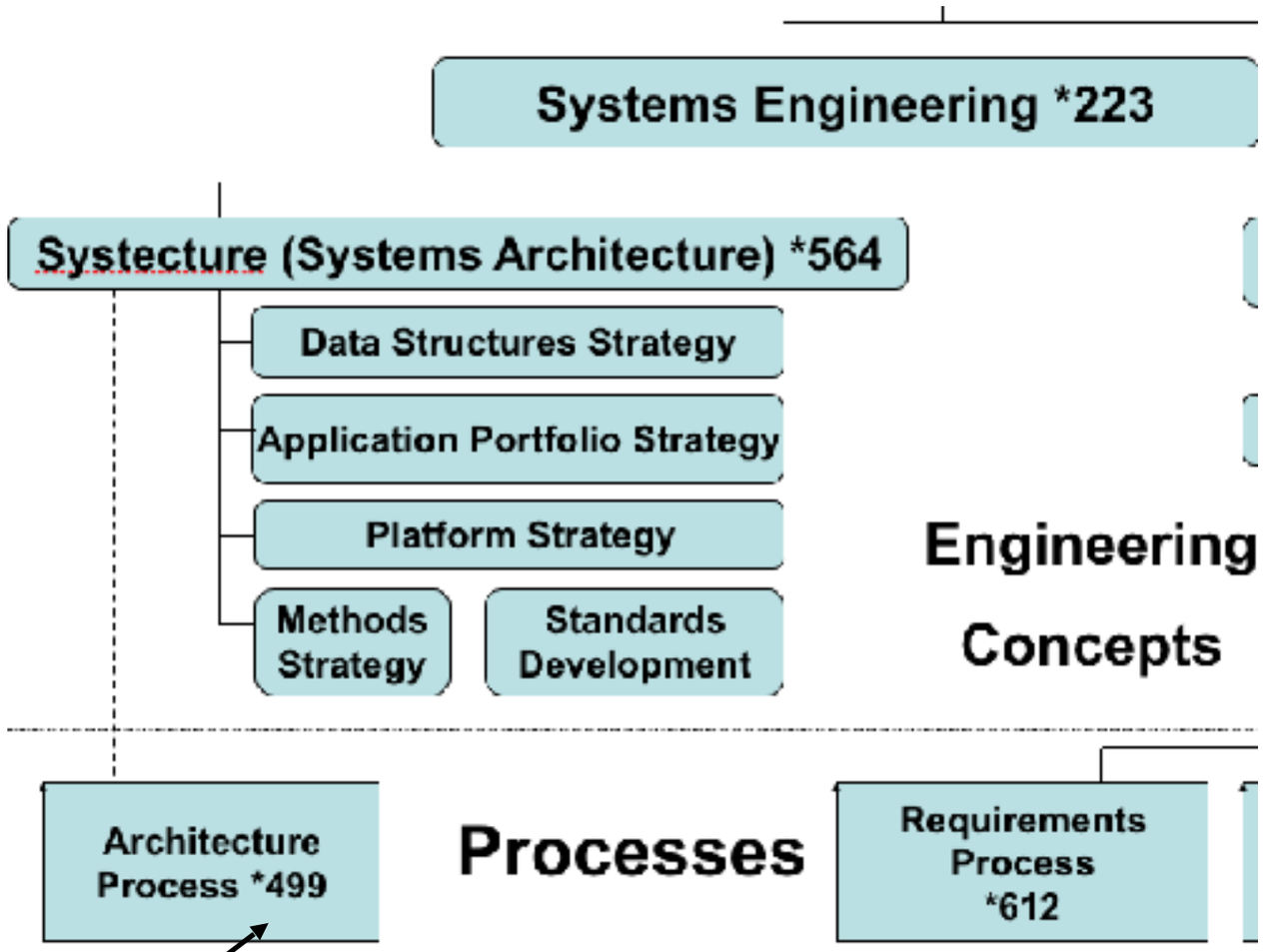
“Architecture Engineering” A

high level design process

- The <u>architecture engineering</u> process
  - puts in place the systems architecture,
  - which is a controlling mechanism for the **design engineering** of any project.

- Architecture engineering
  - defines the strategic framework (the systems architecture),
    - which *specialist* design engineering has to work within. (like Security, UI, Perf.)
  - It lays down the standards, which control such matters as the tradeoff processes amongst requirements.
  - It helps synchronize design engineering disciplines across different systems.

- The architecture engineering process (*499) is a *subset* of the Systems Engineering process (*233).



Engineering Hierarchy

Engineering Concepts

- Engineering *224
  - Systems Engineering *223
    - Systecture (Systems Architecture) *564
      - Data Structures Strategy
      - Application Portfolio Strategy
      - Platform Strategy
      - Methods Strategy
      - Standards Development
    - Program Management
      - Project
  - Other Engineering

Processes

- Architecture Process *499
- Requirements Process *612
- Design Engineering *501
- Evolutionary Project Management (Evo) *355
- Design Process *046
- Impact Estimation *283

Specification Types

- (The) Architecture *192 (Artifacts)
- Architecture Specification *617
- Standards *138
  - Security Standards
  - Interface Standards
  - Requirement Specification Standards
  - Other
- Requirement Specification *508
- Design Specification *586
- Impact Estimation Table
- Evo Step Specification *370
- Evo Plan *322

