

# IT Value Planning

Real Stakeholder Business Value, Explicit Planning  
and Soft Values Quantification

**Tom Gilb**



**BCS – The North London Branch and Elite Group**  
**At BCS, 5 Southampton St, Covent Garden**

Thursday, 11 June 2015

<http://www.gilb.com/dl845>

# Some Practical Cases from my practice

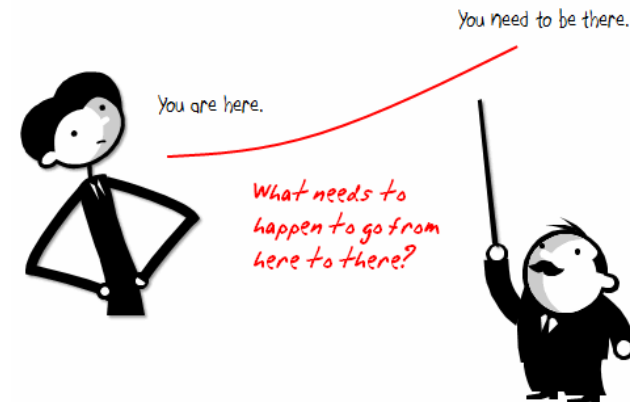
(made suitably anonymous)

# Real Example

## “Platform Rationalisation Initiative”

### Main Objectives

### London Multinational Bank



- Rationalize into a smaller number of core processing platforms. This cuts technology spend on duplicate platforms, and creates the opportunity for operational saves. Expected 60%-80% reduction in processing cost to Fixed Income Business levies.
- International Securities on one platform, Fixed Income and Equities (Institutional and PB).
- Global Processing consistency with single Operations In-Tray and associated workflow.
- Consistent financial processing on one Accounting engine, feeding a single sub-ledger across products.
- First step towards evolution of “Big Ideas” for Securities.
- Improved development environment, leading to increased capacity to enhance functionality in future.
- Removes duplicative spend on two back office platforms in support of mandatory message changes, etc.

# Basic Quality Control Rules for Top Level Objectives

- **CLEAR:** Every word and phrase should be clear enough to allow objective test of a delivery. (we need to know exactly what is required and expected)
- **UNAMBIGUOUS:** Every word and phrase should be unambiguous to all potential intended readers. (no different than intended interpretations should be possible)
- **QUANTIFIED QUALITY:** all qualities (good things we want to improve) shall be expressed quantitatively.
- After we started the exercise I regretted not adding the usual rule:
- **4. NO DESIGN: objectives shall not be expressed in terms of a design or architecture**
  - (a 'means' to reach the 'real' objective), when it is possible and is our real intent, to express the improvements in terms of quality, performance, and cost that are expected, instead.



Potential consequence  
of major defects  
in architecture specs

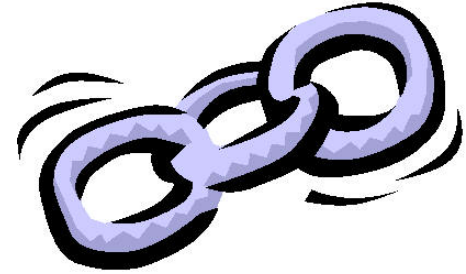
# COUNT THE NUMBER OF MAJOR 'DEFECTS' (RULES VIOLATIONS)

Rules Reminder: **1. Clear, 2. Unambiguous, 3. Quantified Qualities, 4. No Design/Architecture**



- **Rationalize into a smaller number of core processing platforms. This cuts technology spend on duplicate platforms, and creates the opportunity for operational saves. Expected 60%-80% reduction in processing cost to Fixed Income Business lines.**
- **International Securities on one platform, Fixed Income and Equities (Institutional and PB).**
- **Global Processing consistency with single Operations In-Tray and associated workflow.**
- **Consistent financial processing on one Accounting engine, feeding a single sub-ledger across products.**
- **First step towards evolution of “Big Ideas” for Securities.**
- **Improved development environment, leading to increased capacity to enhance functionality in future.**
- **Removes duplicative spend on two back office platforms in support of mandatory message changes, etc.**

## RULE 4. No Design/Architecture



- Rationalize into a smaller number of core processing platforms. *This cuts technology spend* on duplicate platforms, and *creates the opportunity* for operational saves. Expected 60%-80% reduction in processing cost to Fixed Income Business lines.
- International Securities on one platform, Fixed Income and Equities (Institutional and PB).
- Global Processing consistency with single Operations In-Tray and associated workflow.
- Consistent financial processing on one Accounting engine, feeding a single sub-ledger across products.
- First step towards evolution of “Big Ideas” for Securities.
- *Improved development environment*, leading to increased capacity to enhance functionality in future.
- Removes duplicative spend on two back office platforms in support of mandatory message changes, etc.

