Konuşmamıza hoşgeldiniz
(Welcome to my talk! :) )

# "VALUE PLANNING"
# IN A LEAN AND AGILE WAY
# FOR MANAGERS

Tom Gilb. @ImTomGilb, tom@Gilb.com
#AgileDaysIstanbul

KEYNOTE FOR
AGILE DAYS, ISTANBUL ,
http://agiledaysistanbul.org
12 APRIL 2018
AND MAY 7 KATOWICE

09:00 to 10:00 (60 minutes)

Hall 1 - Radisson Blu Hotel

**SLIDES ARE AT**

https://www.dropbox.com/s/vh9ymju3l9evm/Value%20Planning%20TALK%20FOR%20AGILEDAYS%20TURKEY%20BACKUP%20NOT%20DONE%2030%20MAR%202018.pptx?dl=0

© Gilb.com

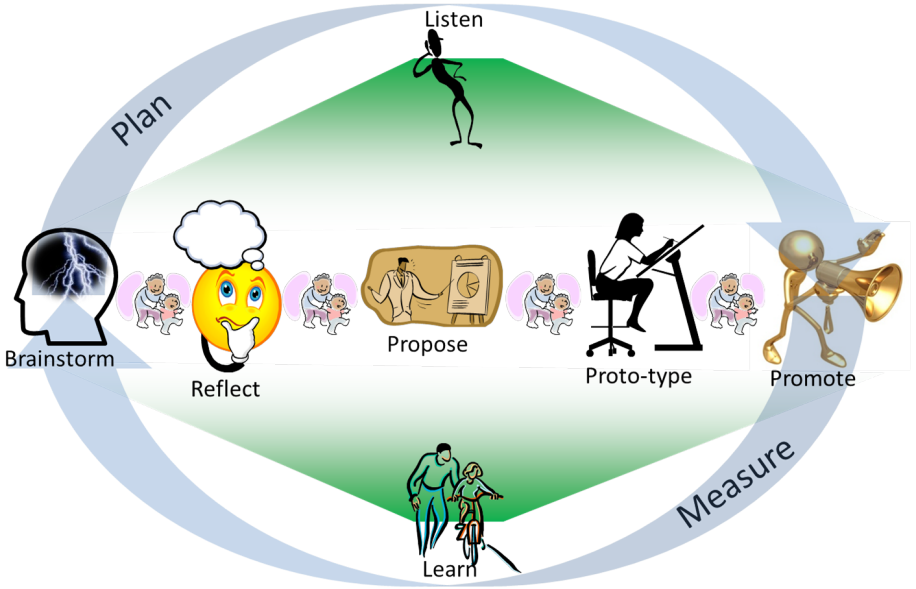

My Advanced Lean Agile Book

# My Definition of 'Agile'

- "***Any*** set of tactics
  - that **enable delivery of**
  - **a *prioritised* stream of *useful* results,**
  - ***in spite of a changing environment***"
    – TsG 7 June 2013/2018 March

- A main focus on 'Agile', is the *wrong* level of focus.
  – Using agile tactics that 'deliver results', is a *good* idea.
- Focus on *results*, no matter what.
- Retitle your conference "Results"
- So we need: **"Value for Money"**
  - by ANY means that work

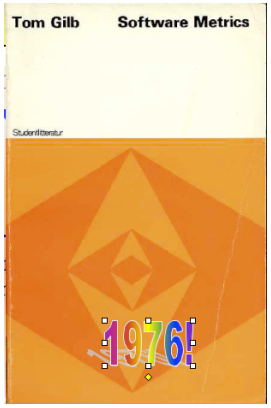
## The Generic Agile Concept

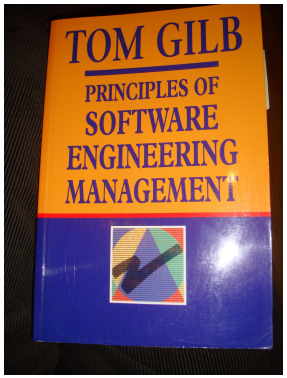




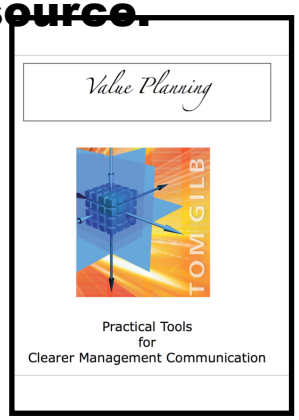
© Gilb.com

# Agile Grandpa




- **The Agile 'Grandfather'**
  - **Practicing 'Agile' IT Projects since 1960**
  - **Preaching Agile since 1970's (Computer Weekly, UK)**
  - **'Acknowledged Pioneer', by Agile Gurus, and Research**
    - **Beck, Sutherland, Highsmith, Cohn, Larman etc.**
    - **Ask me for details on this! I am too shy to show it here!**
- **Agile Practice (we called it 'Evo Results Delivery')**
  - **in IT: for decades**
  - **for Agility in Organisations: for Decades (Citigroup, Intel, HP, Boeing)**
- **Books: Presenting Agile: Incremental Value Delivery**
  - **'Principles of Software Engineering Management' (1988)**
    - **the book Kent Beck and others refer to as Agile source.**
  - **'Competitive Engineering' (2005): method definition**
  - **'Evo': (Kai, evolving, 55 iterations)**
  - **1976 Software Metrics book**
  - **'Value Planning' manuscript 2014-8**
    - **for 'managers'**


# OK I am not *that* shy!
# (but read this later if you are interested)

**Agile References:**

**"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**

**"I've always considered Tom to have been the original agilist. In 1989, he wrote about short iterations (each should be no more than 2% of the total project schedule). This was long before the rest of us had it figured out." Mike Cohn http://blog.mountaingoatsoftware.com/?p=77**

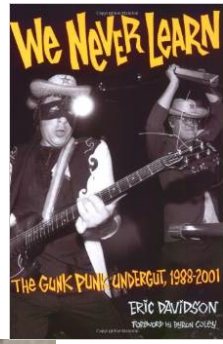
**Comment of Kent Beck on Tom Gilb's book , "Principles of Software Engineering Management": " A strong case for evolutionary delivery – small releases, constant refactoring, intense dialog with the customer". (Beck, page 173).**

**In a mail to Tom, Kent wrote: "I'm glad you and I have some alignment of ideas. I stole enough of yours that I'd be disappointed if we didn't :-), Kent" (2003)**

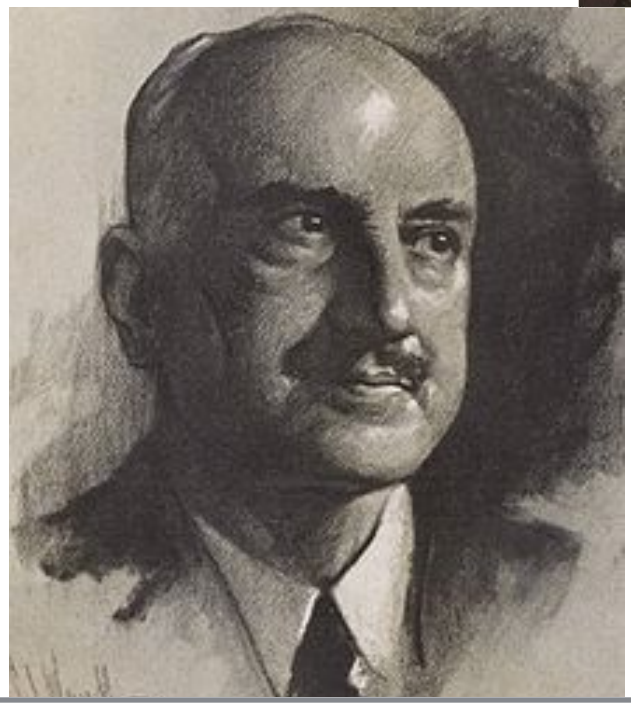
**Jim Highsmith (an Agile Manifesto signatory) commented: "Two individuals in particular pioneered the evolution of iterative development approached in the 1980's – Barry Boehm with his Spiral Model and Tom Gilb with his Evo model. I drew on Boehm's and Gilb's ideas for early inspiration in developing Adaptive Software Development. .... Gilb has long advocated this more explicit (quantitative) valuation in order to capture the early value and increase ROI" (Cutter It Journal: The Journal of Information Technology Management, July 2004page 4, July 2004).**


© Gilb.com

# Will we never learn ?




- **“Those who cannot remember the past are condemned to repeat it.”**
- ***The Life of Reason (1905-1906)***
  - Vol. I, ***Reason in Common Sense***



**Jorge Agustín Nicolás Ruiz de Santayana y Borrás,**
known as **George Santayana**
(December 16, 1863 – September 26, 1952),
was a philosopher, essayist, poet, and novelist.

# Grandpa Guru Tom Speaks

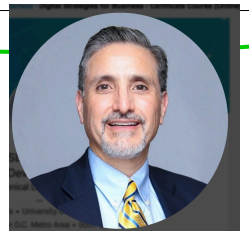
- **I am your historian.**
- **I joined IBM in 1958**
- **And lived intensively through the entire computer age**
- **I'll tell you what I have learned, before I go.**
- **But this might be your last chance. OK, but I am 77.**
- **You, and your teachers, have missed all other such opportunities up to now ….**
- **Are YOU doomed to repeat the errors of the software past?**

Copyright Tom@Gilb.com

# How do Lean & Agile Intersect?

| Agile Values | Lean Pillars | Lean Principles | Lean & Agile Practices | Flow Principles |
|---|---|---|---|---|
| Empowered Teams | Respect for People | Relationships | • Customer relationships, satisfaction, trust, and loyalty<br>• Team authority, empowerment, and resources<br>• Team identification, cohesion, and communication | Decentralization |
| Customer Collaboration | | Customer Value | • Product vision, mission, needs, and capabilities<br>• Product scope, constraints, and business value<br>• Product objectives, specifications, and performance | Economic View |
| | | Value Stream | • As is policies, processes, procedures, and instructions<br>• To be business processes, flowcharts, and swim lanes<br>• Initial workflow analysis, metrication, and optimization | WIP Constraints & Kanban |
| Iterative Delivery | Continuous Improvement | Continuous Flow | • Batch size, work in process, and artifact size constraints<br>• Cadence, queue size, buffers, slack, and bottlenecks<br>• Workflow, test, integration, and deployment automation | Control Cadence & Small Batches |
| Responding to Change | | Customer Pull | • Roadmaps, releases, iterations, and product priorities<br>• Epics, themes, feature sets, features, and user stories<br>• Product demonstrations, feedback, and new backlogs | Fast Feedback |
| | | Perfection | • Refactor, test driven design, and continuous integration<br>• Standups, retrospectives, and process improvements<br>• Organization, project, and process adaptability/flexibility | Manage Queues/ Exploit Variability |

❶ ❷ ❸

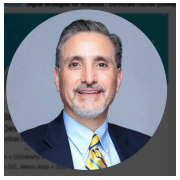


Source: David Rico

© Gilb.com

# Agile Recap

| What | How | Result |
|---|---|---|
| Flexibility | Use lightweight, yet disciplined processes and artifacts | Low work-in-process |
| Customer | Involve customers early and often throughout development | Early feedback |
| Prioritize | Identify highest-priority, value-adding business needs | Focus resources |
| Descope | De-scope complex programs by an order of magnitude | Simplify problem |
| Decompose | Divide the remaining scope into smaller batches | Manageable pieces |
| Iterate | Implement pieces one at a time over long periods of time | Diffuse risk |
| Leanness | Architect and design the system one iteration at a time | JIT waste-free design |
| Swarm | Implement each component in small cross-functional teams | Knowledge transfer |
| Collaborate | Use frequent informal communications as often as possible | Efficient data transfer |
| Test Early | Incrementally test each component as it is developed | Early verification |
| Test Often | Perform system-level regression testing every few minutes | Early validation |
| Adapt | Frequently identify optimal process and product solutions | Improve performance |





© Gilb.com

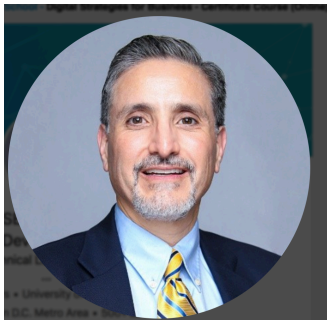
Source: David Rico

# 14 PITFALLS OF AGILE METHODS

- **Change** – Use of top-down, big-bang organization change, adoption, and institutionalization.
- **Culture** – Agile concepts, practices, and terminology collide with well-entrenched traditional methods.
- **Acquisition** – Using traditional, fixed-price contracting for large agile delivery contracts and projects.
- **Misuse** – Scaling up to extremely complex large-scale projects instead of reducing scope and size.
- **Organization** – Unwillingness to integrate and dissolve testing/QA functional silos and departments.
- **Training** – Inadequate, insufficient, or non-existent agile training (and availability of agile coaches).
- **Infrastructure** – Inadequate management and development tools, technologies, and environment.
- **Interfacing** – Integration with portfolio, architecture, test, quality, security, and usability functions.
- **Planning** – Inconsistency, ambiguity, and non-standardization of release and iteration planning.
- **Trust** – Micromanagement, territorialism, and conflict between project managers and developers.
- **Teamwork** – Inadequate conflict management policies, guidelines, processes, and practices.
- **Implementation** – Inadequate testing to meet iteration time-box constraints vs. quality objectives.
- **Quality** - Inconsistent use of agile testing, usability, security, and other cost-effective quality practices.
- **Experience** - Inadequate skills and experience (or not using subject matter experts and coaches).