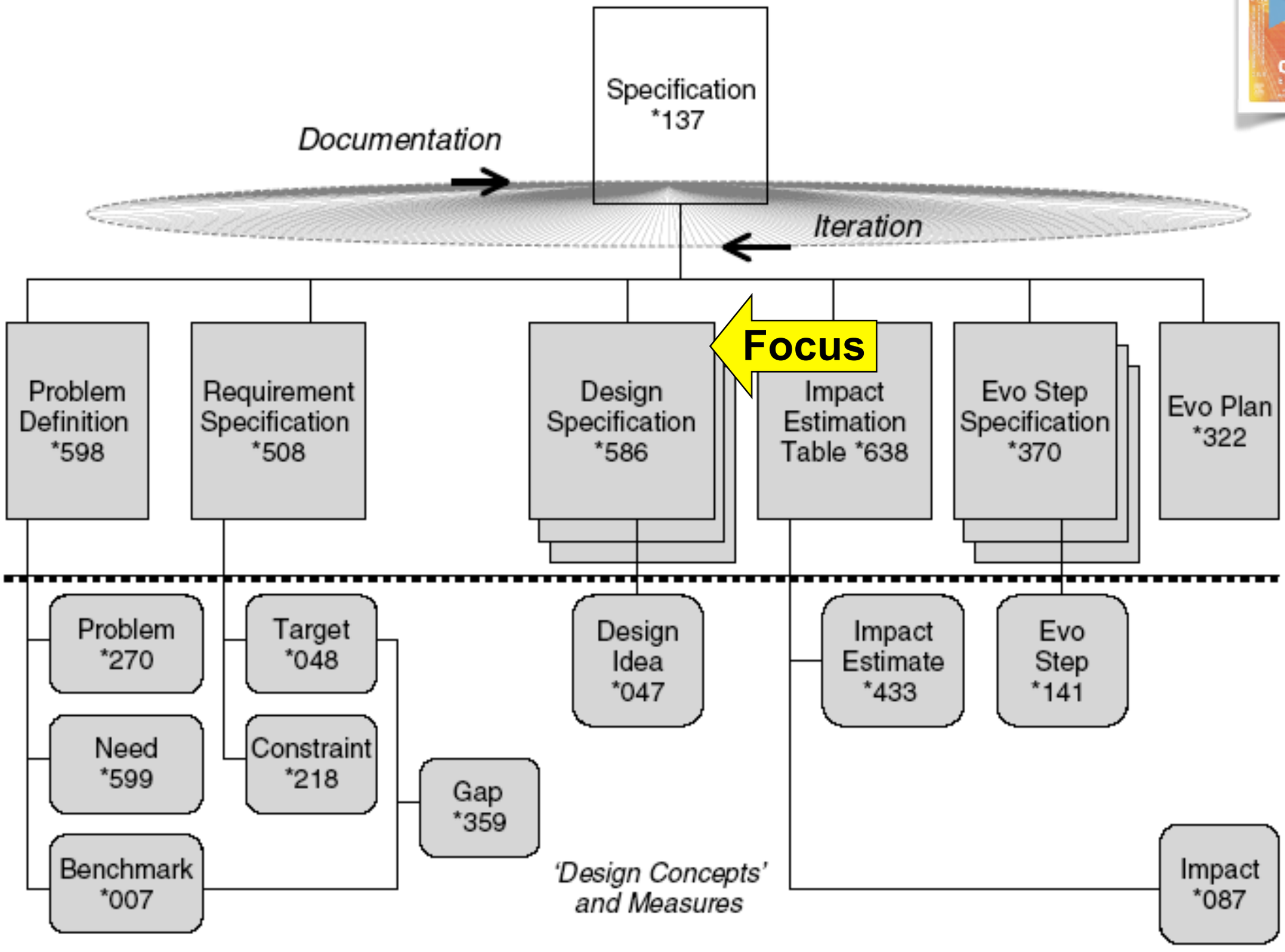
Requirement Concepts for Architects

- Requirement *026
  - Vision *422
  - Function Requirement *074
    - Mission *097
    - Function Target *420
    - Function Constraint *469
  - Performance Requirement *100 (objective)
    - Quality Requirement *453
    - Resource Saving Requirement *622
    - Workload Capacity Requirement *544
    - Performance Target *439 (goal)
      - **Goal *109**
      - Stretch *404
      - Wish *244
    - Performance Constraint *438
      - Fail *098
      - Survival *440
  - Resource Requirement *431
    - Resource Target *436 (budget)
      - **Budget *480**
      - Stretch *404
      - Wish *244
    - Resource Constraint *478
      - Fail *098
      - Survival *440
  - Design Constraint *181
  - Condition Constraint *498

Focus

Focus

Specification Types for Architects



Does IT Architecture (Softecture) have any professional and ethical obligations to their employers and society ?

Some Professional Obligations

1. Push back and help client to articulate requirements fully and clearly

2. Help client to understand the consequences of their requirements

3. Help client to understand why specific architecture is a best choice

4. Help the client to see the many critical dimensions of requirements

5. Quality Control: requirements and design, against best practice rules.

6. Attempt to maximize delivery of real critical values for all resources required

Ethical Obligations

- Be honest and up-front about the limitations of our knowledge of architecture choices

- Do not make assertions without documenting the basis for them

- https://www.nspe.org/resources/ethics/code-ethics

standards of honesty and integrity. Engineering has a direct and vital impact on the quality of life for all people. Accordingly, the services provided by engineers require honesty, impartiality, fairness, and equity, and must be dedicated to the protection

of the public health, safety, and welfare. Engineers must perform under a standard of professional behavior that requires adherence to the highest principles of ethical conduct.

I. Fundamental Canons

Engineers, in the fulfillment of their professional duties, shall:

1. **Hold paramount the safety, health, and welfare of the public.**
2. **Perform services only in areas of their competence.**
3. **Issue public statements only in an objective and truthful manner.**
4. **Act for each employer or client as faithful agents or trustees.**
5. **Avoid deceptive acts.**

The ‘Softect’ Code

© tom@Gilb.com 2017

1. **Explicit**: capture all risks explicitly.

2. **Long Term**: Document long term costs and attributes even when client is not aware or initially interested.

3. **Expert**: Seek specialist knowledge rather than assume or guess

4. **Stakeholder**: Develop deep and broad stakeholder knowledge

5. **Agile**: Learn gradually in small implementation steps, how well architecture works and what it costs. Agile and Lean

6. **Openness**: Develop open-ended architecture so short term and long term changes are easy.

That was the framework.

Here are some technical examples of doing it

Professional Obligations

1. **Push back and help client to articulate critical value requirements fully and clearly**

2. **Help client to understand the consequences of their requirements**

3. **Help client to understand why specific architecture is a best choice**

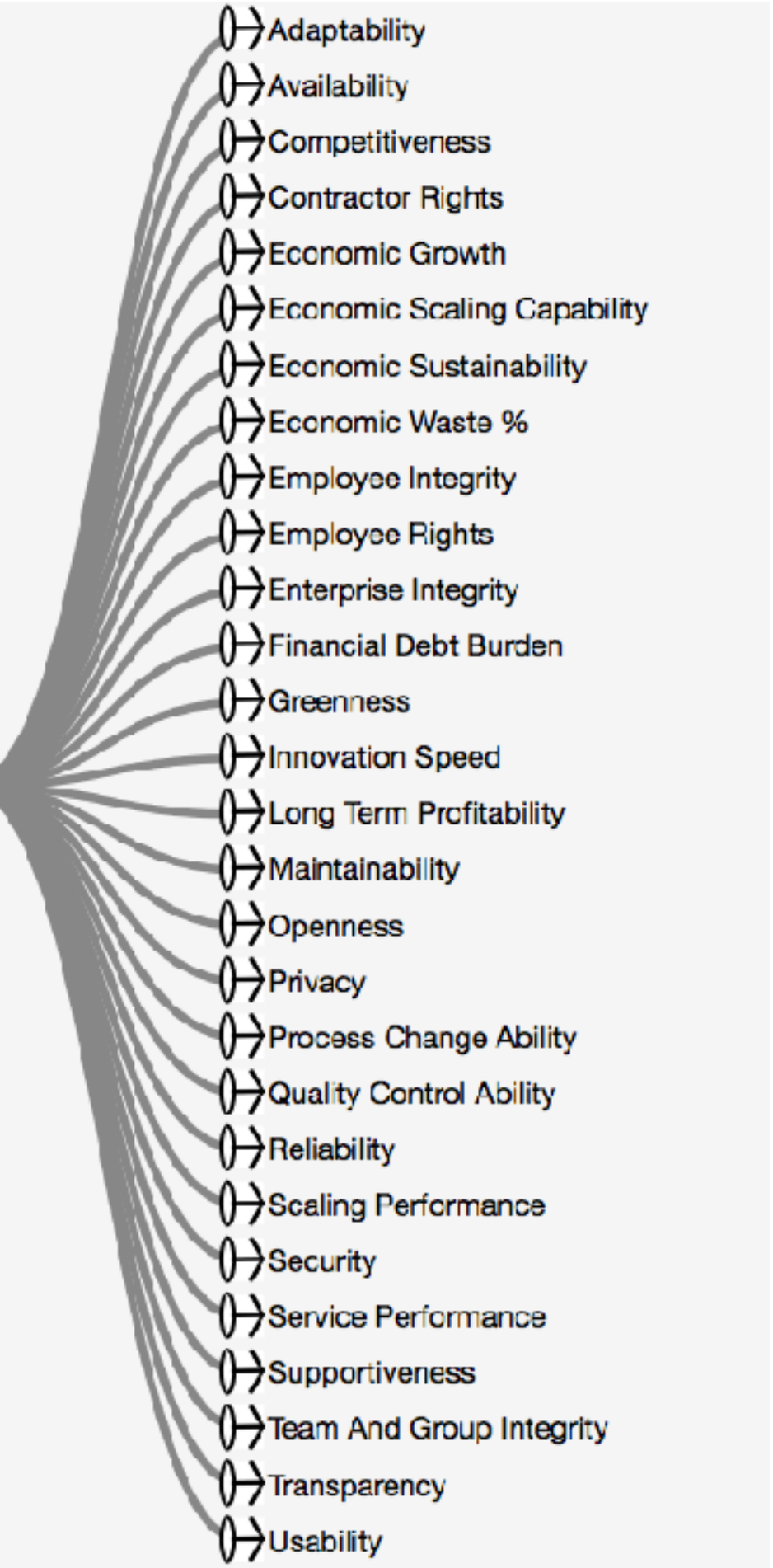
4. **Help the client to see the many critical dimensions of requirements**