# Agile Specification Quality Control Results



Reported major defects =

- Last week: 15, 17, 21
  - Today: 18, 15, 15, 13
  - Others less
- Estimated approx. Total defects found by a small team (2-4 people) =  $36 \pm 6$ 
    - 2x highest found.
  - **Estimated** approx. Total Majors in the 110 words =  $100 \pm 10$ 
    - (3x group total. 30% effectiveness of team)
  - Estimated approximate total defects in normalized page (300 words) =  $280 \pm 20$   
(Majors in 110 words x 3)



# How can we improve such bad specification?..... Use 'Planguage'



## Development Capacity:

**Version:** 3 Sept 2009 16:26

**Type:** Main <Complex/Elementary> Objective for a project.

**Ambition Level:** radically increase the capacity for developers to do defined tasks. <- Tsg

**Scale:** the Calendar Time for defined [Developers] to Successfully carry out defined [Tasks].

**Owner:** Tim Fxxx

**Calendar Time:** defined as: full working days within the start to delivery time frame.

**Past** [2009, {Bxx, Lxx, Gxx}, If QA Approved Processes used, Developer = Architect, Task = Draft Architecture ]: **15 days**  $\pm 4$  ?? <- Rob

**Goal** [2011, { Bxx, Lxx, Gxx }, If QA Approved Processes used, Developer = Architect, Task = Draft Architecture ]: **1.5 days**  $\pm 0.4$  ?? <- Rob

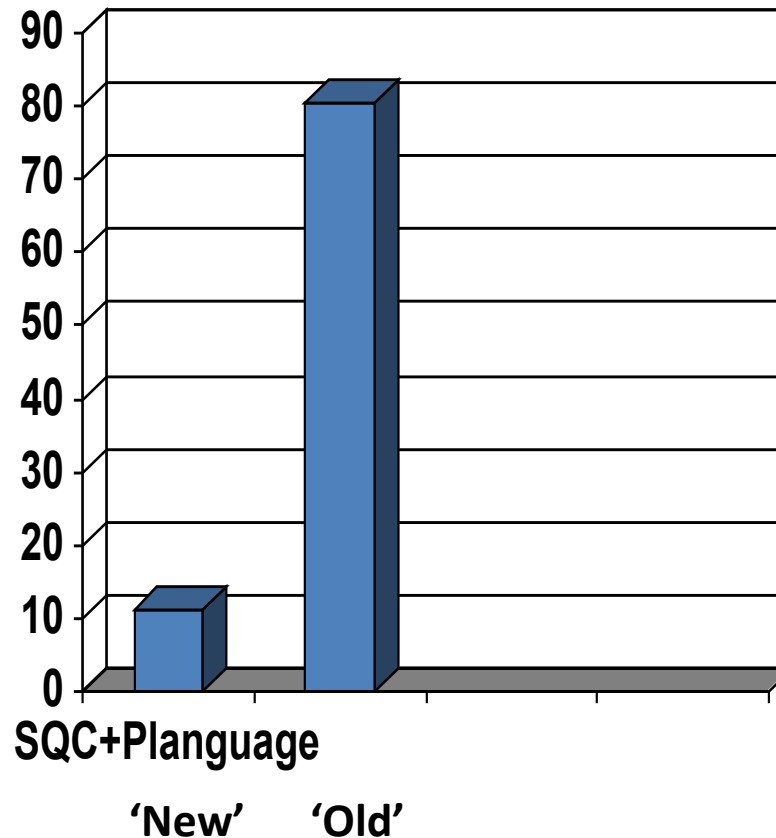
**Justification:** Really good architects are very scarce so we need to optimize their use.

**Risks:** we use effort that should be directed to really high volume or even more critical areas (like Main Objective).



# Reducing unintelligible IT requirements from 80/page to 10/page in 6 months at a London Bank using Specification Quality Control (SQC) / Inspection + Planguage Requirements

Major defects per page  
found on first SQC / Inspection



See Slide Note for details

# Quality Management using Planguage and SQC

Application of SQC by a software team resulted in the following defect density reduction in requirements over several months:

Source Eric Simmons, [erik.simmons@intel.com](mailto:erik.simmons@intel.com) 25 Oct 2011  
[http://selab.fbk.eu/re11\\_download/industry/Terzakis.pdf](http://selab.fbk.eu/re11_download/industry/Terzakis.pdf)

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%

Downstream benefits:

- Scope delivered at the Alpha milestone increased 300%, released scope up 233%
- Software defects reduced by ~50%
- Defects that did occur were resolved in far less time on average
- Teams typically exit with densities ranging from 5 majors per page (600 words) to 1 defect in a couple of pages.

# The Main Ideas

1. MANAGEMENT OF IMPACTS / CHANGES REQUIRED
  - Quantify values
2. MANAGEMENT OF PLANS
  - Quantify plans
3. MANAGEMENT OF PROFIT
  - Value for money
4. MANAGEMENT OF RISKS
  - Do a little, measure, adjust
5. PRIORITIZATION
  - Highest value early. Best ROI

# Some Practical Tools

## Planguage: a value-planning language

### – A paper on Planguage

- <http://www.gilb.com/dl831>

### – The Planguage Handbook

### **“Competitive Engineering”**

- <https://www.dropbox.com/sh/kxl3g8hm7vlbthr/AAC5gdzGAJqocX51q6vV2A92a?dl=0>
- <http://www.gilb.com//dl540>

## Templates, Principles, Processes and Terminology



**“This stuff works”**  
Erik Simmons, Intel

# Some More Practical Experiences with Managing Value

# Large Bank Project's Top 10 Requirements

## Quantifying the most-critical project objectives on day 1, on 1 page

**P&L-Consistency****T P&L**: **Scale**: total adjustments btw Flash/Predict and Actual (T+1) signed off P&L. per day. **Past 60** **Goal**: **15**

**Operational-Control****Timely****Trade-Bookings** **Scale**: number of trades per day that are not booked on trade date. **Past** [April 20xx] **20** ?