- (***Note**. Firms may prematurely "revert" to inexorably slower and more expensive traditional methods or "leap" onto lean methods that may not adequately address common pitfalls of adopting agile methods*.)

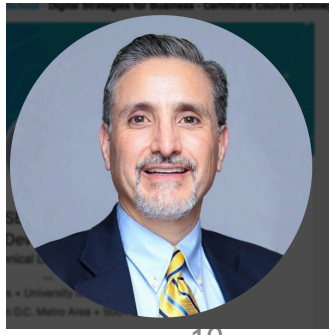
- Source: David Rico http://davidfrico.com/agile-pros-cons.pdf 2012



© Gilb.com

# 14 PROMISES OF AGILE METHODS

- **Value** – Delivers highest-priority customer capabilities, features, requirements, and needs.
- **Risk** – Reduces project scope, requirements, size, complexity, and risk.
- **Discipline** – Fast, flexible, and cost-effective, yet highly disciplined planning and delivery method.
- **Efficient** – Small strategy, portfolio, planning, process, work in process, batch, queue, and team size.
- **Feedback** – Uses planned and unplanned daily, bi-weekly, and release feedback cycles.
- **WIP Constraints** – Uses portfolio, capability, feature, user story, and iteration size constraints.
- **Teamwork** – Small, high-performing, fast, and cost-efficient cross-functional, multi-disciplinary teams.
- **Requirements** – Uses collaboration and rapid feedback to elicit hidden, inexpressible user needs.
- **Architecture** – Uses lean, just-enough, just-in-time, and high-performing architectures and designs.
- **Design** – High-performing, loosely-coupled functional slices validated and delivered one-at-a-time.
- **Flexibility** – Fast, inexpensive, and abstractive workflow, development, and delivery technologies.
- **Quality** – Automated verification, validation, configuration mgt., documentation, and deployment.
- **Complete** – Combines of state-of-the-art business, lean, and technical principles and practices.
- **Improvement** – Built-in daily, bi-weekly, and release process improvement cycles.



- Source: David Rico http://davidfrico.com/agile-pros-cons.pdf 2012

© Gilb.com

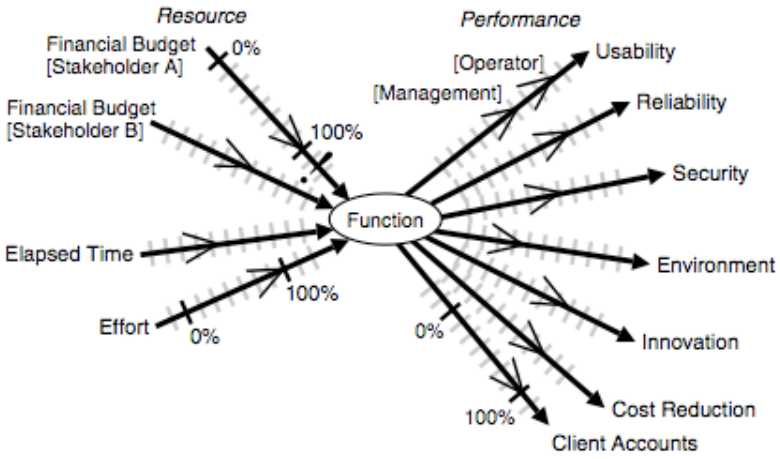
# Gilb Agile/Lean Methods: 'Planguage/Evo/SQC'

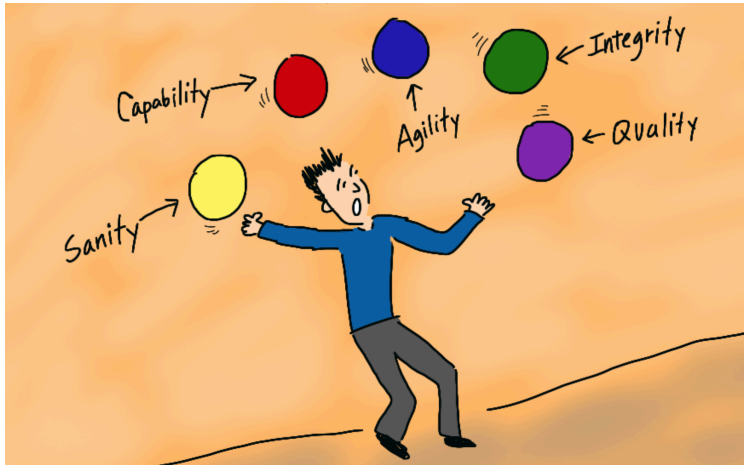
**THESE ARE SUBJECTS OF THE REST OF THIS LECTURE**

- **The concept of quantified multiple stakeholder values.**
- **The requirements specification process: Stakeholders, needs, values, prioritization, experience feedback.**
- **The value driven IT architecture process using the Value Decision Matrix.**
- **The Agile Evolutionary Project Management process.**
- **The One Week Project Startup Process to launch real value delivery.**
- **The Flexible Contracts subcontracting for Value Process**
- **The Agile Specification Quality Control process for agile measuring requirements, architecture and contracts practical quality.**
- **The Ten Principles of Lean and Agile IT System Management.**

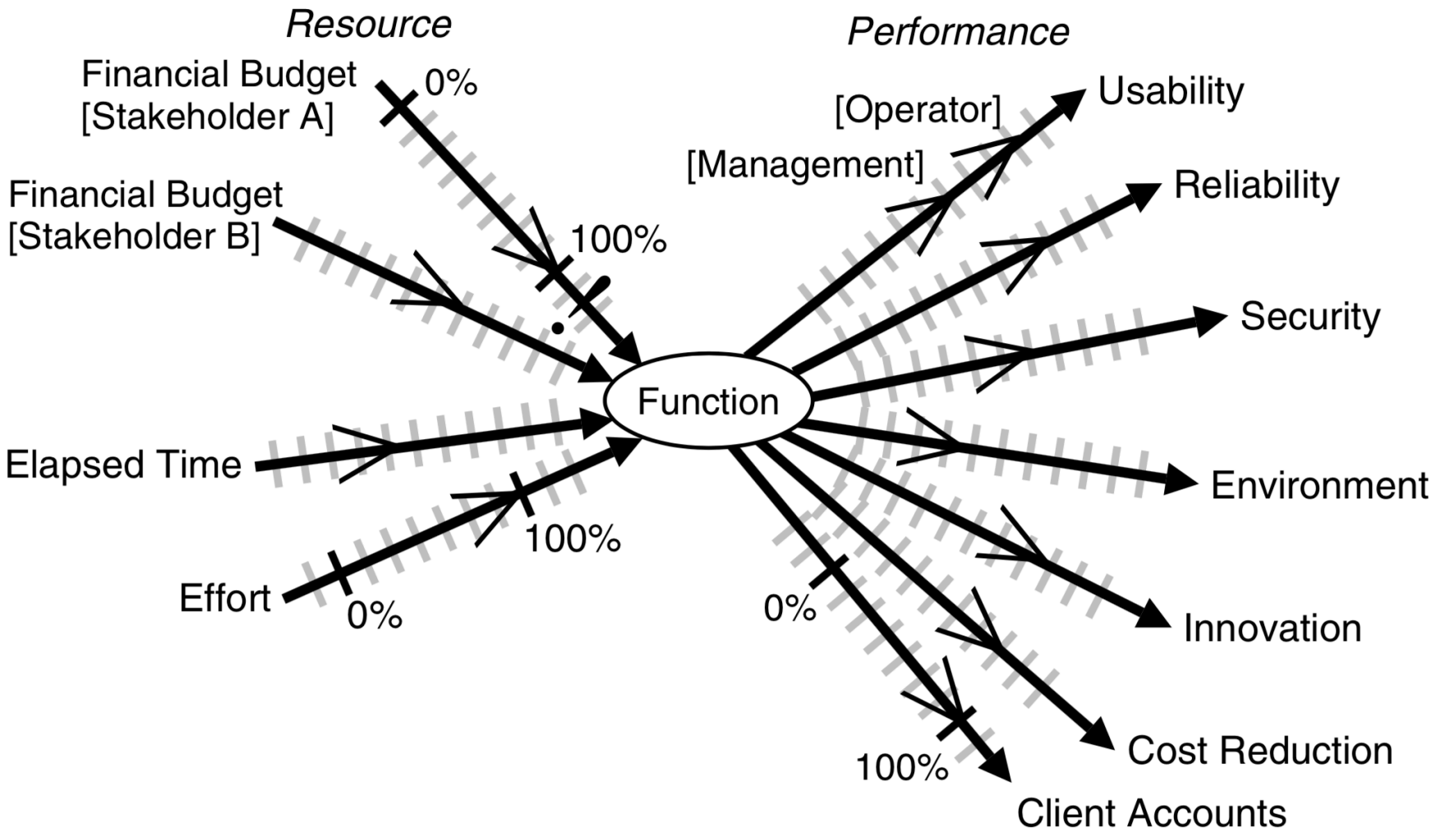

- **The concept of quantified multiple stakeholder values.**

- we need to manage **several** ('top 10 critical') *value objectives*
- at the *same time*
- and **several resources** at the *same time*
- it is a difficult juggling act!



# Many variable Critical Values to be managed at once



# Top 10 Large Bank Project 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**

**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**

**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 %**

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

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

**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

**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

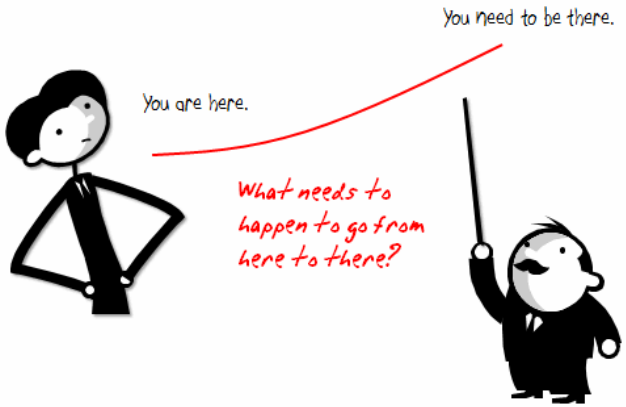
**Risk. user-configurable Scale:** ??? pretty binary – feature is there or not – how do we represent?
**Past** [April 20xx] **1% Goal** [Dec. 20xy] **0%**

**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 = E1 – 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 %**


© Tom@Gilb.com

# 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.


© Tom@Gilb.com

# How can we improve such bad specification? ('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

C**alendar 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** ±4 ?? <- Rob

**Goal**[ 2011, { Bxx, Lxx, Gxx }, If QA Approved Processes used, Developer = Architect, Task = Draft Architecture ] **1.5 days** ± 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).

# **Why** *is* this ‘Lean Agile for Managers’ ? (‘Top Level Critical Values Quantified’)

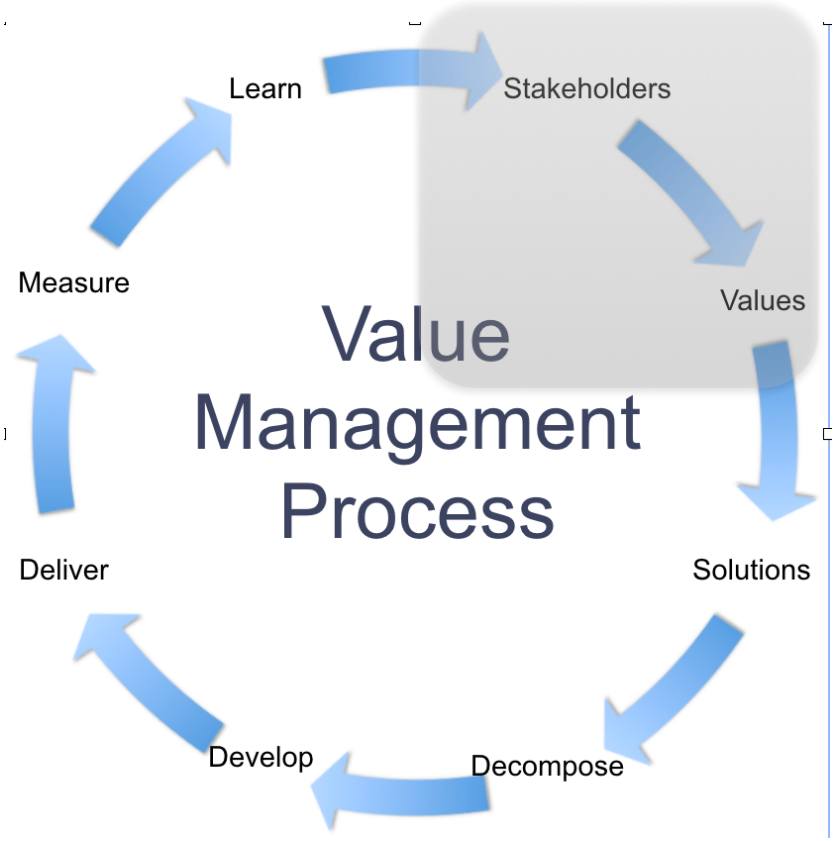
- Managing the value delivery is the primary management job
- **Lean**: this most-critical decision is most upstream.
  - You don’t get quality by testing it in
  - You get qualities by designing them in.
- **Agile**: quantified quality improvements can be chopped up, prioritized, and delivered as an increasing flow