5. **Quality Control: requirements and design, against best practice rules.**

6. **Attempt to maximize delivery of real critical values for all resources required**

Professional Obligations

Note: the Presenter Notes contain considerable explanation of what I intend to point out for each slide.



1. Push back and help your client to articulate critical value requirements fully and clearly

1. Push back and help client to articulate critical value requirements fully and clearly

Security

Business Value ***Empty***

(by - 10 minutes ago)

Permalink

0.0.1

Is Part Of: Qualities Assurance

Ambition Level: IT System High levels of security against all forms of attack, by conscious planning and investment

Scale: % [Attack Types] per Year from [Attack Sources] using [Attack Methods] under [Environment Conditions] towards [Organizat

Stakeholders: CTO Chief Technical Officer, Board Of Directors, Enterprise Architects, Quality Assurance Security Director

Status: Level: **1** % Successful Attacks [Attack Types = { <All> }, Attack Sources = { <All> }, Attack Methods = { <All> }, Environment Conditions =

Wish: Level: **0.01** % Successful Attacks [Attack Types = { <All> }, Attack Sources = { <All> }, Attack Methods = { <All> }, Environment Conditions

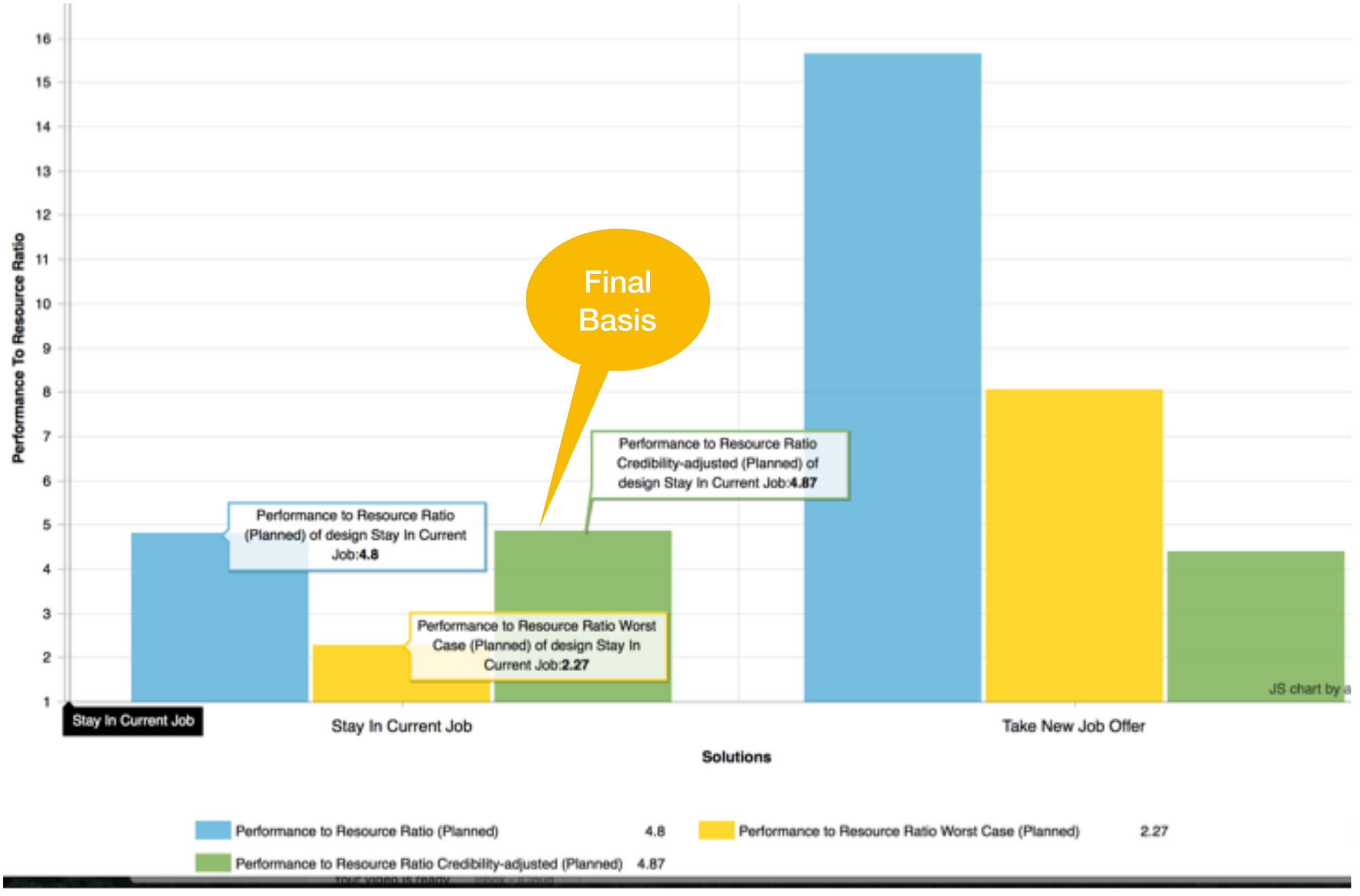
2. Help client to understand the consequences of their requirements

Requirements		NeedsAnd Means Tool	Planguage	Sum
Security Status: **1** → Wish: **0.01** % Success % [Attack Types] per Year from [Atta... [Attack Types = { <All> }, June 2019	Δ: Δ%:	-0.5 **51** %	-1 **101** %	ΣΔ%: **152** %
Sum Of Values:	Σ%:	**51** %	**101** %	
Capital Spend Budget Status: **0** → Budget: **100** % of budget % of a defined monetary budget *No qualifiers* 18th June 2019	Δ: Δ%:	2 **2** %	30 **30** %	ΣΔ%: **32** %
Development Time Status: **0** → Budget: **100** % of Weekly... % of annual weeks needed to successf... *No qualifiers* 50	Δ: Δ%:	3 **3** %	30 **30** %	ΣΔ%: **33** %
Sum Of Development Resources:	Σ%:	**5** %	**60** %	
Value To Cost:		**10.20**	**1.70**	