**Speed-To-Deliver**: **Scale**: average Calendar days needed from New Idea Approved until Idea Operational, for given Tasks, on given Markets.

**Past** [2009, Market = EURex, Task =Bond Execution] **2-3 months** ?

**Goal** [Deadline =End 20xz, Market = EURex, Task =Bond Execution] **5 days**

**Front-Office-Trade-Management-Efficiency** **Scale**: Time from Ticket Launch to trade updating real-time risk view

**Past** [20xx, Function = Risk Mgt, Region = Global] ~ **80s +/- 45s** ??

**Goal** [End 20xz, Function = Risk Mgt, Region = Global] ~ **50% better?**  
Managing Risk – Accurate – Consolidated – Real Time

**Operational-Control**: **Scale**: % of trades per day, where the calculated economic difference between OUR CO and Marketplace/Clients, is less than “1 Yen”(or equivalent).

**Past** [April 20xx] **10%** change this to 90% NH **Goal** [Dec. 20xy] **100%**

**Operational-Control****Consistent**: **Scale**: % of defined [Trades] failing full STP across the transaction cycle. **Past** [April 20xx, Trades=Voice Trades] **95%**

**Past** [April 20xx, Trades=eTrades] **93%**

**Goal** [April 20xz, Trades=Voice Trades] **<95 ± 2%>**

**Goal** [April 20xz, Trades=eTrades] **98.5 ± 0.5 %**

**Risk****Cross-Product** **Scale**: % of financial products that risk metrics can be displayed in a single position blotter in a way appropriate for the trader (i.e. – around a benchmark vs. across the curve).

**Past** [April 20xx] **0%** 95%. **Goal** [Dec. 20xy] **100%**

**Risk****Low-latency** **Scale**: number of times per day the intraday risk metrics is delayed by more than 0.5 sec. **Past** [April 20xx, NA] **1%** **Past** [April 20xx, EMEA] ??% **Past** [April 20xx, AP] **100%** **Goal** [Dec. 20xy] **0%**  
Risk.Accuracy

**Risk****user-configurable** **Scale**: ??? pretty binary – feature is there or not – how do we represent?

**Past** [April 20xx] **1%** **Goal** [Dec. 20xy] **0%**

**Operational Cost Efficiency** **Scale**: <Increased efficiency (Straight through processing STP Rates )>

**Cost-Per-Trade** **Scale**: % reduction in Cost-Per-Trade

**Goal** (EOY 20xy, cost type = I 1 – REGION = ALL) **Reduce cost by 60%** (BW)

**Goal** (EOY 20xy, cost type = I 2 – REGION = ALL) **Reduce cost by x %**

**Goal** (EOY 20xy, cost type = E 1 – REGION = ALL) **Reduce cost by x %**

**Goal** (EOY 20xy, cost type = E 2 – REGION = ALL) **Reduce cost by 100%**

**Goal** (EOY 20xy, cost type = E 3 – REGION = ALL) **Reduce cost by x %**

**Operational-Control****Timely****End&OvernightP&L** **Scale**: number of times, per quarter, the P&L information is not delivered timely to the defined [Batch-Run].

**Past** [April 20xx, Batch-Run=Overnight] **1** **Goal** [Dec. 20xy, Batch-

Run=Overnight] **<0.5>** **Past** [April 20xx, Batch-Run= T+1] **1** **Goal** [Dec. 20xy, Batch-Run=End-Of-Day, Delay<1hour] **1**

**Operational-Control****Timely****IntradayP&L** **Scale**: number of times per day the intraday P&L process is delayed more than 0.5 sec.

# ON STABILITY OF 'REAL REQUIREMENTS' AND INSTABILITY OF 'DESIGN' AND 'ARCHITECTURE'



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 over one weekend for 800 users worldwide**, and **was seen as a big success by the sponsoring stakeholders**.

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

# Richard Smith's Planning Tool which we are using on BCS Courses



needsandmeans.mod.bz

Tom Gilb & K...ents-Material appleinsider.com Google Docs TOM'S NET Services Resources » NORSKE STEDER Travel 4 TOM Social Sites NEWS ALLE AND

Startup Planning Course Doc - Dropbox Needs & Means - Demo

needs&means Specifications Impact Tables Documents Glossary Follow Me Help

Home / Impact Tables / IET-6PGBWPE

**BCS.Managing-Software-Technology**

Requirements	BCS.Copies-Of-CE...	BCS.Evo-Process	BCS.Simple-Standards	BCS.Project-Star...	Sum
<b>BCS.Software-Productivity</b> Increase from 3.5 to 5 kNCSS By end of December 2015	0.4 kNCSS 27 % 27 %	2 kNCSS 133 % 160 %	0.2 kNCSS 13 % 173 %	0.5 kNCSS 33 % 206 %	206 %
<b>BCS.Lead-Time</b> Decrease from 20 to 10 Months By end of December 2015	0 Months 0 % 0 %	12 Months 120 % 120 %	2 Months 20 % 140 %	1 Months 10 % 150 %	150 %
<b>BCS.TtoM-Predictability</b> Decrease from 75 to 5 % By end of December 2016	0 % 0 % 0 %	50 % 71 % 71 %	10 % 14 % 85 %	5 % 7 % 92 %	92 %
<b>BCS.Customer-Satisfaction</b> Increase from 4 to 5 1 to 6 (6 best)	0 1 to 6... 0 % 0 %	1 1 to 6... 100 % 100 %	0.2 1 to 6... 20 % 120 %	0 1 to 6... 0 % 120 %	120 %