**The 'requirements specification' process:**

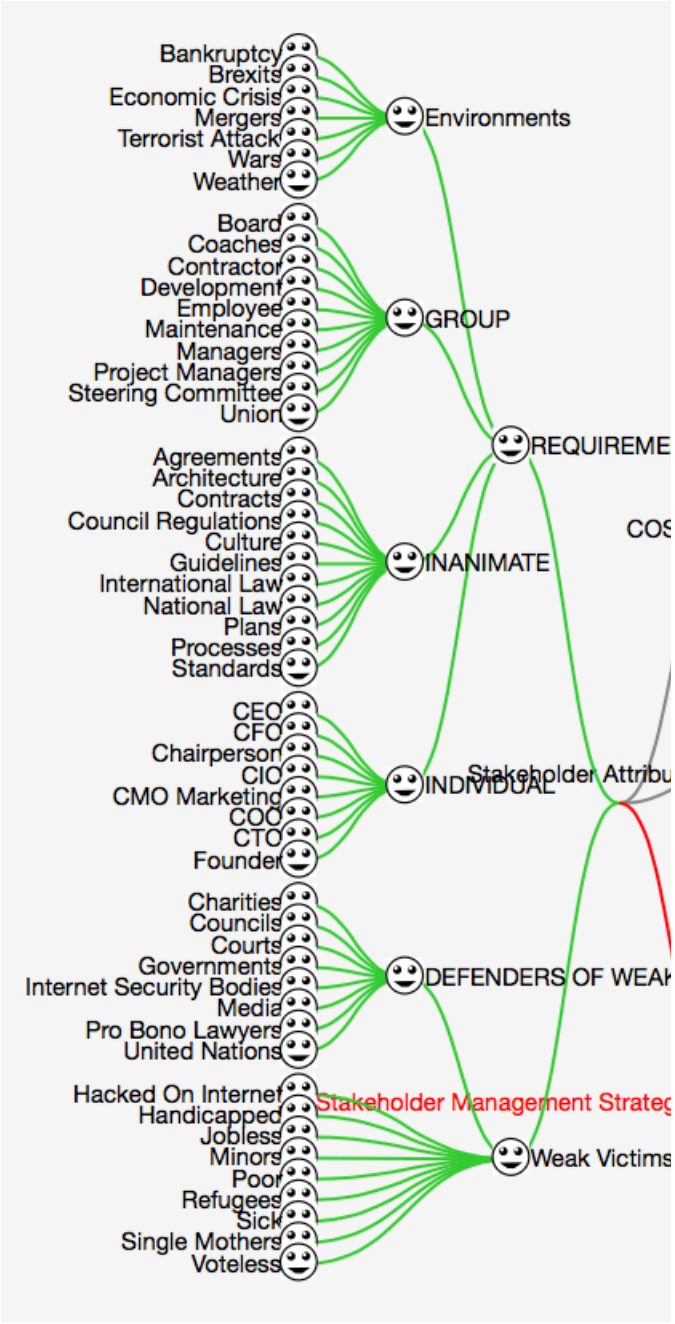
**<u>Stakeholders: 'Requirements Sources'</u>**

**their needs, values, prioritization, experience feedback.**

- we need to consider all critical stakeholders
- all 50 to 500 types
- not merely the narrow 'Agile Manifesto'
  - users and customers
- we need to consider their critical values
- and to choose which ones we can and should try to satisfy - or not



# Here is an example of some stakeholder types

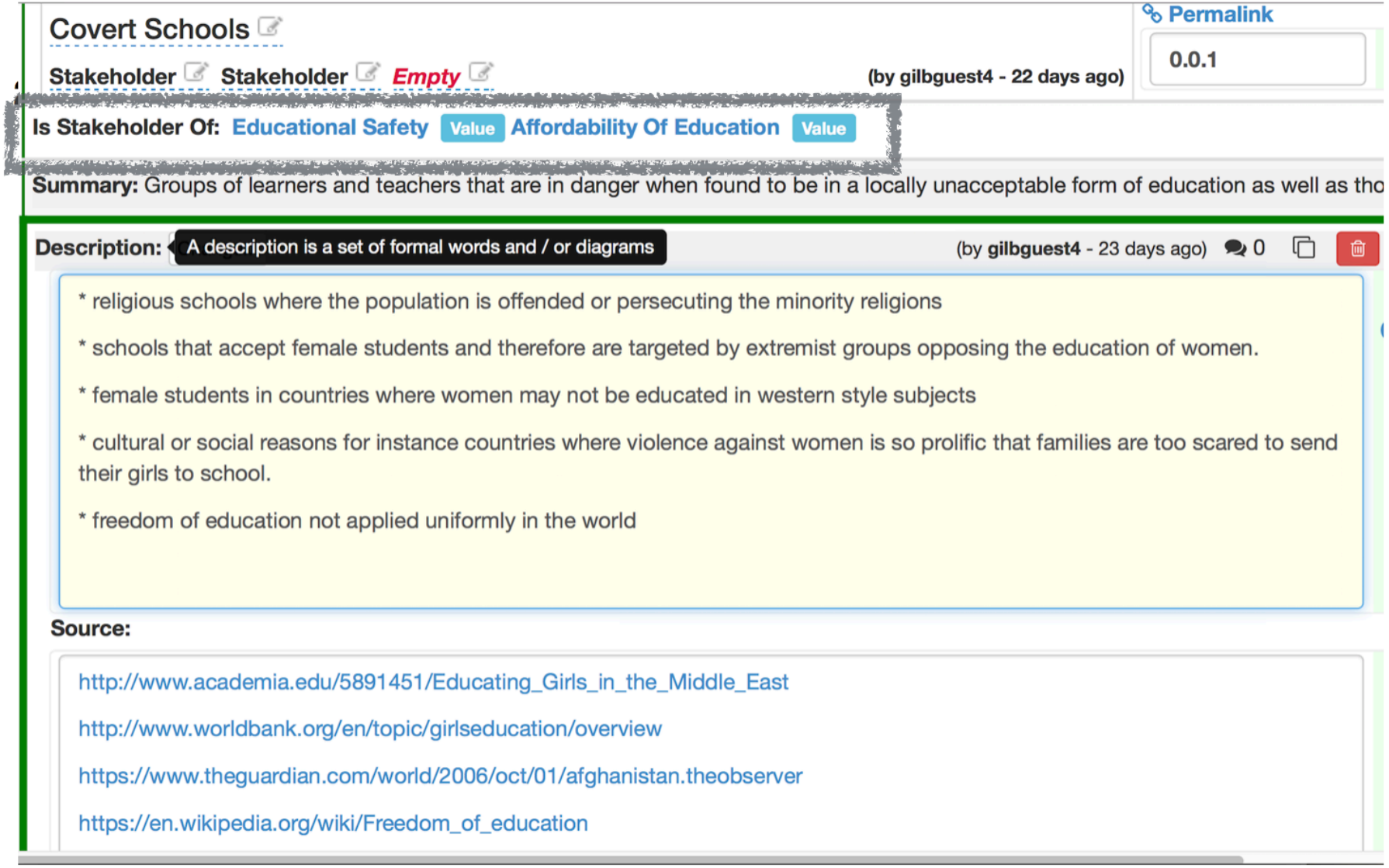


# Stakeholder Values

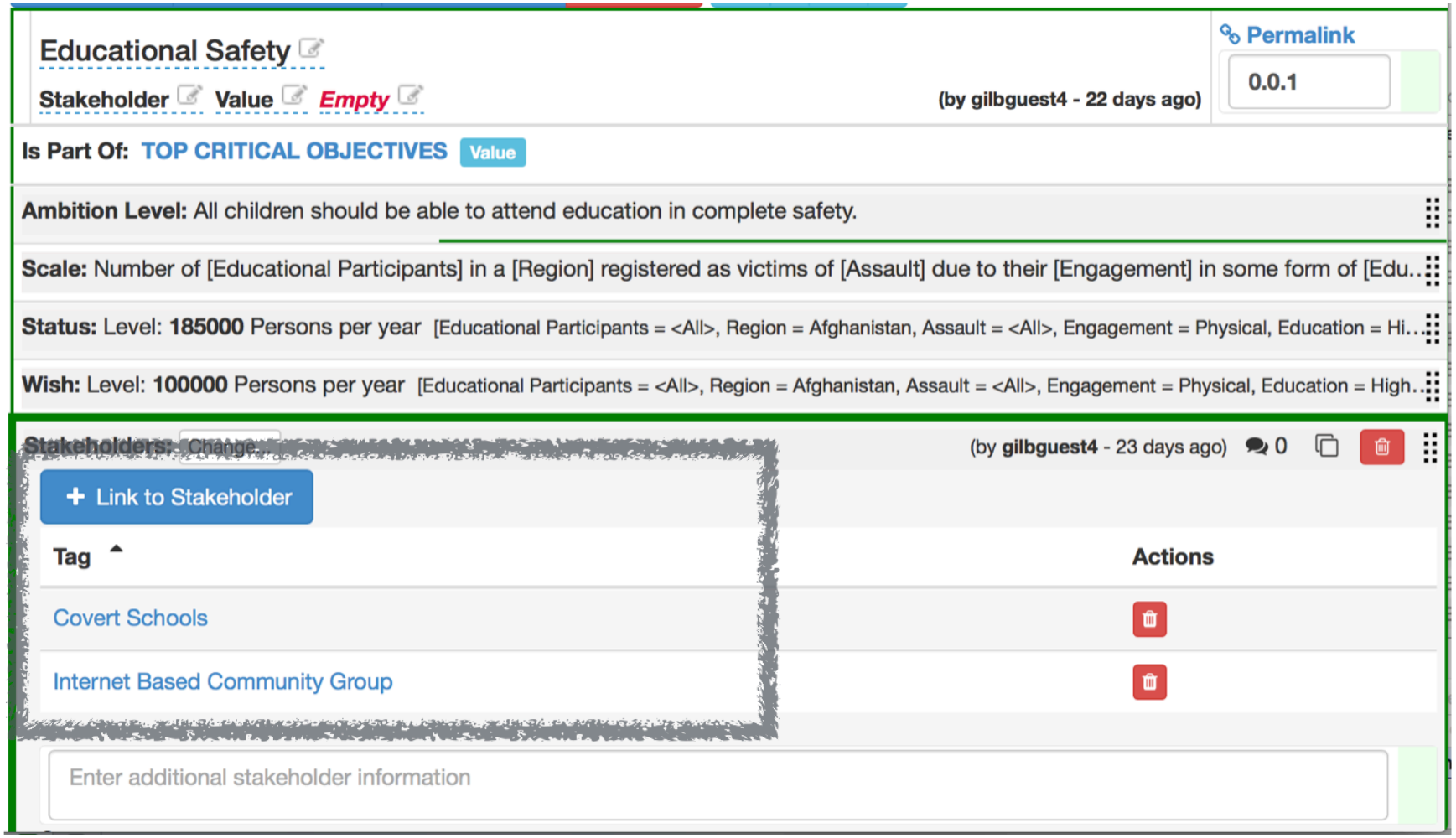
## What we found: Customer Segments

Customer archetype: Inpatient EHR user – Specialist

Interventional Radiologist
Male, 40-65 years old
Attending physician, specialist
Not the buyer, but the champion
**Motivations**: Less time using EHR and more with patient; Easy clinical documentation; High risk patient care; See more patients; Optimize revenue.
**Influenced by**: Department chair, Peers, Scientific knowledge (journals, web)




**EXAMPLE 13: HERE IS A STAKEHOLDER, DEFINED USING THE NEEDSANDMEANS APP [4], BY A STUDENT TEAM. OSWA MEETUP WORKSHOP, OSLO 2017. WE KEEP TRACK OF EXACTLY WHICH VALUE OBJECTIVES THEY ARE RELATED TO. IN THIS CASE 'EDUCATIONAL SAFETY' AND AFFORDABILITY OF EDUCATION'.**




**EXAMPLE 14: HERE IS THE 'EDUCATIONAL SAFETY' OBJECTIVE, WHICH 'COVERT SCHOOLS WAS A STAKEHOLDER OF, IN THE EXAMPLE 13 ABOVE. IT IS LINKED TO 2 STAKEHOLDERS, 'COVERT SCHOOLS', AND 'INTERNET BASED COMMUNITY GROUP'. WE CAN KEEP TRACK OF ANY USEFUL NUMBER OF STAKEHOLDERS FOR A SINGLE OBJECTIVE, AND ANY NUMBER OF USEFUL OBJECTIVES FOR A SINGLE STAKEHOLDER. THIS KIND OF BACKGROUND INFORMATION IS AVAILABLE 'AT A CLICK' IN THE NEEDSANDMEANS [4] TOOL'S IMPACT ESTIMATION TABLE.**