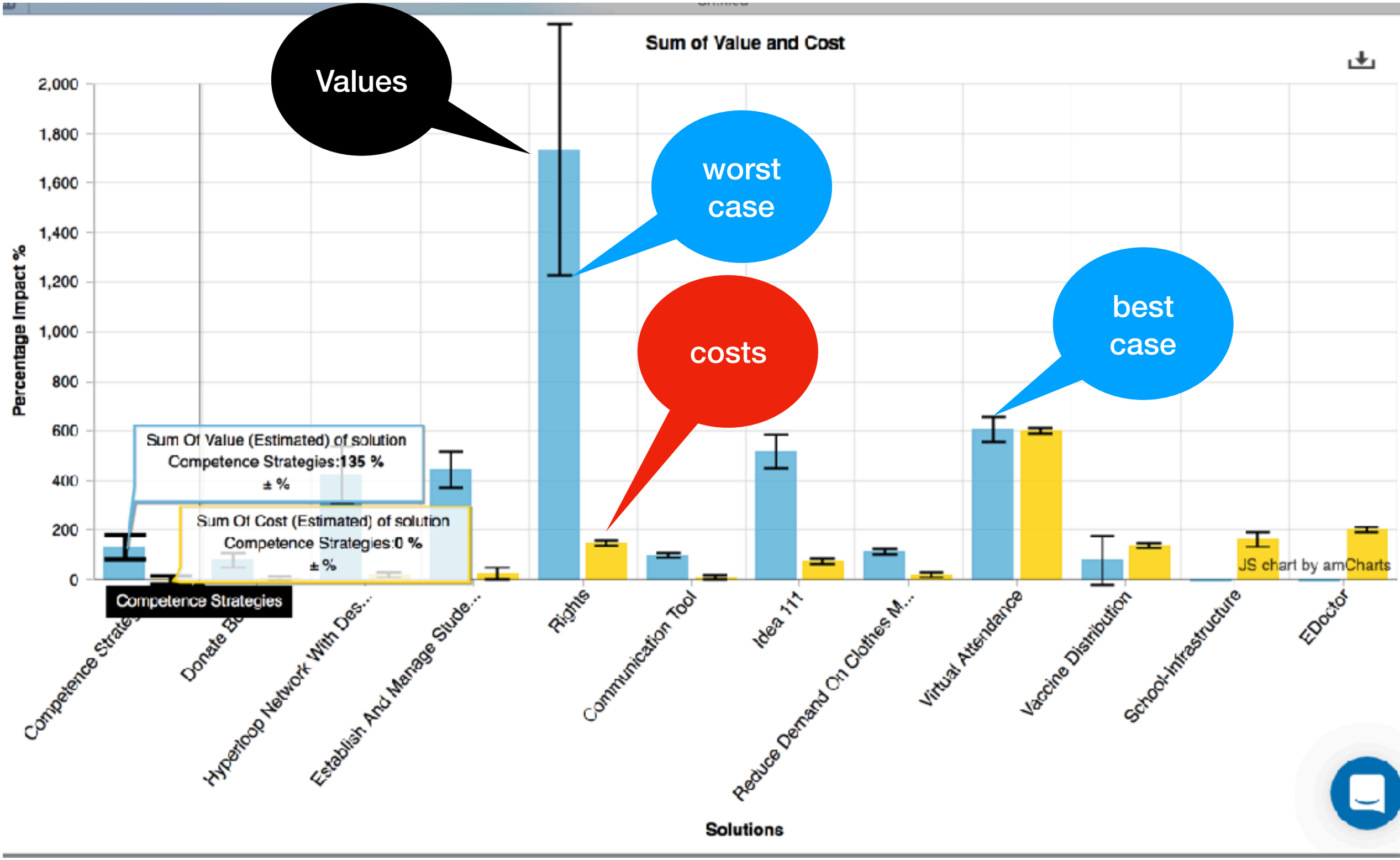
2. Help client to understand the consequences of their requirements

Requirements		NeedsAnd Means Tool	Planguage	Sum
Security Status: **1** → Wish: **0.01** % Success % [Attack Types] per Year from [Atta... [Attack Types = { <All> }, June 2019	Δ: Δ%:	-0.5 **51** %	-1 **101** %	ΣΔ%: **152** %
Sum Of Values:	Σ%:	**51** %	**101** %	
Capital Spend Budget Status: **0** → Budget: **100** % of budget % of a defined monetary budget *No qualifiers* 18th June 2019	Δ: Δ%:	2 **2** %	30 **30** %	ΣΔ%: **32** %
Development Time Status: **0** → Budget: **100** % of Weekly... % of annual weeks needed to successf... *No qualifiers* 50	Δ: Δ%:	3 **3** %	30 **30** %	ΣΔ%: **33** %
Sum Of Development Resources:	Σ%:	**5** %	**60** %	
Value To Cost:		10.20	1.70	

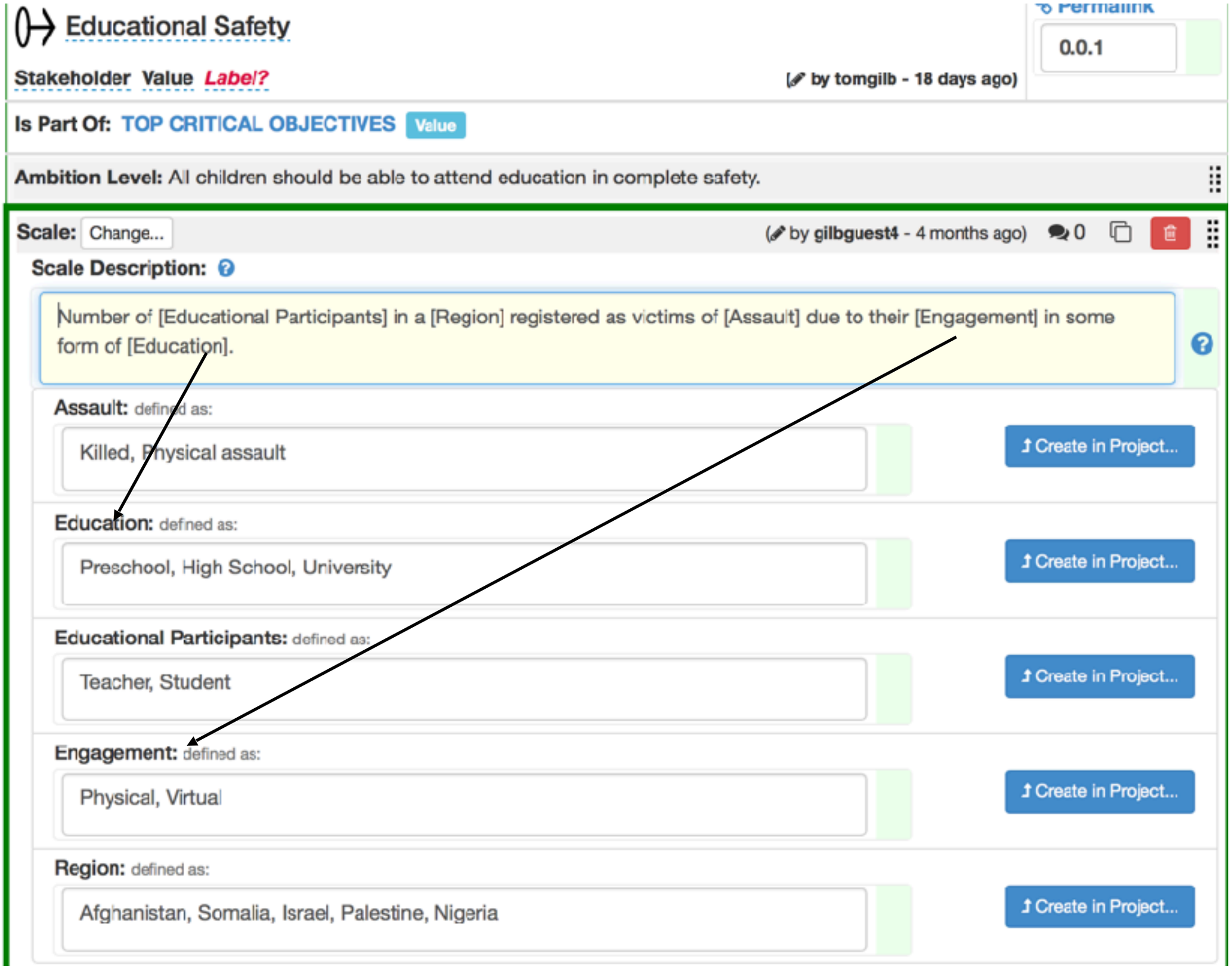
3. Help client to understand why specific architecture is a best choice