<https://app.needsandmeans.com>



# Quantify Values the First Week

## Start Delivering from the Next Week

An Agile Project Startup Week:

See 'Evo Start': a column in AgileRecord.com

as published 7 March 2013

<http://www.gilb.com/dl568>

The Standard:

<http://www.gilb.com/dl562>

This is a detailed standard for conducting 'Evo' (Evolutionary Project Management) as described in my book, *Competitive Engineering*, Chapter 10

[[http://www.gilb.com//tiki-download\\_file.php?fileId=77](http://www.gilb.com//tiki-download_file.php?fileId=77)]

Also talk slides pdf from ACCU Conference April 9 2014 – includes Startup Planning for Business Startups, Conformat, US DoD case, 2 bank cases, Detailed Startup week outlines and links to sources. See <http://www.gilb.com/dl812>

**Gilb's Methodology Column**  
**An Agile Project Startup Week: 'Evo Start'**  
*by Tom & Kai Gilb*

We would like to describe how we start up agile projects, which are completed using our 'Evo' [6] agile method [2, 3].

We have been using exactly this Project start-up method worldwide, in many companies, and for both software/IT projects and other systems engineering projects (like 25 (now) Boeing Aircraft Projects in 1990) for decades, and it works. It gives a flying start to the incremental value delivery process; starting with value delivery, the 2nd week.

This process is appropriate for any consequent agile process, such as our 'Evo', which is focussed on delivering real measurable stakeholder value incrementally, as opposed to the majority of current agile methods which are focussed on delivering code; but, which do not attempt to define or deliver real stakeholder value itself, directly.

One solution to the agile problem of 'code fixation', which one of our multinational bank clients has recently adopted, for the wide variety of agile methods being used in the bank, is to suggest that the 'Evo' process [2] be added on top of their current agile process, for example on Scrum or/and XP. Evo then manages the stakeholder value, and Evo provides value design ideas to the code development team.

Evo will not only output ideas for code (a burn down stack), but will in fact output any (non code) design ideas that will help deliver stakeholder value, such as training programmes, database construction, or motivational tactics. Evo operates at the systems engineering level, as Scrum allows in principle.

The Evo startup week is a sort of feasibility study, in the sense of

- Day 1: Drafting a feasible set of top 10 quantified project value objectives
- Day 2: Drafting a top 10 architecture hypothesis set
- Day 3: Estimating the multiple effects of all architecture on all value objectives, and critical resource constraints (budget, deadline)
- Day 4: Suggesting initial value delivery steps, next week
- Day 5: Getting management approval to proceed with the second week, and to see if we can really deliver value to stakeholders.

The Evo week is intentionally time boxed (one week), no matter what the size of the project. This is done so that:

- We do not get into weeks and months of bureaucratic start up overhead, before we have to deliver real value to stakeholders
- We will focus on the critical top level objectives [5]
- The detailed design will emerge iteratively, as a result of value measurement, and feedback.

Figure 1: Two levels of result management, above a Scrum process. The 'Business level', on top of the stakeholder level is missing from this illustration here.

Page 29 Agile Record - [www.agilerecord.com](http://www.agilerecord.com)

# Startup Process: Days 1 and 2

- **Day 1: Project Objectives:** The top few critical objectives quantified.

- **Objective:** Determine, clarify, agree critical few project objectives – results – end states
- **Process:**
  - Analyze current documentation and slides, for expressed or implied objectives (often implied by designs or lower level objectives)
  - Develop list of **Stakeholders** and their needs and values
  - Brainstorm 'top ten' critical objectives names list. Agree they are top critical few.
  - Detail definition in Planguage – meaning quantify and define clearly, unambiguously and in detail (a page)
  - Quality Control Objectives for Clarity: Major defect measurement. Exit if less than 1.0 majors per page
  - Quality Control Objectives for Relevance: Review against higher level objectives for alignment
  - Define constraints, resources, traditions, policies, corporate IT architecture, hidden assumptions.
  - Define Issues – yet unresolved
- **Note:** we might well choose to say all things in quantified and measurable language. *Stakeholder* data specified.
- **Participants:** anybody who is concerned with the business results, the higher the management level the better.
- **End of Day Process:** meet 30 minutes with any responsible interested managers to present the outputs, and to get preliminary corrections and go-ahead.
- **Note:** this process is so critical and can be time consuming, so if necessary it can spill over to next day. Perhaps in parallel with startup of the strategy identification. *Nothing is more critical or fundamental than doing this well.*

- **Day 2: Project Strategies and Architecture:** the top few critical strategies for reaching the critical objectives