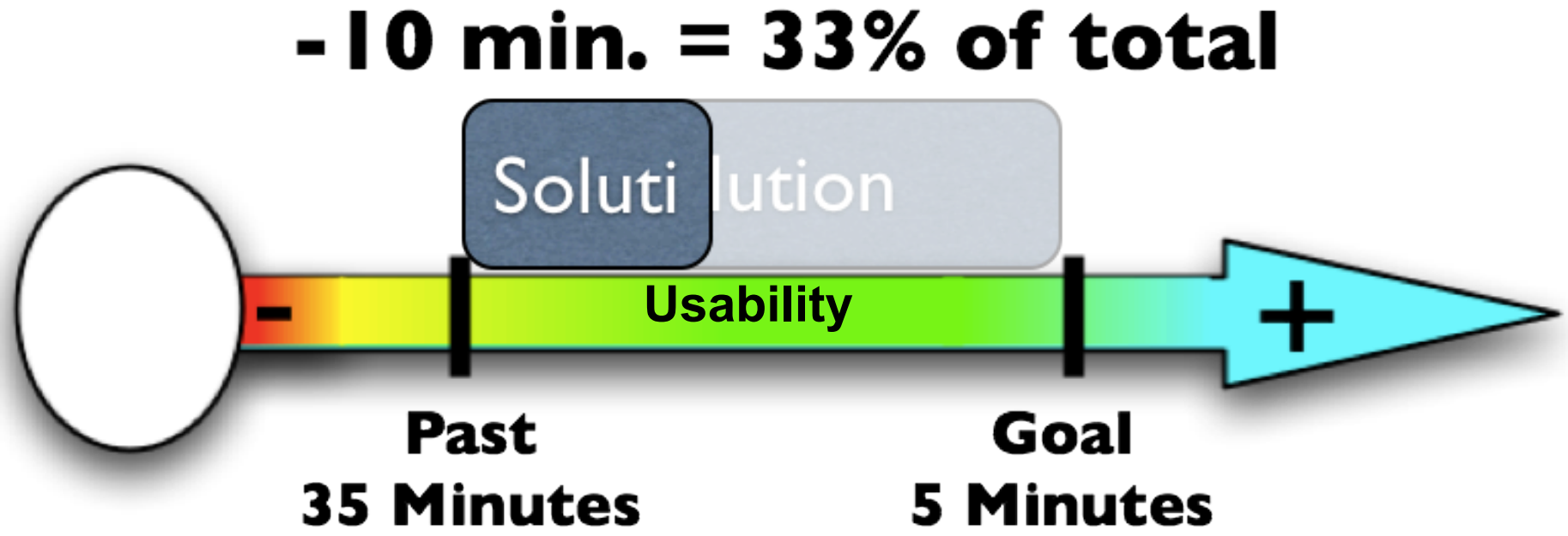
# Why is ‘Stakeholders’ Lean Agile for Managers?

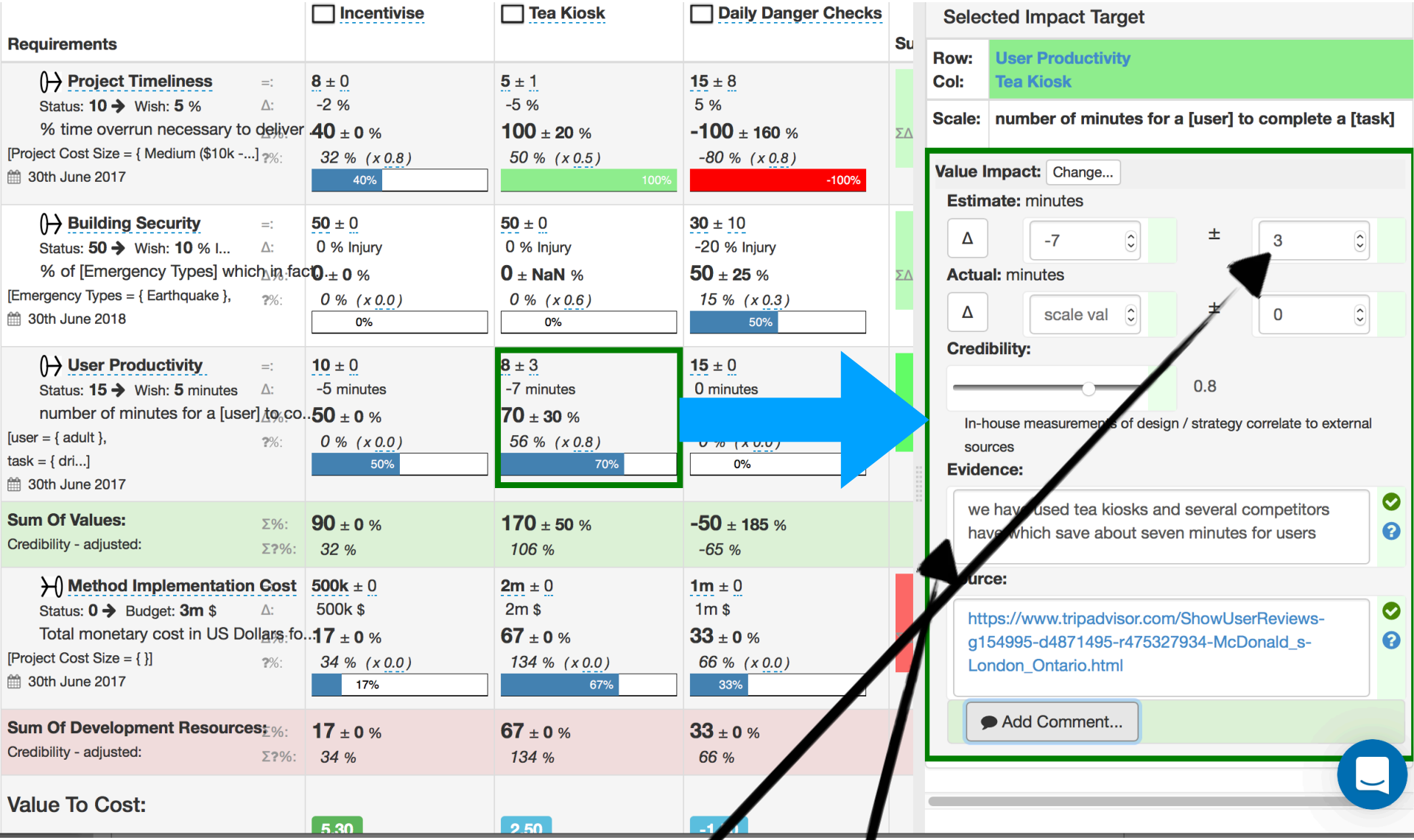
- Management:
  - responsibility for the ‘big picture’
- Lean:
  - preventing bad news, too late
- Agile:
  - you can decompose and prioritize stakeholders and their needs, in the delivery stream

# • The value-driven IT-architecture process using the Impact Estimation Table (IET)

- All strategies
  - that we suggest
- need to be justified
- by *estimates*
- of their *impacts*
- on all concurrent objectives (top 10)
- and all concurrent resource budgets

# Assuring that Designs give Qualities






We estimate benefits based on facts, evidence, and consider ‘uncertainty’ (10±6)

see needsandmeans.com, free app

# Impact Estimation: Value-for-Money Delivery Table

| **STRATEGIES ➔**<br><br>**OBJECTIVES** | Technology Investment | Business Practices | People | Empow-erment | *Principles of IMA Management* | Business Process Re-engineering | SUM |
|---|---|---|---|---|---|---|---|
| Customer Service<br>?➔0 Violation of agreement | 50% | 10% | 5% | 5% | 5% | 60% | 185% |
| Availability<br>90% ➔ 99.5% Up time | 50% | 5% | 5-10% | 0 | 0 | 200% | 265% |
| Usability<br>200 ➔ 60 Requests by Users | 50% | 5-10% | 5-10% | 50% | 0 | 10% | 130% |
| Responsiveness<br>70% ➔ ECP's on time | 50% | 10% | 90% | 25% | 5% | 50% | 180% |
| Productivity<br>3:1 Return on Investment | 45% | 60% | 10% | 35% | 100% | 53% | 303% |
| Morale<br>72 ➔ 60 per mo. Sick Leave | 50% | 5% | 75% | 45% | 15% | 61% | 251% |
| Data Integrity<br>88% ➔ 97% Data Error % | 42% | 10% | 25% | 5% | 70% | 25% | 177% |
| Technology Adaptability<br>75% Adapt Technology | 5% | 30% | 5% | 60% | 0 | 60% | 160% |
| Requirement Adaptability<br>? ➔ 2.6% Adapt to Change | 80% | 20% | 60% | 75% | 20% | 5% | 260% |
| Resource Adaptability<br>2.1M ➔ ? Resource Change | 10% | 80% | 5% | 50% | 50% | 75% | 270% |
| Cost Reduction<br>FADS ➔ 30% Total Funding | 50% | 40% | 10% | 40% | 50% | 50% | 240% |
| ***SUM IMPACT FOR EACH SOLUTION*** | ***482%*** | ***280%*** | ***305%*** | ***390%*** | ***315%*** | ***649%*** | |
| Money % of total budget | 15% | 4% | 3% | 4% | 6% | 4% | |
| Time % total work months/year | 15% | 15% | 20% | 10% | 20% | 18% | |
| ***SUM RESOURCES*** | ***30*** | ***19*** | ***23*** | ***14*** | ***26*** | ***22*** | |
| **BENEFIT/RESOURCES RATIO** | ***16:1*** | **14:7** | **13:3** | **27:9** | **12:1** | **29.5 : 1** | |