3. Help client to understand why specific architecture is a best choice



4. Help the client to see the many critical dimensions of requirements



4. Help the client to see the many critical dimensions of requirements

Educational Safety

Permalink 0.0.1

Stakeholder Value *Label?* (by tomgilb - 18 days ago)

Is Part Of: TOP CRITICAL OBJECTIVES Value

Ambition Level: All children should be able to attend education in complete safety.

Scale: Number of [Educational Participants] in a [Region] registered as victims of [Assault] due to their [Engagement] in some form ...

Status: Level: **185k** Persons per year [Educational Participants = { <All> }, Region = { Afghanistan }, Assault = { <All> }, Engagement = { Physical },

Wish: Change... (by **tomgilb** - 18 days ago) 0

Scale Level: Persons per year

100000

By When:

2020

Date format: yyyy (e.g. "2017")

Qualifiers: Copy from...

[Educational Participants] =

× Teacher × Student

[Region] =

× Afghanistan

[Assault] =

× Killed × Physical assault

[Engagement] =

× Physical

[Education] =

× High School

+Add additional qualifier

Source:

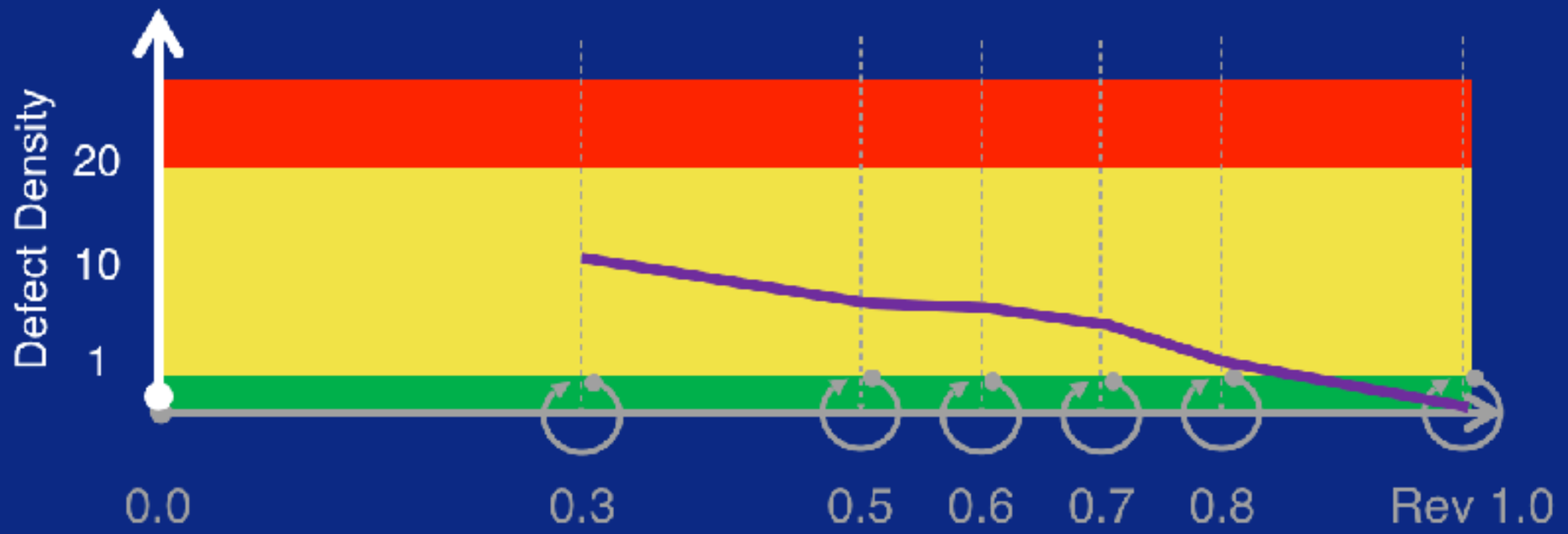
Type something

Add Comment...

Stakeholders: Covert Schools, Internet Based Community Group

5. Quality Control: requirements and design, against best practice rules.

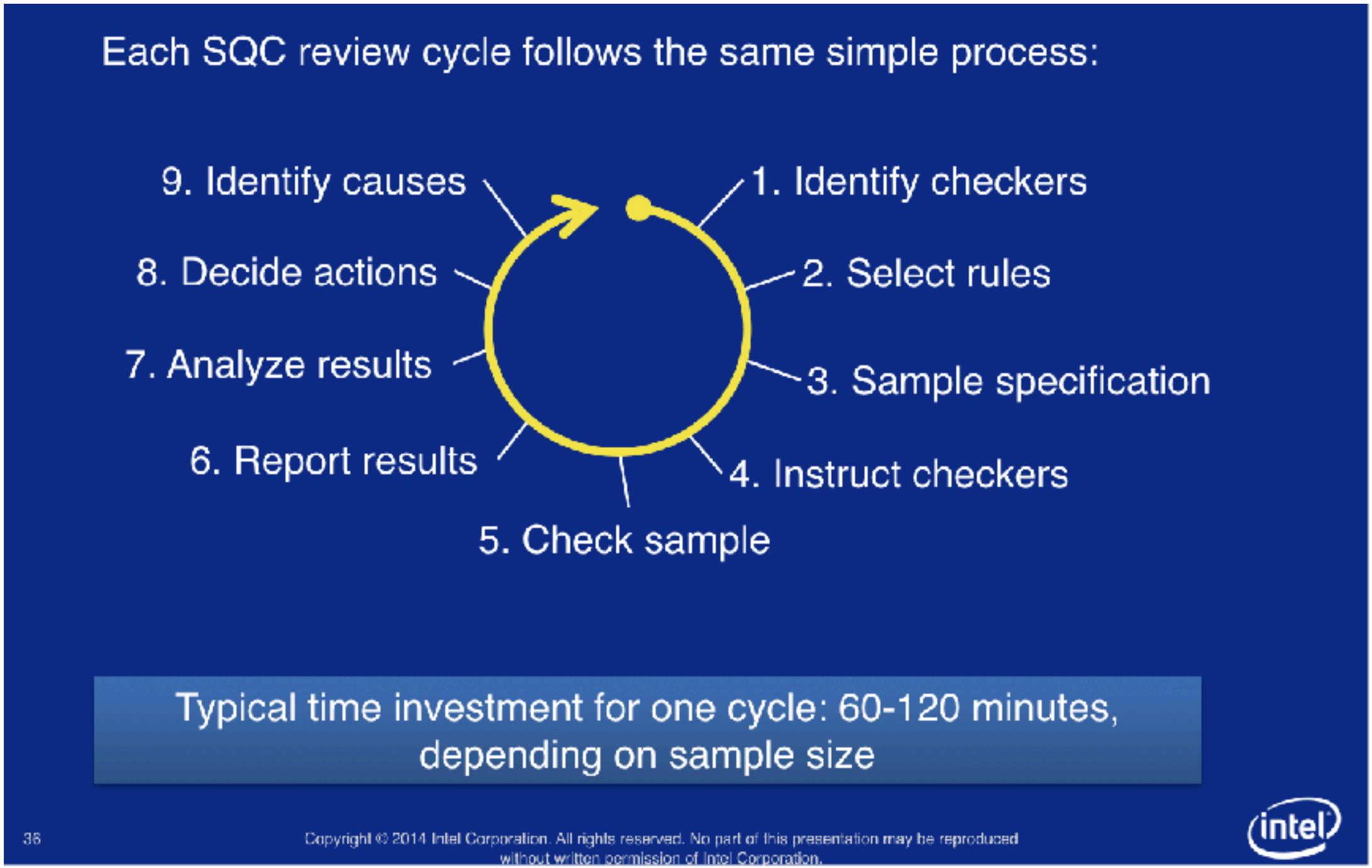
A team in Client BIOS used SQC to reduce requirements defect density by 98% over six cycles:



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

Copyright © 2014 Intel Corporation. All rights reserved. No part of this presentation may be reproduced without written permission of Intel Corporation.

5. Quality Control: requirements and design, against best practice rules.



5. Quality Control: requirements and design, against best practice rules.

Over a series of SQC applications, the team was able to reduce defect density dramatically

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

233% to 300% Engineering productivity increase, as a result: Terzakis, Intel

64

Copyright © 2014 Intel Corporation. All rights reserved. No part of this presentation may be reproduced without written permission of Intel Corporation.

intel

John Terzakis, Intel Boston

The Impact of Requirements on Software Quality across Three Product Generations

John Terzakis

Intel Corporation, USA
john.terzakis@intel.com

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

***Index Terms*—Requirements specification, requirements defects, reviews, software defects, software quality, multi-generational software products.**

II. Product Backgrounds

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

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

After analyzing the software defect data from the Gen 1 release, the Gen 2 team identified requirements as a key improvement area. A requirements Subject Matter Expert (SME) was assigned to assist the team in the elicitation, analysis, writing, review and management of the requirements for the second generation product. The SME developed a plan to address three critical requirements areas: a central repository, training, and reviews. A commercial Requirements Management Tool (RMT) was used to store all product requirements in a database. The data model for the requirements was based on the Planguage keywords created by Tom Gilb [2]. The RMT was configured to [illegible] formatted Product Requirements Document [illegible] revision control. Architecture specifications, des[illegible]

usin

https://www.dropbox.com/sh/cs9hke3uvgg4gp3/AACadHeI95lZpHzVqGKXSXDra?dl=0
See also Gilb: Value Planning book for more detail on this SQC & Planguage method. https://www.gilb.com/store/2W2zCX6z

6. Attempt to maximize delivery of real critical values for all resources required

Table Top

TOP LEVEL VALUE TABLE

Settings... | Add | Sort | Duplicate... | Δ: INCREMENTAL | Show Sidebar

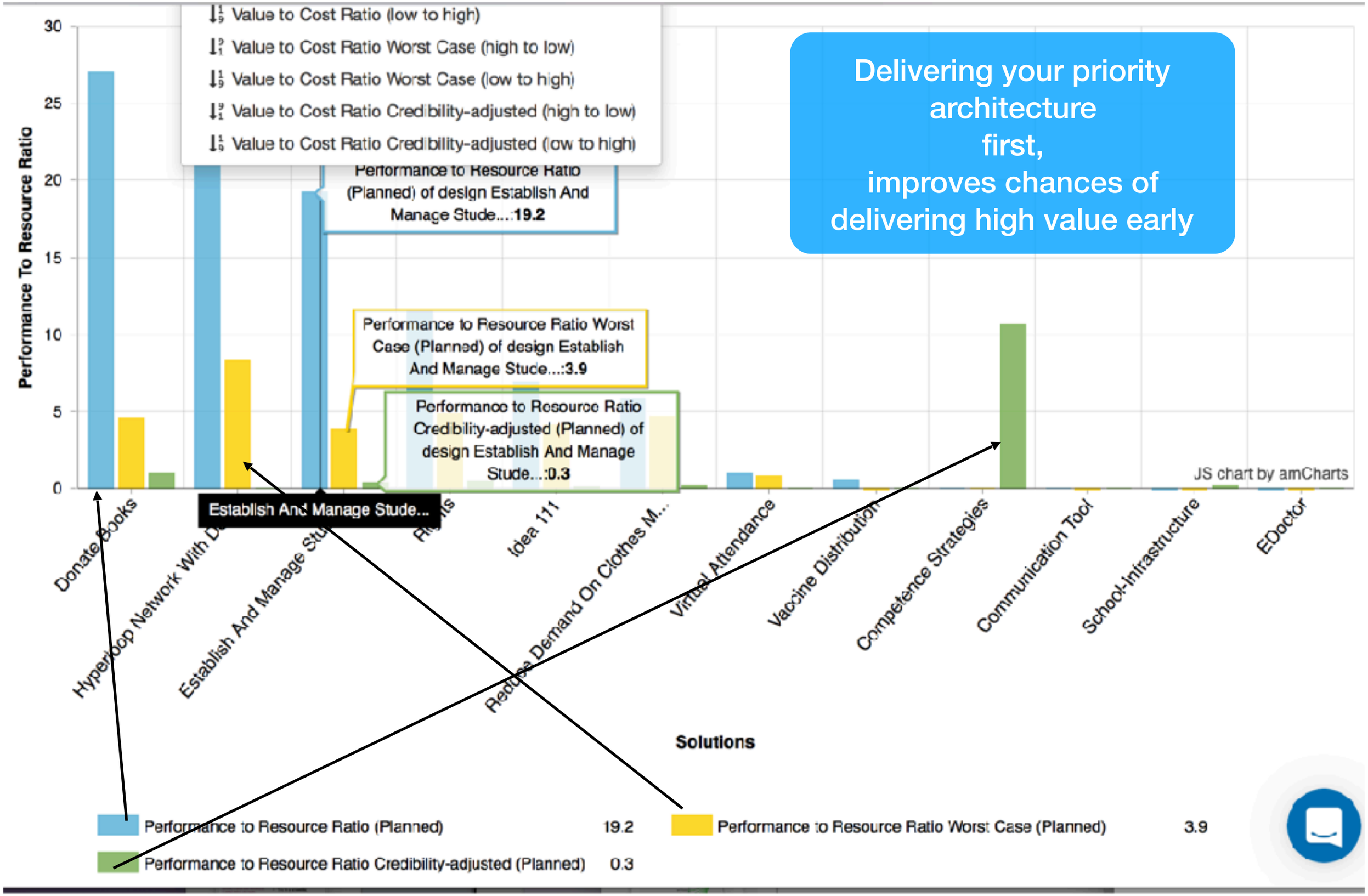
Requirements	Competence Strat...	Communication Tool	Donate Books	Rights	Establish And Ma...	Reduce De
Educational Safety Δ: Status: **185k** → Wish: **100k** Per... Δ%: Number of [Educational Participants]... [Educational Participants = { Teacher...] 2020	**0** ± 0 **0** ± 0 % *0 % (x 0.0)*	**0** ± 0 **0** ± 0 % *0 % (x 0.0)*	**0** ± 0 **0** ± 0 % *0 % (x 0.1)*	**10k** ± 20k **-12** ± 24 % *0 % (x 0.0)*	**0** ± 0 **0** ± 0 % *0 % (x 0.0)*	**0** ± 0 **0** ± 0 % *0 % (x 0.0)*
Decision Influence Δ: Status: **0** → Wish: **100** Percent Δ%: % of achieved [Number of members] wi... [Number of members = { 10.000.000 }, ...] 1st January 2025	**20** ± 5 **20** ± 5 % *4 % (x 0.2)*	**0** ± 5 **NaN** ± 5 % *0 % (x 0.6)*	**0** ± 1 **0** ± 1 % *0 % (x 0.0)*	Credibility - weighted % scale impact (= % scale impact x credibility rating) *0 % (x 0.0)*	?%: *0 % (x 0.0)*	*0 % (x 0.0)*
Accident Emergency Healthc... Δ: Status: **8** → Wish: **6** Minutes Δ%: Time required to get by [Transportat... [Transportation = { Ambulance }, 13th March 2019	**????** ± 0 **0** ± 0 % *0 % (x 0.0)*	**0** ± 0 **0** ± 0 % *0 % (x 0.0)*	**0** ± 0 **0** ± 0 % *0 % (x 0.0)*	**-5** ± 1 **250** ± 50 % *0 % (x 0.0)*	**0** ± 0 **0** ± 0 % *0 % (x 0.0)*	**0** ± 0 **0** ± 0 % *0 % (x 0.0)*
Youth Literacy Δ: Status: **50** → Wish: **75** % o... Δ%: % of [Youths] considered literate in... [Youths = { Teen }, Areas = { A...] 13th March 2025	**5** ± 5 **20** ± 20 % *0 % (x 0.0)*	**0** ± 0 **0** ± 0 % *0 % (x 0.0)*	**10** ± 0 **40** ± 0 % *4 % (x 0.1)*	**55** ± 10 **220** ± 40 % *0 % (x 0.0)*	**10** ± 5 **40** ± 20 % *0 % (x 0.0)*	**4** ± 1 **16** ± 4 % *8 % (x 0.5)*
Save Lives Δ: Status: **60** → Wish: **70** [Li... Δ%: [Lives saved] [Period of time] [Quality] [Lives saved = { yes },	**????** ± 0 **0** ± 0 % *0 % (x 0.0)*	**0** ± 0 **0** ± 0 % *0 % (x 0.0)*	**0** ± 0 **0** ± 0 % *0 % (x 0.0)*	**2** ± 1 **20** ± 10 % *0 % (x 0.0)*	**25** ± 5 **250** ± 50 % *0 % (x 0.0)*	**0** ± 0 **0** ± 0 % *0 % (*