- **Objective:** to identify the top 'ten' most critical strategic decisions or architectures; the ones that will contribute or enable us most, to reach our primary objective goal levels on time.
- **Process:**
  - Analysis of current documentation and slides to identify candidate strategies, implied or expressed.
  - Brainstorming of the 'names' of the specific strategy list, the top ten and a set of less powerful ideas (say 11-30)
  - Detail \_\_\_\_\_ top ten strategy sufficiently to understand impacts (on objectives, time and costs)
  - Specify, for each strategy all critical related information (like stakeholders, risks, assumptions, constraints, dependencies, etc.)
  - Quality Control for clarity: correct unclear items. Exit if less than 1.0 defects per page.
  - Likely that work will need to be done in parallel in order to do ten strategies to a rich level of specification.
- **Output:** final strategy specification, ready for evaluation, and comparison and delivery of partial value results.
- **Participants:** system architects, project architects, strategy planners. And members of the project team who will be in the entire weeks process. The major output is the list of strategies and architectures (the means to reach the objectives)
- **End of Day Process:** meet 30 minutes with any responsible interested managers to present the outputs, and to get preliminary corrections and go-ahead.

Quantify  
Critical Goals

Identify  
Best  
Architecture

# Startup Process: Days 3 and 4

**Day 3: Evaluation of Strategies using Impact Estimation:** our best estimates with experience and risk. How sure are of the major strategy decisions.

- **Objective:** to estimate to primary effects and all side effects of all top critical strategies on all top critical objectives, and on some resources (time, cost, effort). The estimates will be backed up by evidence, or their credibility will be rated low.
- **Process:**
  - Using the objectives and strategies developed on first 2 days as inputs
  - Populate an Impact Estimation table (aka Value Decision Table) with estimates of the expected result of deploying defined strategies. Estimate main intended impacts
  - And all side effects (on other core objectives)
  - And on all resources (time, money, Effort)
  - Estimate  $\pm$  ranges
  - Identify evidence and sources for estimates
  - Determine credibility level
  - Quantity of the IE table (a lot) and Rules for the table (a lot) as it's a lot of work
  - Lots of parallel work needed and expected to do a good job.
- **Output:**
  - A fairly decent Impact Estimation table, possibly a several level set of them.
- **Participants:** architects, planners, anybody with strong views on any of the strategies. The team for the week.
- **Note:** it might be necessary and desirable, now or later, to do this impact estimation process at several levels (Business, Strategy, IT systems) to improve the Business-IT relationship. Early on, it might be time limits and be done parallel or later.
- **End of Day Process:** meet 30 minutes with any responsible interested managers to present the outputs, and to get preliminary corrections and go-ahead.

**Day 4: Evolutionary Step Decomposition:** what are the high value short term value delivery steps we can execute.

- **Objective:** to identify near team candidates for real value delivery to real stakeholders. What can we do for real next week!
- **Process:**
  - Identify highest value (to costs) strategies and sub-sets of strategies
  - Decompose into doable subsets in weekly to monthly cycles or result level
  - Plan the near steps (1 or more) in detail so that we are ready to execute the step in practice.
    - Who does it, main responsible, team.
    - Expected measurable results and costs
    - Stakeholder involved in receiving
    - Deliverable (if any)
- **Output:** 1 or more potential steps for value delivery to some stakeholders, a plan good enough to approve and execute in practice.
- **Participants:** Project Management, architects, planners, anybody who use architecture in practice. The week's team for this start up study.
- **End of Day Process:** meet 30 minutes with any responsible interested managers to present the outputs, and to get preliminary corrections and go-ahead.

Identify Quick Wins

Deliver Value

Next Week

# Day 5: Boss Says 'Go' (Next Week Only)

- Boss approves doing the next week
  - This is normally used to present the plan to management and get approval to go forward the next week.
  - In our case we have chosen a 4 days model due to Easter Holidays. So we have to find another way to present and approve.
  - **Objective:** To present the entire set of plans to responsible executives and discuss them, with approval if possible, or approve with changes.
  - **Process:**
    - Present all planned outputs
    - Discuss them and answer questions
    - Take corrections
    - Get approval for the next implementation step.
  - **Output:** Approval for next implementation step, corrections
  - **Participants:** project team + key manager above the project manager.
  - **End of Day Process:** none, unless corrections needed before execute OK.
    - Possible Corrections not ready to execute a delivery step next week
  -

Day 3 of Project  
Startup

How do the  
strategies/  
architecture  
deliver value  
for your  
quantified value  
requirements?

Strategies Goals	Identify Binding Compliance Requirements Strategy	System Control Strategy	System Imple- mentation Strategy	Find Services That Meet Our Goals Strategy	Use The Lowest Cost Provider Strategy
Security Administration Compliance 25 % → 90 %	100 %	100 %	100 %	50 %	0 %
Security Administration Performance 24 hrs → 4 hrs	75 %	100 %	100 %	100 %	0 %
Security Administration Availability 10 hrs → 24 hrs	0 %	0 %	0 %	100 %	0 %
Security Admin- istration Cost 100 % → 60 %	50 %	100 %	100 %	100 %	100 %
Total Percentage Impact	225 %	300 %	300 %	350 %	100 %
Evidence	ISAG Gap Analysis Oct. 03	John Collins	John Collins	John Collins	John Collins
Cost to Imple- ment Strategy	15 man days (US\$ 5,550)	15 man days (US\$ 5,550)	15 man days (US\$ 5,550)	15 man days (US\$ 5,550)	1man day (US\$ 1,110)
Credibility	0.9	0.6	0.6	0.75	0.9
Cost Adjusted Percentage Impact	202.5 %	180 %	180 %	262.5 %	90 %