© Gilb.com

# Graphical presentation of organizational architecture impact and costs

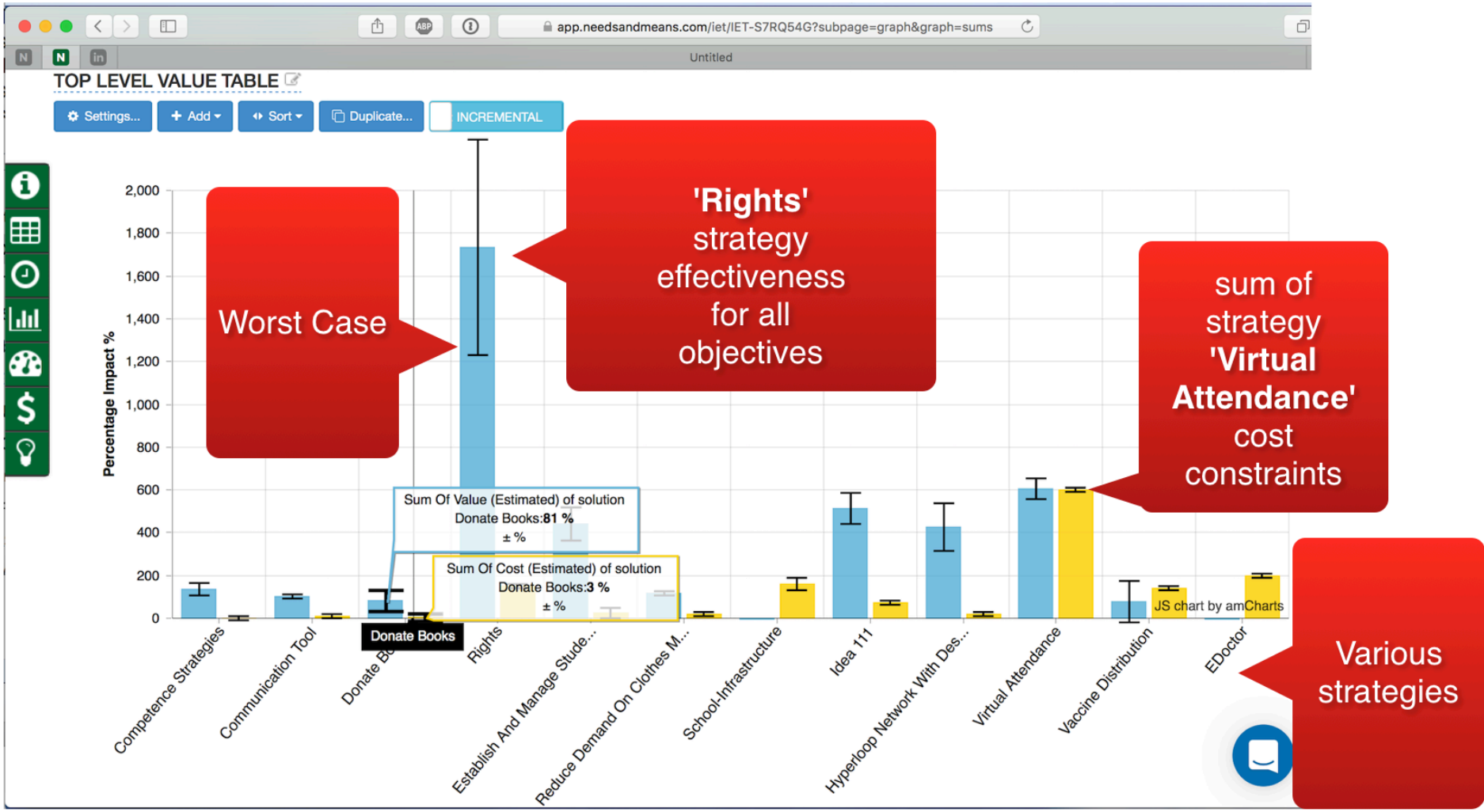


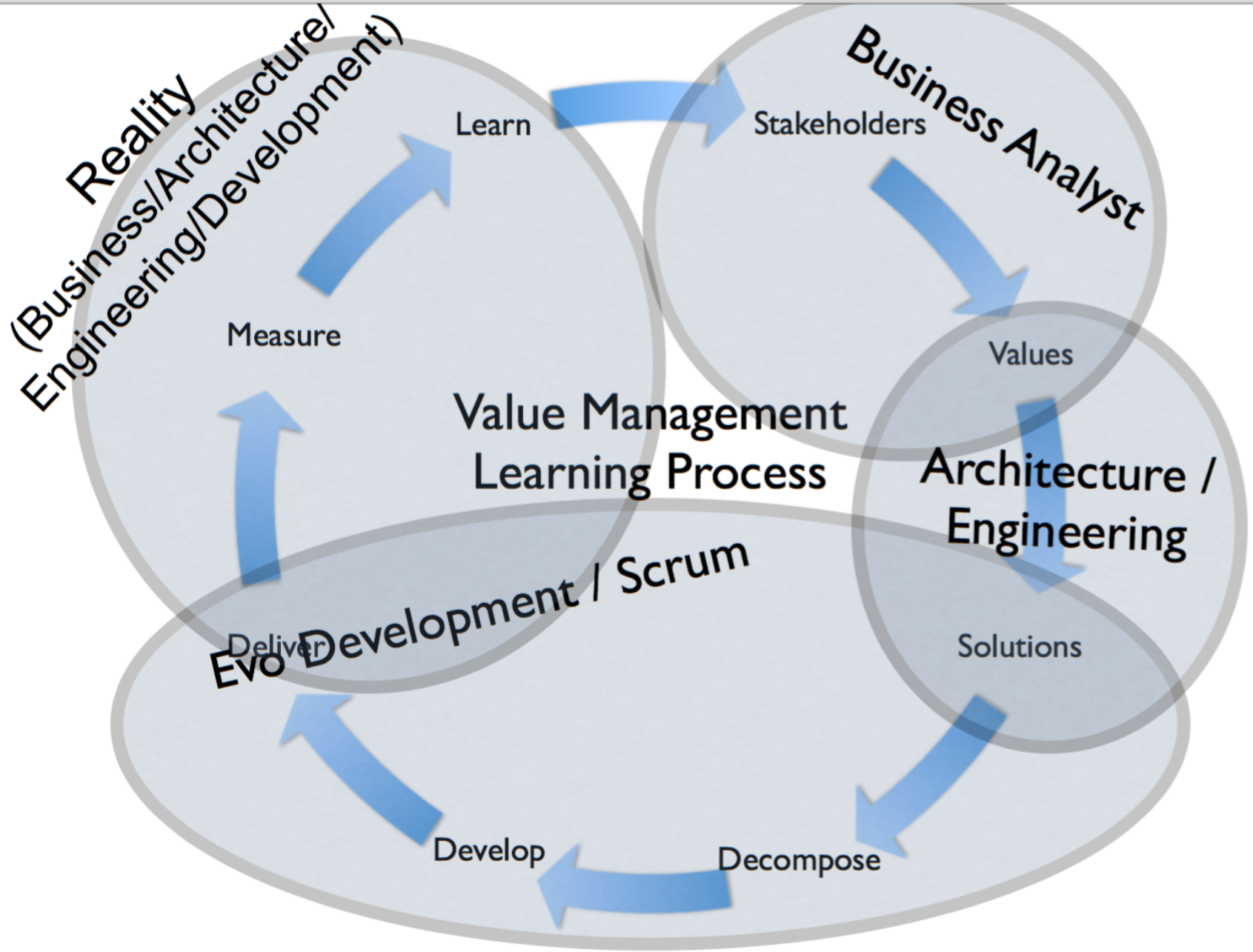

**DIAGRAM 1: THE IMPACT ESTIMATION DISCIPLINE CAN GIVE US A *QUANTIFIED* OVERVIEW OVER THE OVERALL (ALL OBJECTIVES) EFFECTIVENESS OF ALL PROPOSED STRATEGIES.**
**Source: needsandmeans.com [4] tool as used at workshop, Oslo 2017 OSWA Meetup.**

# Why is ‘Impact Estimation’ useful for Management/Lean/Agile ?

- Management
  - managing all requirements and architecture
- Lean
  - Early upstream insight into problems and side effects
- Agile
  - it is a tool for decomposition into prioritized agile delivery steps

# • The Agile ‘Evolutionary’ Project Management process.

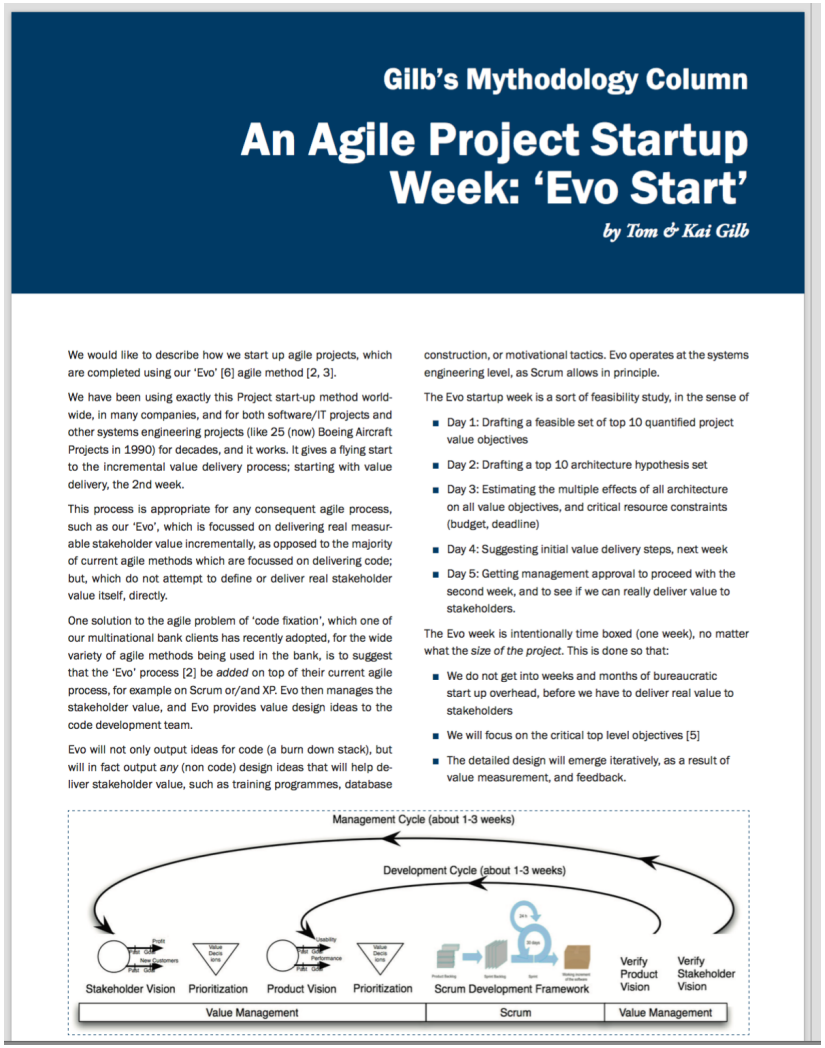
- your agile process *cannot* be primarily focussed on delivering ‘code’
- it must be *‘systems oriented’* (not just ‘IT’)
- it must focus on delivering *measurable* value improvement
- traditional agile does NOT focus on *measurable* values
- *you* have to add ‘value’ mechanisms
  - to your agile own framework


Reality
(Business/Architecture/
Engineering/Development)
Learn
Measure
Business Analyst
Stakeholders
Values
Value Management
Learning Process
Architecture /
Engineering
Evo Development / Scrum
Deliver
Solutions
Develop
Decompose

© Kai@Gilb.com

# • The One-Week Project-Startup Process to launch *real* value delivery.

- We practice a 1-week project startup
- followed by *weeks* of value-stream delivery
- meaning; increments of the value objectives, towards Goal levels
- day 1, the top 10 critical value objectives are drafted
- day 4 the next week value delivery 'sprint' is planned



Gilb's Mythodology 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 world-wide, 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 measur-able 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 de-liver 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.



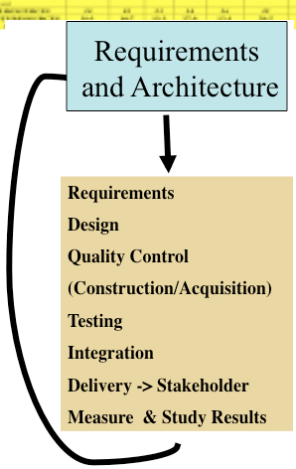


www.gilb.com/dl568

# The ´Evo´ Planning Week at DoD

- Monday
  - Define top Ten critical objectives, quantitatively
  - Agree that thee are the main points of the effort/project
- Tuesday
  - Define roughly the top ten most powerful strategies
  - for enabling us to reach our objectives on time
- Wednesday
  - Make an Impact Estimation Table for Objectives/Strategies
  - Sanity Test: do we seem to have enough powerful strategies to get to our Goals, with a reasonable safety margin?
  - A tool for decomposing the value steps and seeing best value for resources
- Thursday
  - Divide into rough delivery steps (annual, quarterly)
  - Derive a delivery step for 'Next Week'
- Friday
  - Present these plans to approval manager (Brigadier General Pellicci)
  - get approval to deliver next week
  - (they can´t resist results next week!