6. Attempt to *maximize delivery* of *real critical values* for *all resources required*

Table Bottom

	Competence Strat...	Communication Tool	Donate Books	Rights	Establish And Ma...	Reduce De
% of defined [Rights] of defined [C... [Rights = { National Law Rights }, 13th March 2020	*0 % (x 0.0)*	*0 % (x 0.0)*	*0 % (x 0.0)*	*47 % (x 0.7)*	*0 % (x 0.0)*	*0 % (x 0.0)*
Affordability Of Education Δ: Status: **58** → Wish: **70** % o... Δ%: % of [Children] in given [Areas] tha... [Children = { Kid }, Areas = { ...] 13th March 2017	**????** ± 0 **0** ± 0 % *0 % (x 0.0)*	**0** ± 0 **0** ± 0 % *0 % (x 0.0)*	**1** ± 1 **8** ± 8 % *0 % (x 0.0)*	**15** ± 2 **125** ± 17 % *100 % (x 0.8)*	**0** ± 0 **0** ± 0 % *0 % (x 0.0)*	**0** ± 0 **0** ± 0 % *0 % (x 0.0)*
Sum Of Values: Σ%: Credibility - adjusted: Σ?%:	**135** ± 58 % *11 %*	**0** ± 5 % *0 %*	**81** ± 58 % *4 %*	**1732** ± 1012 % *14*	**441** ± 151 % Uncertainty of estimate of scale level impact	**115** ± 12 % *8 %*
BUDGET EXPENDITURE Δ: Status: **0** → Budget: **100** %$$ Δ%: % OF financial Bucget Allocation used ?%: [New Qualifier 1 – { Qualifier Value]] ?	**????** ± 0 **0** ± 0 % *0 % (x 0.0)*	**10** ± 0 **10** ± 0 % *20 % (x 0.0)*	**1** ± 1 **1** ± 1 % *1 % (x 0.8)*	**50** ± 0 **50** ± 0 % *100 % (x 0.0)*	Δ:**3** ± 2 Δ%: **3** ± 2 % ?%: *6 % (x 0.0)*	**10** ± 2 **10** ± 2 % *20 % (x 0.0)*
CALENDAR TIME TO DEADLINE **????** ± 0 Status: **0** → Budget: **100** Δ%: *No qualifiers* ?% 14th March 2017	**0** ± 0 % *0 % (x 0.0)*	**0** ± 0 **0** ± 0 % *0 % (x 0.0)*	**2** ± 1 **2** ± 1 % *3 % (x 0.5)*	**100** ± 0 **100** ± 0 % *200 % (x 0.0)*	**20** ± 50 **20** ± 50 % *40 % (x 0.0)*	**10** ± 0 **10** ± 0 % *20 % (x 0.0)*
Sum Of Development Resources: Σ%: Credibility - adjusted: Σ?%:	**0** ± 0 % *0 %*	**10** ± 0 % *20 %*	**3** ± 2 % *4 %*	**150** ± 0 % *300 %*	**23** ± 52 % *46 %*	**20** ± 2 % *40 %*
Value To Cost:	0.00	0.00	27.00	11.50	19.20	5.80
Ratio (Worst Case)	0.00	-0.50	4.60	4.80	3.90	4.70
Ratio (Cred. - adjusted)	10.60	0.00	1.00	0.50	0.30	0.20
Ratio (Worst Case Cred. - adjusted)	7.00	-3.00	1.30	124.20	6.00	6.00

6. Attempt to maximize delivery of real critical values for all resources required



Heavy Stuff?

- **Study the papers, blogs, videos, cases at Gilb.com**
- **Read a book, Value Planning, Competitive Engineering**
- **Come of our BCS Courses**
- **Give yourself time for mastery: maybe 10,000 hours**
- **3 of 300 Architects claimed they estimated costs (London, SW Arch. Conf.)**
- **1 of 300 claimed they estimated technical impacts (security)**

www.Gilb.com

Value Planning (digital manuscript)
https://www.gilb.com/store?tag=books
Purchase Offer Coupon Code: VP20 (€20 off)

Weekly **Blog** (Based on Value Planning book): www.Gilb.com/blog

http://gilb.com/p/principles
This will allow you to sign up for Gilb Principles videos.