Figure 4. Acer Project: Impact Estimation Table. Citigroup, London

# Selling 'Value' to Your IT Boss

## Value Planning

- Links directly to managements values and plans
- Is visible and measurable evidence of IT value to the organization
- Is methods for very early increments value delivery (weeks, not years)
- Is intelligible to 'your boss' (is not IT technology, it is results that make everybody look like good managers)
- Can be used to manage outsourcing contracts:  
    'No value, No pay'

# Just doing 'Value': Nobody can stop you

- Whenever you encounter value-talk at meetings and in documents
  - Quantify it
- Whenever you are selling or being sold 'technology'
  - Quantify (estimate, measure, contract for) specific value delivery
    - NOT: *"it is cutting edge technology"* (Management BS)
      - <http://www.gilb.com/dl465>
    - BUT *"It will deliver 50-70% of the Productivity Goal by next year, contractually guaranteed."*
- Measure the BS Level, and don't accept it
  - Is 100 fudge words per page in requirements OK to hand on to the rest of the organization?

# 12 Tough Questions



# Involving Management

## The 12 Tough questions

- Are a way to get managers interested in metrics
- They lead directly
  - to quantified management objectives
  - to quantified project and product requirements
  - to the use of impact estimation tables evaluating alternative solutions quantitatively

# 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?**

# Longer explanation of these simple, but powerful value questions

12 tough questions paper

[http://www.gilb.com/tiki-download\\_file.php?fileId=24](http://www.gilb.com/tiki-download_file.php?fileId=24)

Page 1

"Twelve Tough Questions" by Tom Gilb

## **Twelve Tough Questions.**

**VERSION 0.23 July 18, 2002, May 4 2006**

-This is summarized on a page of the 'Competitive Engineering' book manuscript.

By © Tom Gilb 1991-99 Iver Holtersvei 2, N-1410 Kolbotn, Norway. Tel. (+47 66) 801697 Tom@Gilb.com, URL www.gilb.com



### **Introduction.**

Managers don't ask tough enough questions about written material. I know because I have spent decades watching them fail to ask the questions which would have exposed the proposals as dangerous or not well thought out.

I have to conclude that managers need training to ask these questions. But I forgive any reader who thinks that asking such questions is just good common sense. It is.

The questions all refer to a larger method I teach; "Competitive Engineering" and books published and in manuscript ("Principles of Software Engineering Management"). But these books exceed 400 pages, the courses take several days. The patience of top managers for such detail is necessarily limited in a high pressure world. So this paper is offered as a simplification and an appetizer. If you want more substance and detail, it exists. If this alone is useful, be happy!

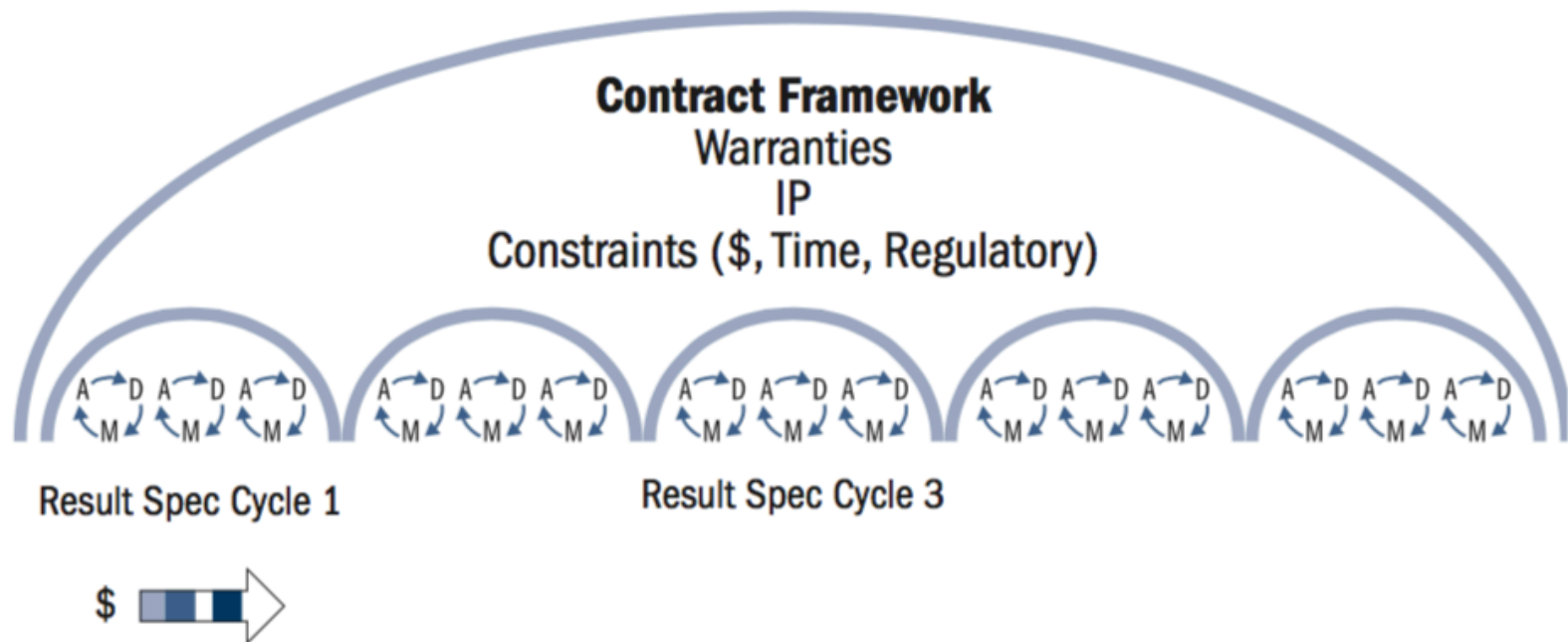
### **Basic Philosophy: here is what the Kindergarten of Consultancy has taught me:**