US Army Example: PERSINSCOM






© Gilb.com

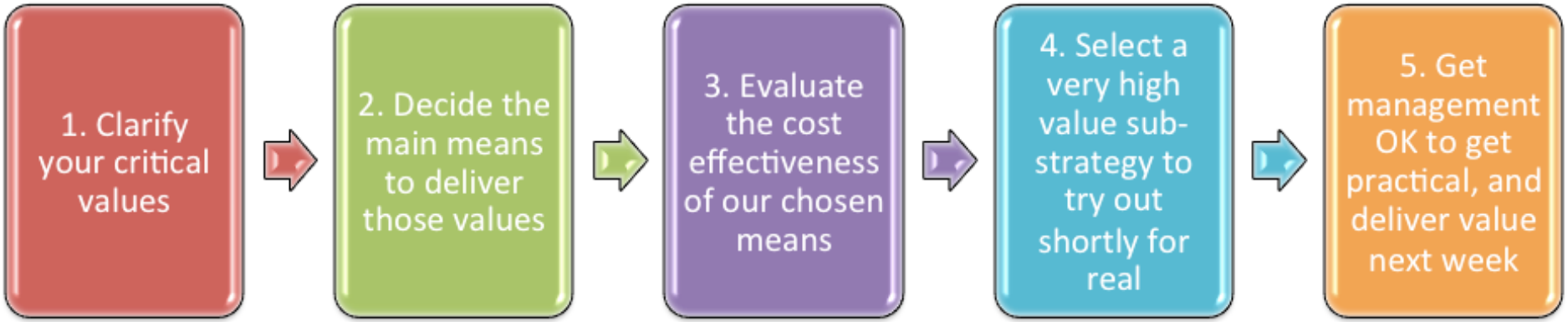
# Startup Week: **Process**


**An Agile Project Startup Week**

**Gilb's Mythodology Column**

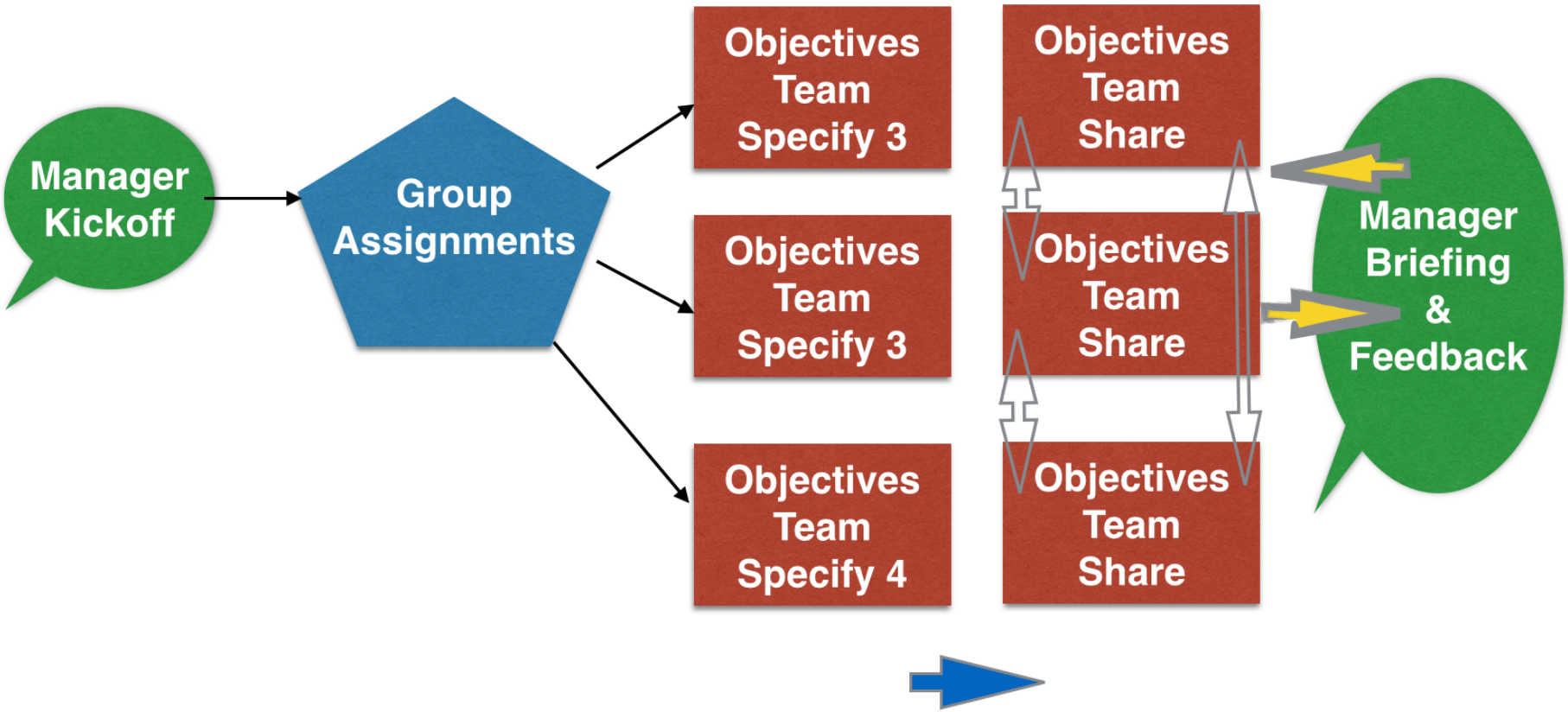
**www.gilb.com/dl568**

# Startup Week **Purposes**




Evo Startup Standard, Jan 12 2013 http://www.gilb.com/dl562

# The First Day of the Startup Process. 'Top Ten Critical Values', a *quantification* process



# So how does ‘Evo’ relate to Management/Lean/agile?

- Management
  - a primary project management method
- Lean:
  - Early feedback, learning and correction
- Agile
  - the best agile method for those who are focussed on value, quality and costs (vs. ‘functions’ and ‘use cases’ focus)

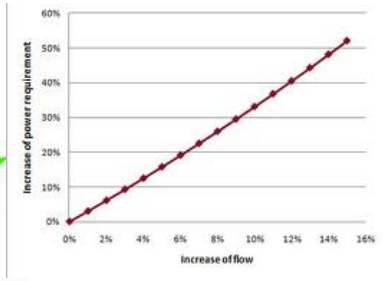
# Decomposition

dividing things up
to simplify
to prioritize
to deliver early

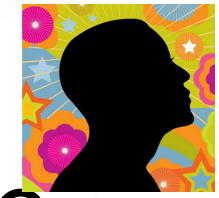
# 1 1 1 1 1 1 Weekly Value Delivery Decomposition Paradigm

–1% increase at least

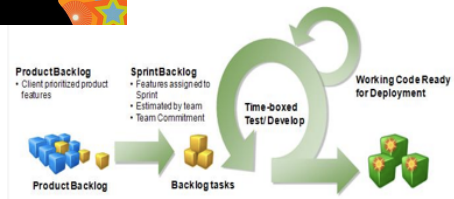
–1 stakeholder



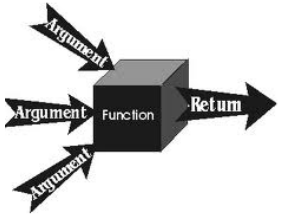

–1 quality or value



–1-week delivery cycle

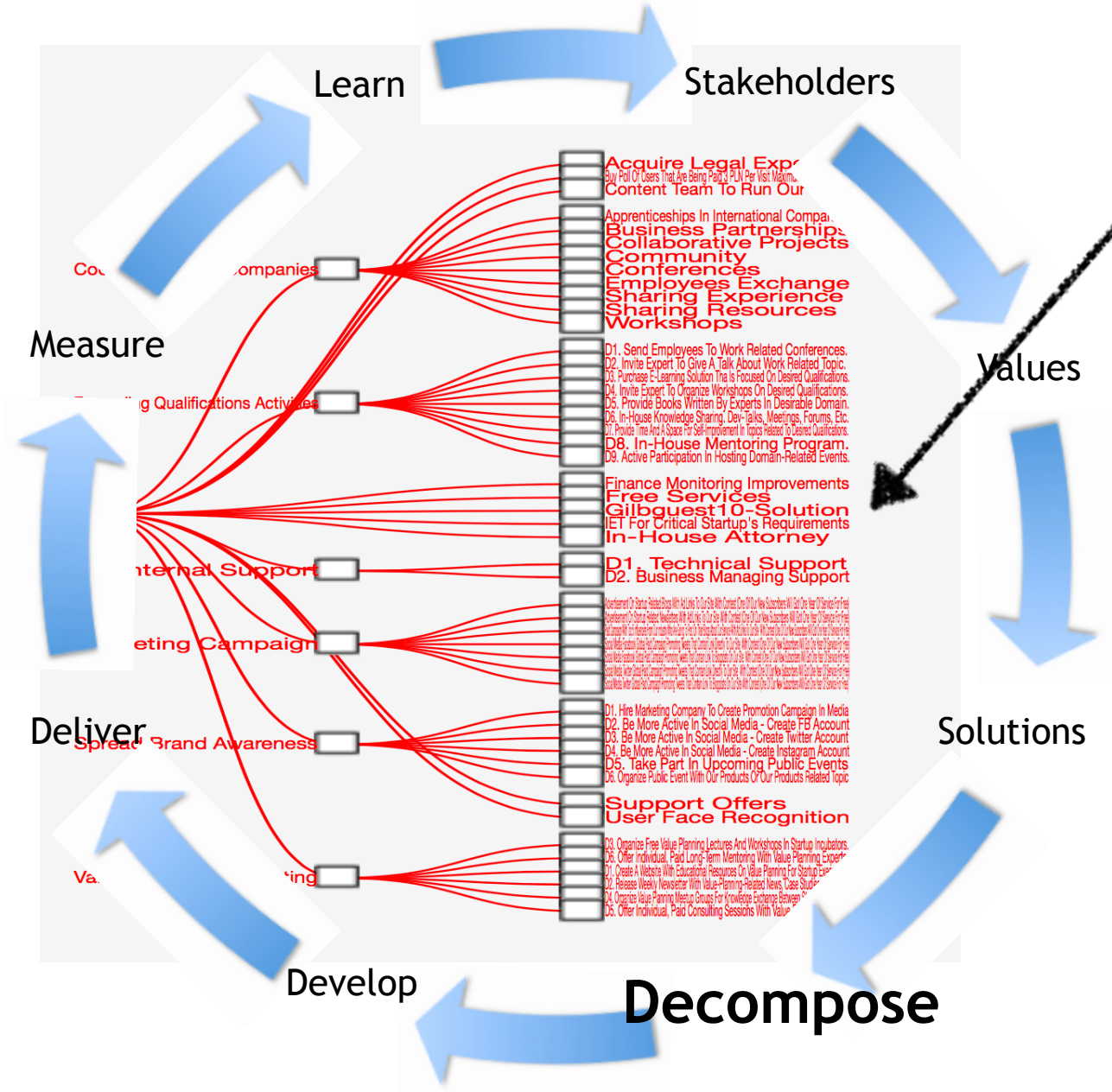



–1 function focus




–1 design used

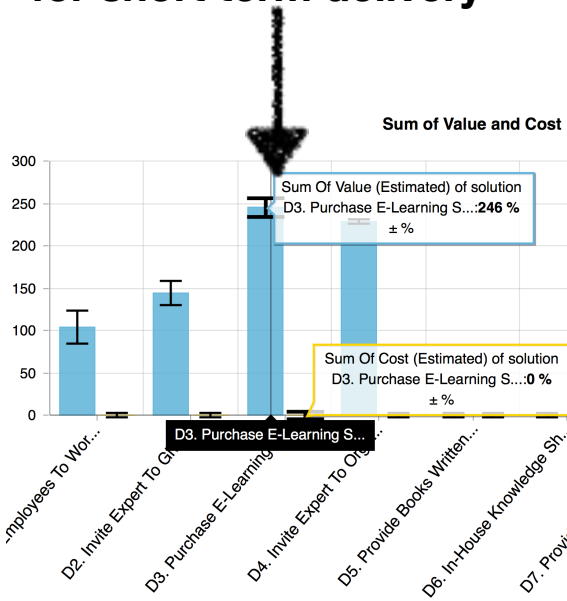
http://www.gilb.com/DL451




**The solutions can be <u>decomposed</u> by 10x or 100x**

**And we can <u>estimate</u> the solution <u>sub-component</u> value and cost,**

**so as to prioritize the best value/cost for short term delivery**



# How does ‘Decomposition’ help us?

Management
- better value for cost flow to stakeholders
- do a little and learn a lot

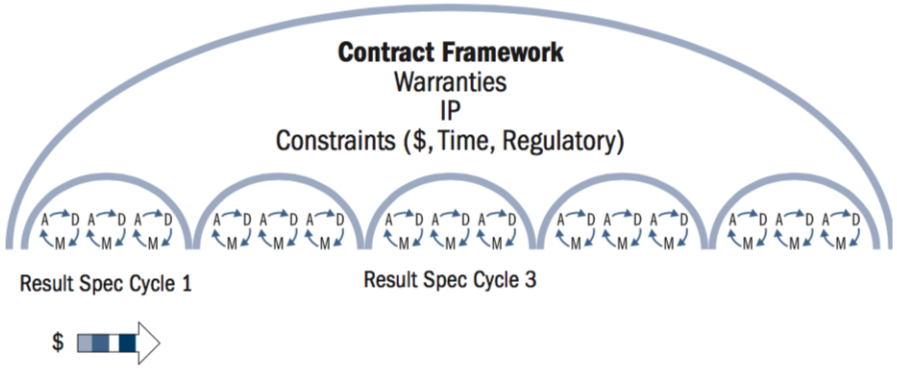
Lean
- prevent large failures, learn fast-fix fast

Agile
- better cause-and-effect understanding
- better agile response to problems

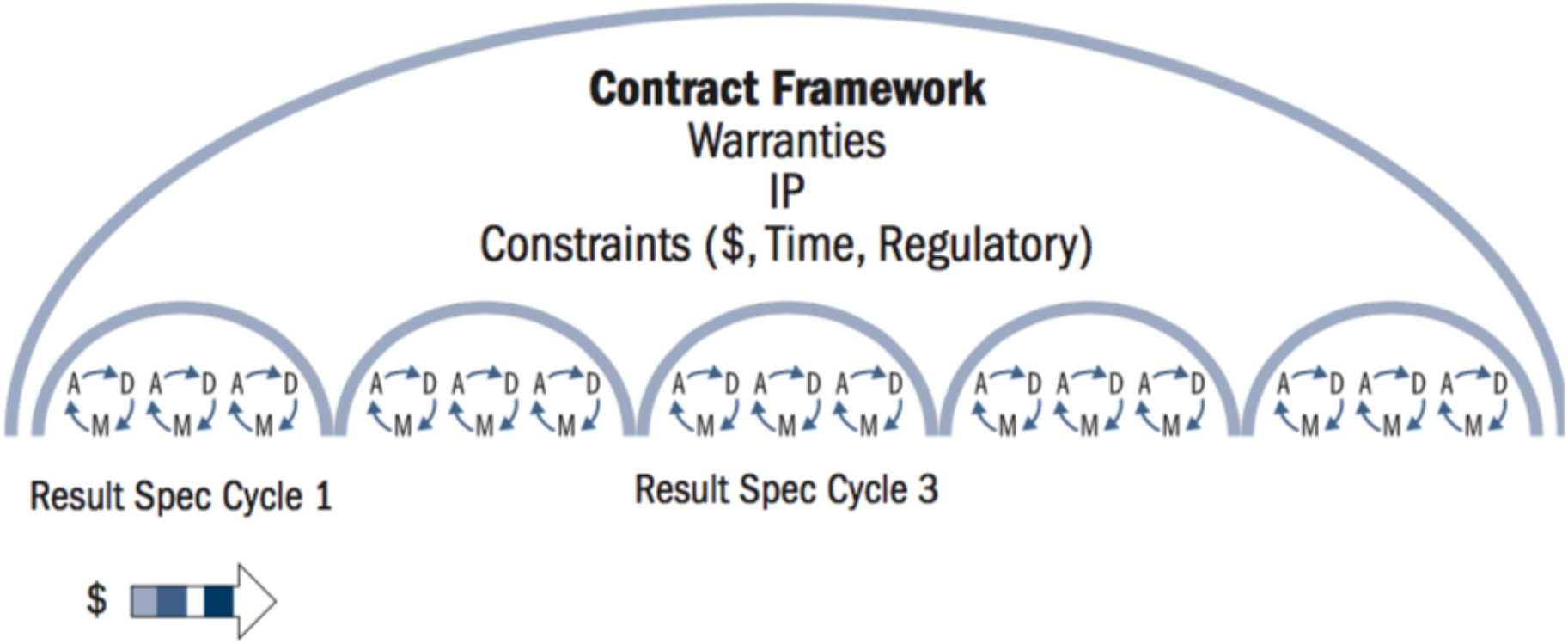
- **The ‘Flexible Contracts’ subcontracting for Value, Process.**