OUR WEBSITE FOR ALL DOWNLOADS
http://concepts.gilb.com/file24
This will contain all old papers, slides etc

BCS Courses at
http://www.bcs.org/category/10136

What is Wrong with Software Architecture. My Keynote in London Oct 2013

(If you want a tool to understand our ideas, and share with others, in some depth
https://www.youtube.com/watch?v=HNasoyrxzy8
1.5 Hours
link tested May 25 2016
SHORTER VERSION OSLO JAVAZONE
https://vimeo.com/28763240

Enterprise Architecture Ethical Obligations

- **Be honest and up-front about the limitations of our knowledge of architecture choices**
- **Do not make assertions without documenting the basis for them**

The ‘Softect’ Code

© tom@Gilb.com 2017

1. **Explicit: capture all risks explicitly.**

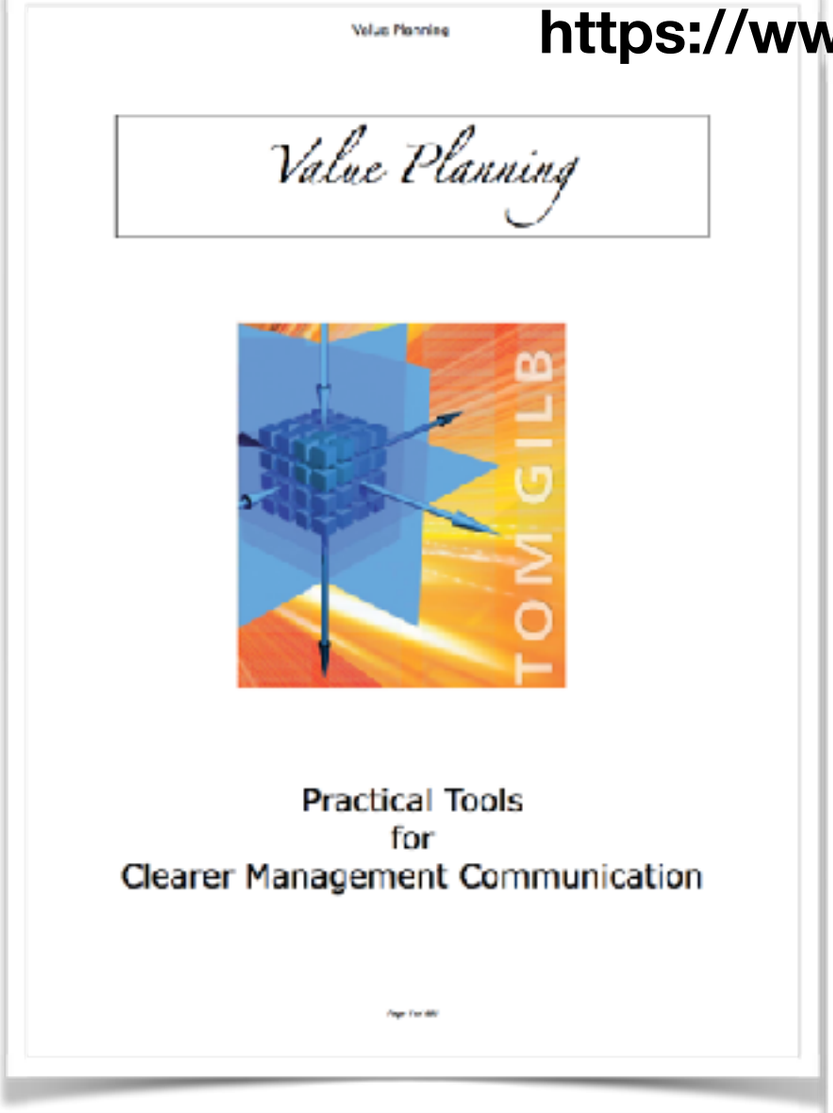
2. **Long Term: Document long term costs and attributes even when client is not aware or initially interested.**

3. **Seek specialist knowledge rather than assume or guess**

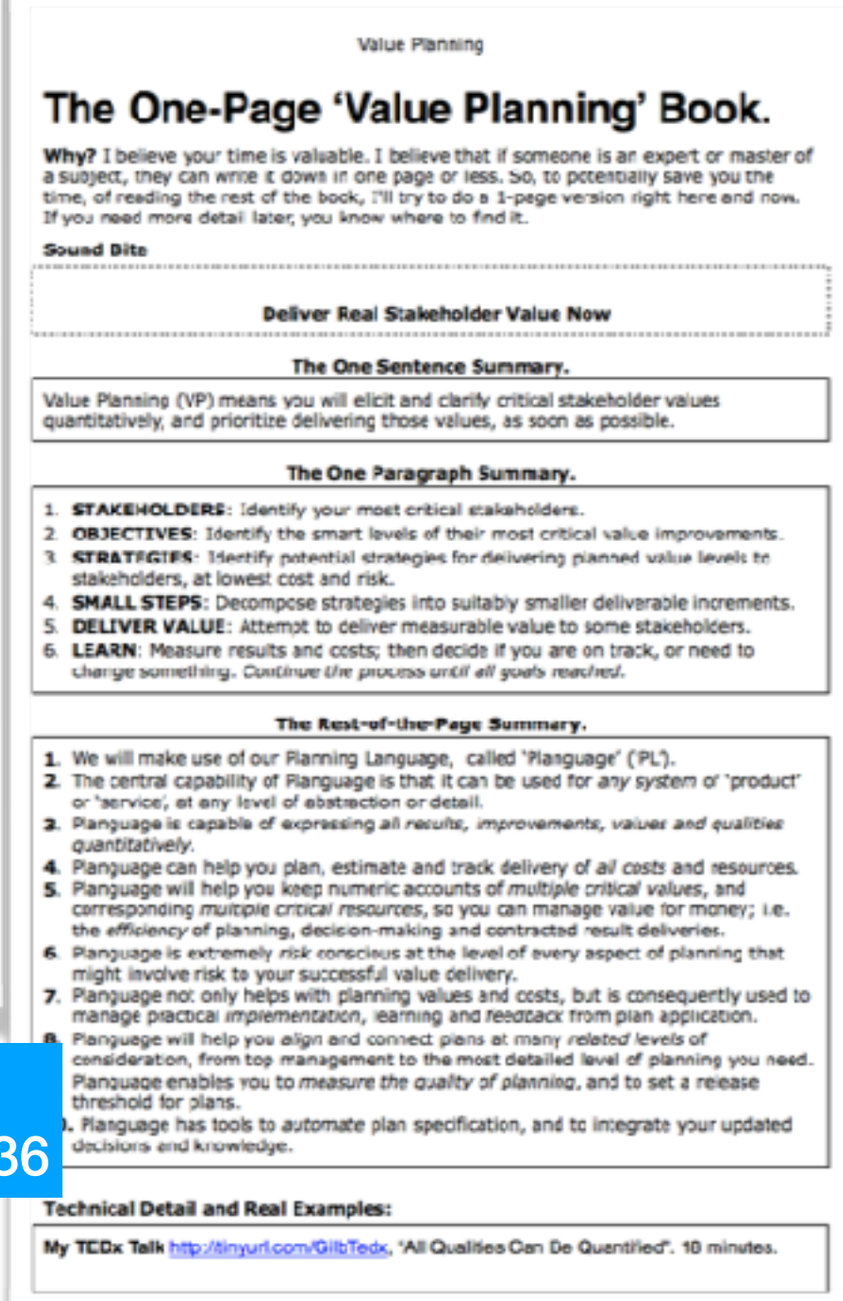
4. **Stakeholder: Develop deep and broad stakeholder knowledge**

5. **Agile: Learn gradually in small implementation steps, how well architecture works and what it costs. Agile and Lean**

6. **Openness: Develop open-ended architecture so short term and long term changes are easy.**



https://www.gilb.com/store/2W2zCX6z



GILB COURSES AT BCS
http://www.bcs.org/category/10136

#BCSEASG

www.gilb.com/p/competitive-engineering

needsandmeans.com = the tool

THESE SLIDES = tinyurl.com/Gilb17July