Numbers make benefits, advantages and quality easier to understand. Numbers provide a basis for tracking and control. Numbers can be useful in an uncertain and changing world. All quality concepts can be treated measurably. People are sloppy in analyzing and presenting ideas unless you insist on something better. The objectives are constantly changing in the real world. Strategies have a large number of impacts on all your critical objectives and constraints. Combinations of strategies are almost impossible to predict the impact of until you have implemented them in your new system. Most people have no real knowledge of the effects of the strategies they propose, until you force them to admit it and find the facts. We can

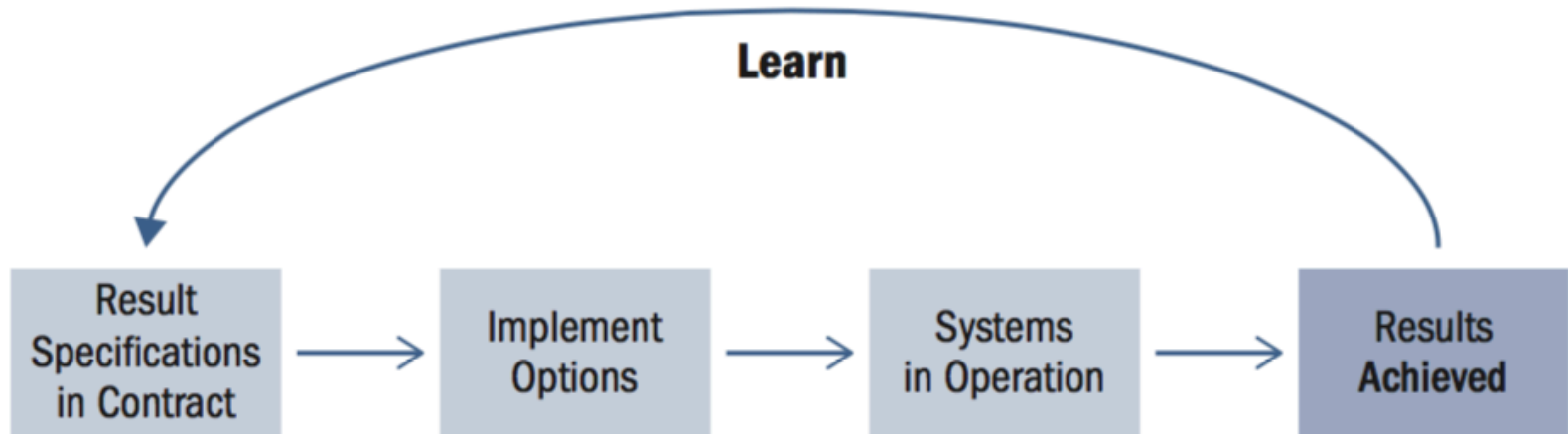
www.gilb.com

# CONTRACTING FOR VALUE

# Contract Framework



# Result Contract Structure



# Old way and new way

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.

# WHAT IS A FLEXIBLE CONTRACT?

## WHAT IS A FLEXIBLE CONTRACT?

A 'flexible contract' is a contract that delivers value. It achieves this in a number of ways:

The contract focuses on business results (such as features). By focusing on business results, it helps to align their interests.

The supplier is given the freedom to deliver the terms of the contract in the way that they see fit.

The fees (or at least part of them) are incentivized to achieve business results.

The contract is structured in a way that allows for Conditions, under which the supplier can deliver as a Statement of Work or a Statement of Results. The parties can respond to changes in a flexible way.

In respect of each SOTO, the parties can learn rapidly what works and what doesn't.

The contract adopts light touch governance. SOTO at a time, so the contract is easier to understand and manage. It is deliberately NOT focused on technical details.

Define what you want, as you go, in small increments.

Learn what works

Focus on business results, not 'code'

Pay for real value delivered

Prioritize high value results early.

Very low risk

Not tied in to suppliers who cannot deliver

(such  
which

the

nd  
way  
The

rties

one

s  
d.