Extremely agile and lean

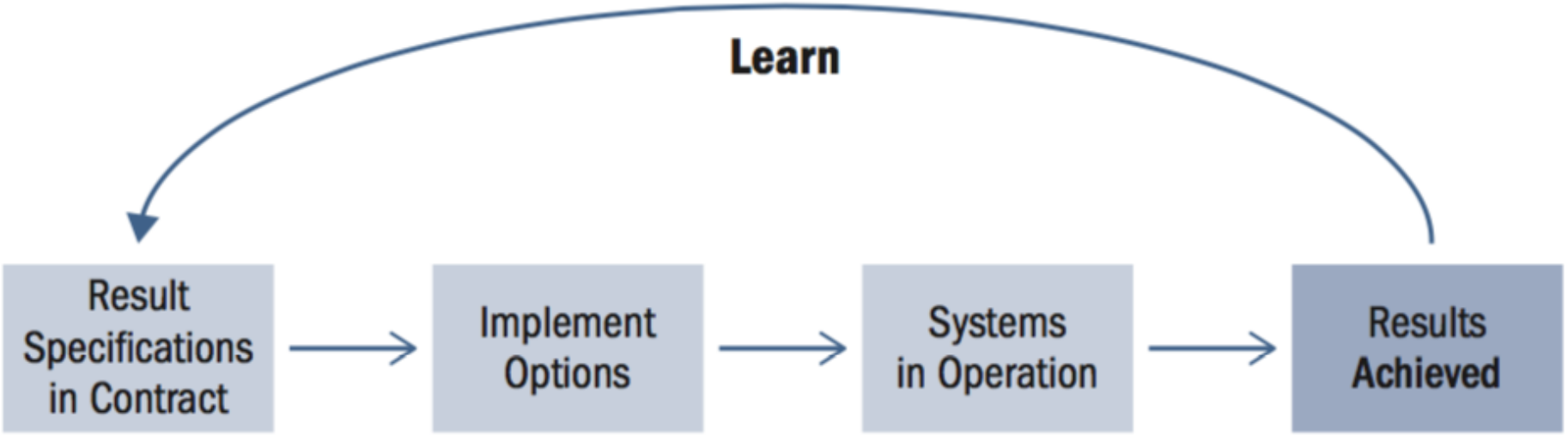
- **management needs to pay for value delivered**
- **not for work done**
- **especially not pay when expected value is not delivered**
- **we need to adjust contract ‘deals’**
- **on every new delivery cycle**
- **based on experience**
- **and changed needs**
- **we need to ‘motivate’ sub-contractors to deliver the value we expect and need**



# 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 serious 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 develop-ment process. | Testing occurs throughout the development process, providing feedback for improve-ments. |
| Success is measured by refer-ence to conformance with the change-controlled contract. | Successs is measuered, cycle by cycle, by requirements delivered, driving value to the customer. |

# WHAT IS A FLEXIBLE CONTRACT?

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

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

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

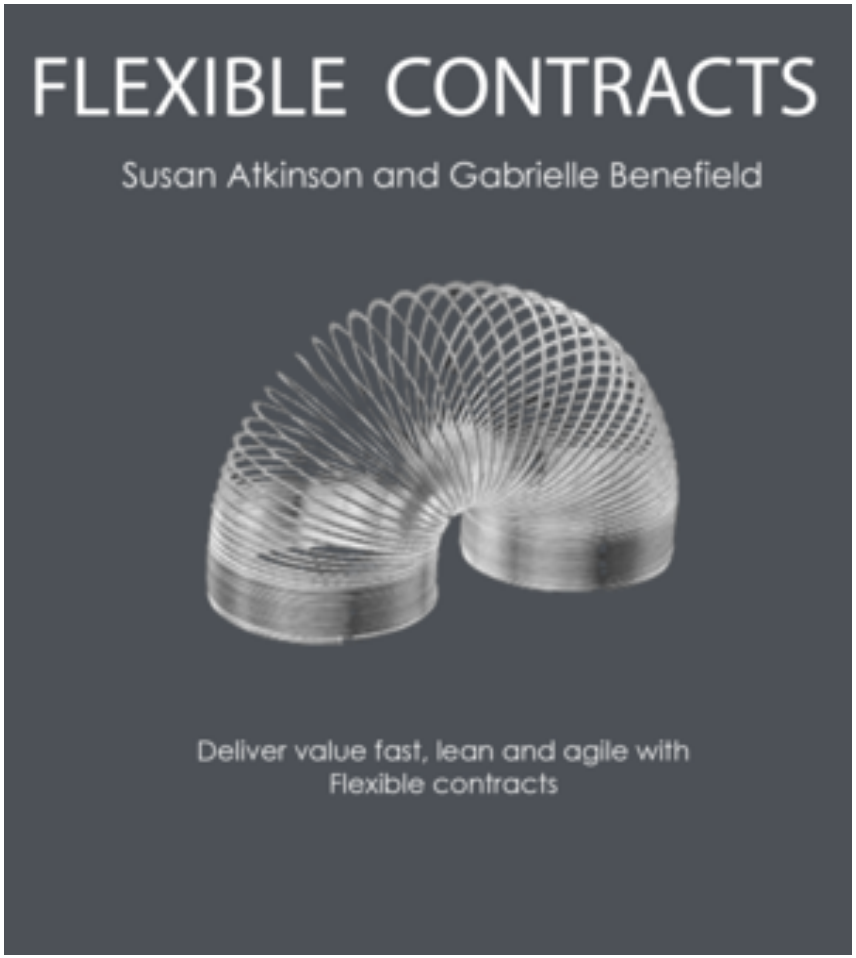
(from contract template)

**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:<br>• Unit of measure:<br>• party responsible for conducting measurement:<br>• Method for measurement:<br>• Frequency of measurement:<br>• Baseline (starting point): | <br>i.e. the metric used to measure e.g. time, percentage or number<br>i.e. a named person or group responsible for conducting the measurement e.g. the Customer<br>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<br>i.e. The period of time when measurements will be taken e.g. every ***[2 weeks]*** with their end-users<br>i.e. the baseline that will be used as the starting point against which to compare results |

# Credits for most slides to




Forthcoming Book

- www.flexiblecontracts.com
- https://www.linkedin.com/groups/Flexible-Agile-contracts-7460556/about
- http://www.gilb.com/dl581 Paper

- **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 in tune with Evo and Planguage methods in my writings, such as**
- **Competitive Engineering (2005), and Value Planning (2016-18, digital)**

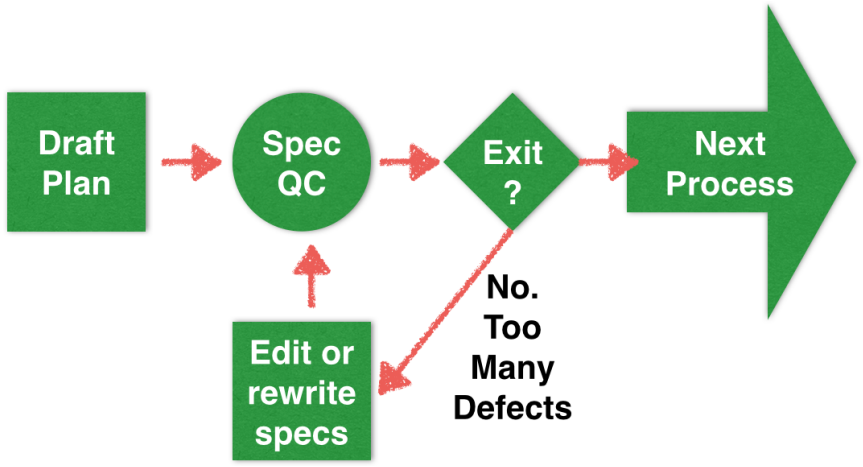
# How does Flexible Contracting help us?

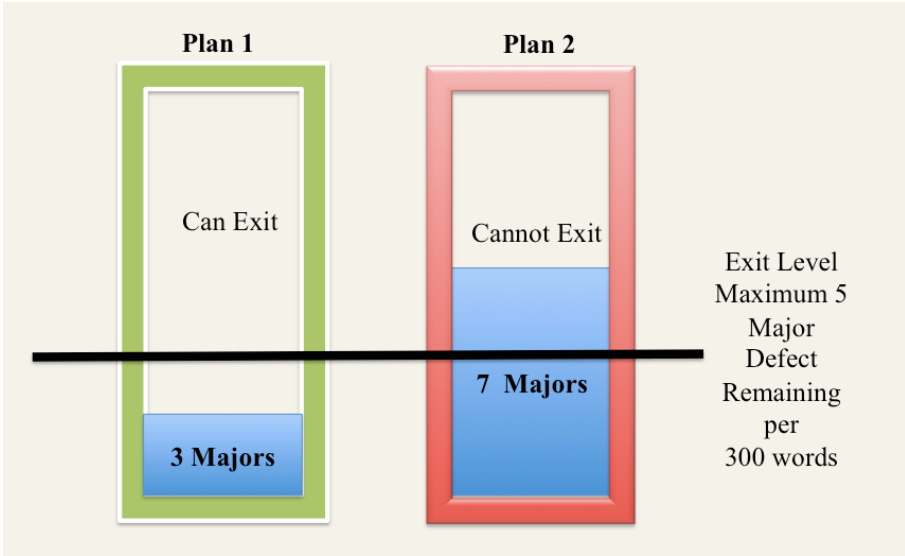
- Management
  - no cure, no pay
  - don’t waste money
  - select suppliers who deliver results
- Lean
  - learn early
- Agile
  - change suppliers when they are proven incompetent

# The Agile Specification Quality Control process

**for lean (early, prevents defect injection) measurement of quality of requirements, architecture specs, and contracts**

- Our IT planning documents are heavily polluted
- with dozens of ‘major defects’ per page
- we need to measure defects by sampling
- and we need to refuse to ‘exit’ garbage out
- this lean approach can improve productivity 2x and 3x (Intel)

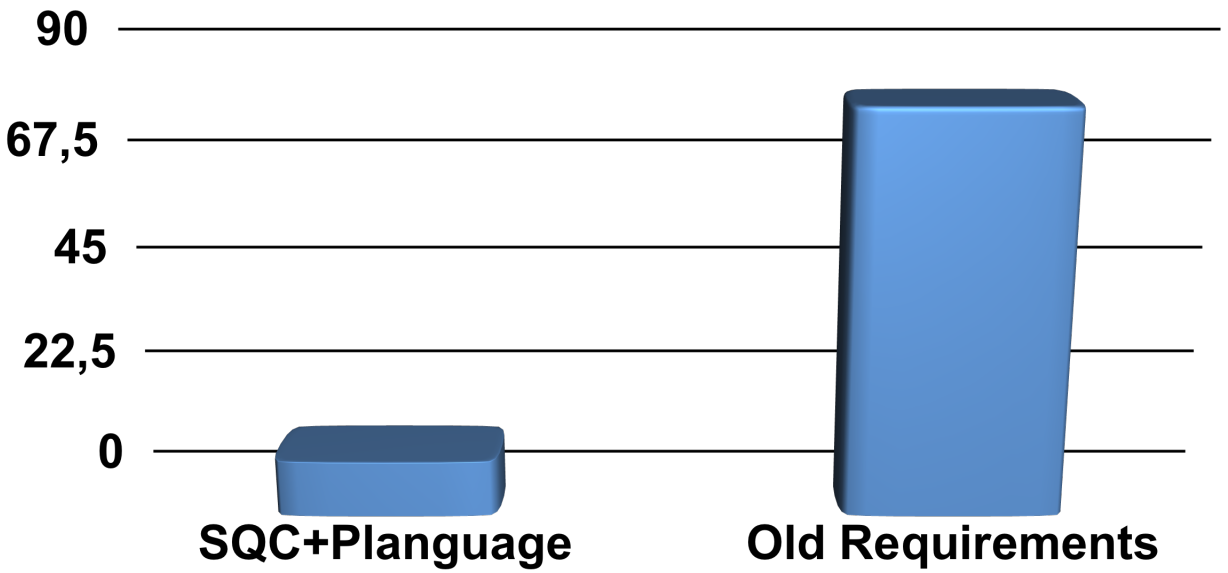


# Reducing unintelligible IT requirements

**from 80/page to 10/page in 6 months**
**London, Citigroup**
**Spec QC/Extreme Inspection + Planguage Requirements**

Major defects/page on 1st Quality Control




© Tom@Gilb.com

See Slide Note for details

Source Eric Simmons, erik.simmons@construx.com 25 Oct 2011
http://selab.fbk.eu/re11_download/industry/Terzakis.pdf