# Extract from Contract Template: SOTO Specification Short-term Statements Of Target Outcomes

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

# Extract from Contract Template

## Target Outcomes

[COMPLETE THE FOLLOWING TABLE FOR EACH TARGET OUTCOME]

Name of Target Outcome:	In the form Action Verb + Noun Phrase
Outcome Value:	Time or money over a defined period
Outcome Measure: <ul style="list-style-type: none"><li>• Unit of measure:</li><li>• party responsible for conducting measurement:</li><li>• Method for measurement:</li><li>• Frequency of measurement:</li><li>• Baseline (starting point):</li></ul>	<p>i.e. the metric used to measure e.g. time, percentage or number</p> <p>i.e. a named person or group responsible for conducting the measurement e.g. the Customer</p> <p>i.e. the systems used to collect data or the tests that will be run e.g. data analytics report or usability tests for target users</p> <p>i.e. The period of time when measurements will be taken e.g. every <b>[2 weeks]</b> with their end-users</p> <p>i.e. the baseline that will be used as the starting point against which to compare results</p>

# Credits for many of the contract slides to:



Forthcoming Book

- [www.flexiblecontracts.com](http://www.flexiblecontracts.com)
- <https://www.linkedin.com/groups/Flexible-Agile-contracts-7460556/about>
- [www.mobiusmodel.org](http://www.mobiusmodel.org)
- I have been working together with Susan Atkinson and Gabrielle Benefield for several years regarding these ideas.
- So it is no surprise that they are very complimentary to the Evo and Planguage methods in my writings, such as
- Competitive Engineering (2005), and Value Planning (2014, manus)

# Free BCS Value and Quality Courses

## FALL COURSES

LINKS SHORTLY AT

<http://www.bcs.org/category/10136>

9-10 September

**Value Requirements: with emphasis on Quality**

same syllabus as earlier

11. September

**Architecture Engineering: with emphasis on Quality**

Same syllabus as before but a 1 day version, with the Value Requirements course as a recommended pre-requisite

23 September

**Lean Quality Assurance**

as held before

25 September 2015

**IT Decision-Making for Managers and Senior Consultants: using Value Planning Methods.**

**IT Decision-Making for Managers and Senior Consultants: using Value Planning Methods.**

## Summary

IT planning is constant decision-making and prioritisation. This course will outline and document a series of integrated tools to help make better decisions with regard to value, quality, costs and risks.

This will not train in depth but will quickly make you aware of some available methods, give you access to documentation, and welcome you to more depth training on other BCS courses.

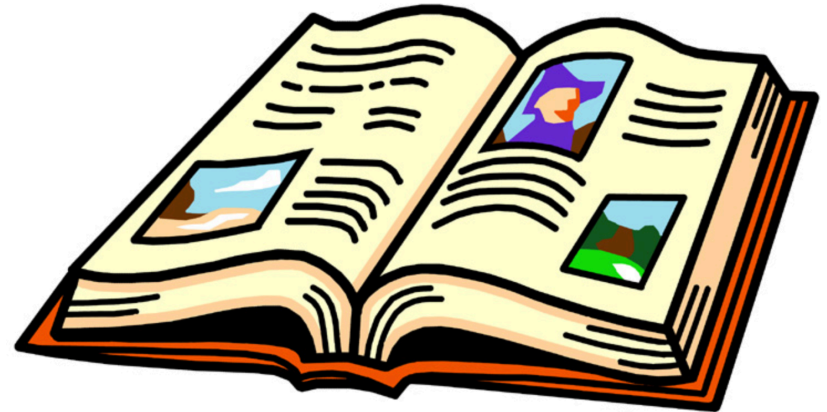
## Content. (About one classroom hour each)

1. Quantification of critical values and qualities in requirements and objectives
2. Specification of background information to help understand risks and priorities
3. Impact Estimation Tables: a tool for comparing complex options, architectures and strategies.
4. Dynamic Decision Making: learning fast, committing late
5. Delegation of Decision Making: to where the action and competence is placed.
6. Agile Contracting: decisions and commitments in smaller increments
7. Evo: a project planning framework for decision making

NEW COURSE

# Free Book Manuscript

- [Tinyurl.com/ValuePlanning](http://Tinyurl.com/ValuePlanning)  
(a live dropbox)
- Manuscript 104  
subchapters
- Drafted Summer/Fall 2014
- Major edit of chapters 5-10  
to happen July/August 2015
- Feedback appreciated
- Aimed at  
'Management' (not IT or  
Engineers)



# END

- THESE SLIDES ARE AT  
<http://www.gilb.com/dl845>
- FEEL FREE TO SHARE AND TWITTER
- @ImTomGilb
- OR LINKEDIN TO SPREAD THE WORD







# References

*[www.flexiblecontracts.com](http://www.flexiblecontracts.com)*

- [1] Highly recommended in-depth analysis of good and bad agile practices, even if you are NOT in the public sector: Wernham, Brian. *Agile Project Management for Govern- ment*. Maitland and Strong.
- [2] Gilb, Tom. “The Top 10 Critical Requirements are the Most Agile Way to Run Agile Projects”. *Agile Record*, Au- gust 2012, 11: pp. 17–21. <http://www.gilb.com/dl554>
- [3] Gilb, Tom. “No Cure No Pay.”
- [http://www.gilb.com/tiki-download\\_file.php?fileId=38](http://www.gilb.com/tiki-download_file.php?fileId=38)
- [4] Gilb, Tom. “Chapter 5: Scales of Measure.” *Competitive Engineering*.
- [http://www.gilb.com/tiki-download\\_file.php?fileId=26](http://www.gilb.com/tiki-download_file.php?fileId=26)
- [5] This initiative is a draft idea and would welcome coopera- tion and feedback from people who would like to try it out in practice!  
*www.flexiblecontracts.com*
- [6] Gilb, Tom. “Real Architecture Engineering.” Lecture slides from ACCU Bristol, April 2013.  
<http://www.gilb.com/dl574>