# Extreme Quality Management using Planguage and my Spec QC

Application of Specification Quality Control by a SW team resulted in the following defect density reduction in requirements over several months:

| 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% |

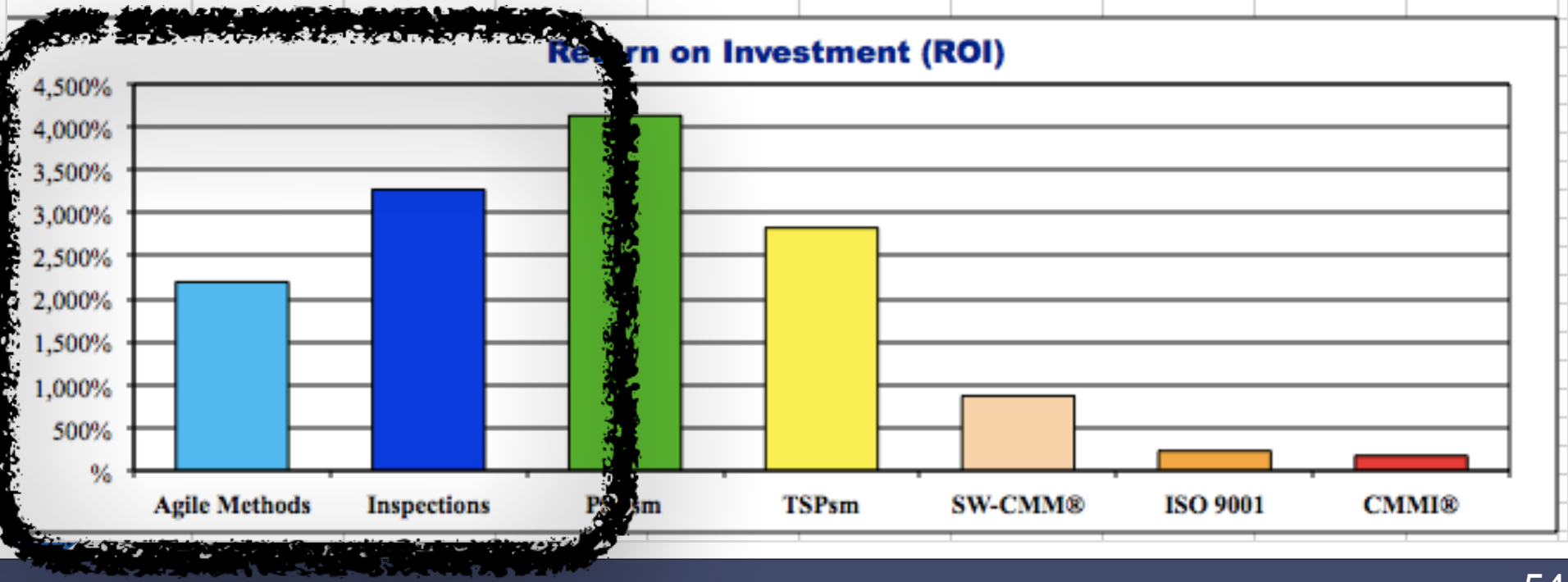
Downstream benefits:
- Scope delivered at the Alpha milestone increased 300%, released scope up 233%
- SW defects reduced by ~50%
- Defects that did occur were resolved in far less time on average
- teams typically exit with densities ranging from 5 majors per page (600 words) to 1 defect in a couple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

© Gilb.com

# Value for Money
# Inspection and CMMI
## David Rico, http://davidfrico.com

### ROI Comparison

| | Costs | Benefits | B/CR | ROI% | NPV | BEP | Cost/Person | Risk | ROA |
|---|---|---|---|---|---|---|---|---|---|
| Agile Methods | $188,199 | $4,321,798 | 23:1 | 2,196% | $3,554,026 | $8,195 | $47,050 | 52.19% | $4,175,664 |
| Inspections | $82,073 | $2,767,464 | 34:1 | 3,272% | $2,314,261 | $51,677 | $20,518 | 26.78% | $2,703,545 |
| PSPsm | $105,600 | $4,469,997 | 42:1 | 4,133% | $3,764,950 | $945 | $26,400 | 6.44% | $4,387,756 |
| TSPsm | $148,400 | $4,341,496 | 29:1 | 2,826% | $3,610,882 | $5,760 | $37,100 | 37.33% | $4,225,923 |
| SW-CMM® | $311,433 | $3,023,064 | 10:1 | 871% | $2,306,224 | $153,182 | $77,858 | 83.51% | $2,828,802 |
| ISO 9001 | $173,000 | $569,841 | 3:1 | 229% | $320,423 | $1,196,206 | $43,250 | 98.66% | $503,345 |
| CMMI® | $1,108,233 | $3,023,064 | 3:1 | 173% | $1,509,424 | $545,099 | $277,058 | 100.00% | $2,633,052 |



# How does Spec QC/Exit Help

- Management
  - quantified management of all software processes *(facts, not just 'beliefs')*
  - motivated practical teaching, all people, all processes (req., arch., code, test, contract)
- Lean
  - prevents bad work going downstream
- Agile
  - can be done any stage, every day, adapt to new people, new processes

# 1999-2016 Observations by Erik Simmons, Intel: **It Scales**

**January 8, 2016 Email.**

**"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 CE, 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").**

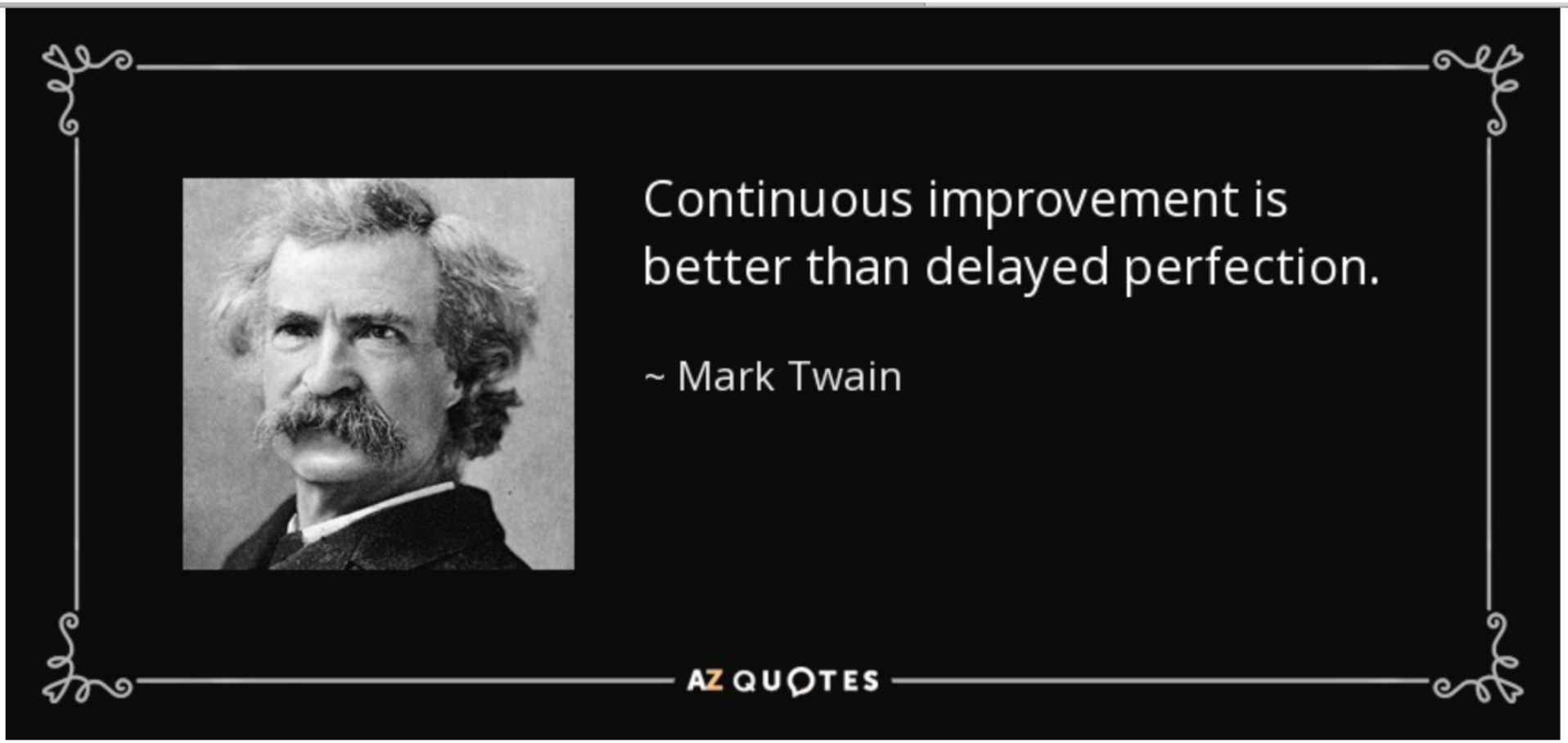
**Anyone who sees a random page of your new paper would probably not guess the topic is scaling (unless you happen to mention that in the text on that particular page). CE does not scale. It doesn't need to.**

**Your work for decades has been focused on a very good set of these. SQC, for example, works on any size specification. It does not (need to) scale."**

**"Some Advanced Tools and Principles for Scaling Agile Projects - Agile Engineering."**

**40 practical Engineering ideas for scaling agile development successfully all the time.**
**A very short pdf paper, supported by references to necessary detail.**
**Not least the new LeanPub.com/ ValuePlanning book**

**http://www.gilb.com/dl865**




**Samuel Langhorne Clemens** (November 30, 1835 – April 21, 1910),[1] better known by his pen name **Mark Twain**, was an American author and humorist. He wrote *The Adventures of Tom Sawyer* (1876) and its sequel, *Adventures of Huckleberry Finn* (1885),[2] the latter often called "The Great American Novel".
(and friend of Nikolai Tesla!)

# Tom's Ten Principles of Lean and Agile IT System Management. © gilb.com, 2016-8

1. quantify *critical* improvement objectives
2. estimate *multiple* impacts of strategies
3. reject *polluted* specifications
4. plan for 1 week only, before starting 'value delivery'
5. deliver *some* value every week, or 2% of time of project
6. measure *real* value, and costs, and learn fast
7. contract for *value delivery*, not for work done
8. operate at the *systems* level, not the 'code' level
9. let *critical stakeholders* decide your critical objectives
10. Keep it simple: 'top 10 objectives quantified' is 'master', everything else is a 'servant'

# Digital Book

**My current book manuscript Value Planning**
**50% Off**
**Use this link: https://goo.gl/XGMgwg**
**Code: WS50**

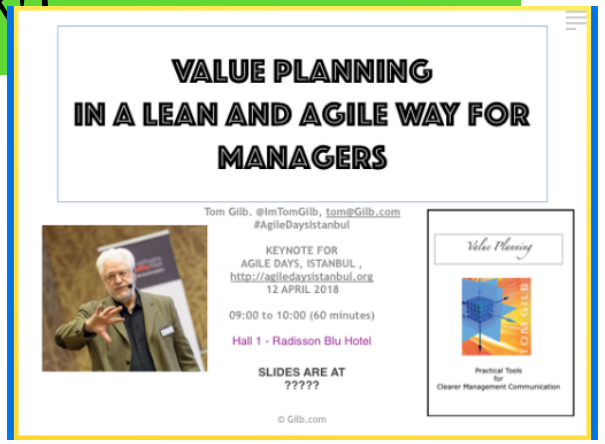
**Who wants to translate this to Turkish?**
**Or translate a 20 or 60 page subset of it?**
**Email me: tom@gilb.com**

**Email me if you want to *read* a digital copy of 'Competitive Engineering' (for free).**
**(*Paper* copy from Amazon!),**

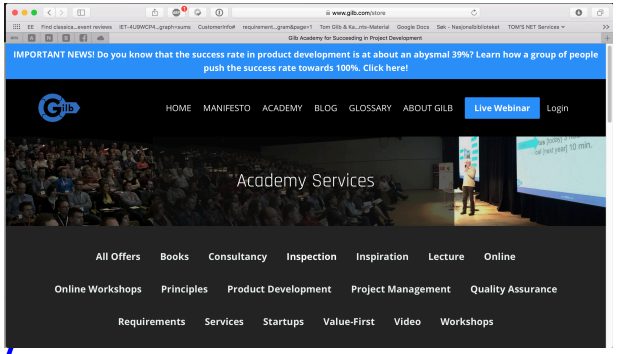

**https://www.gilb.com/store/eooAAw85**

Read about & Sign Up NOW to 4 weeks FREE access

# Konuşmamı dinlediğiniz için teşekkürler.
(Thanks for listening to my talk!)




- THESE SLIDES ARE AT
  - WWW.GILB.COM
    http://concepts.gilb.com/file24
- FEEL FREE TO SHARE AND TWITTER
- @ImTomGilb




- OR cite this talk on LINKEDIN
  - TO SPREAD THE WORD
  - https://www.linkedin.com/in/tomgilb/
  - email contact: tom@gilb.com
  - I live in Oslo and London.

## THIS TALK CONTAINED IDEAS IN THE FOLLOWING AREAS

**Managers need to lead in specific ways in order for their projects to succeed.**

**They need to lead by making the 'values delivered' the priority.**

**<u>Not</u> by focussing on the the IT development task itself.**

**They need to set value objectives quantitatively.**

**They need to start real measurable value deliveries, very early, and very frequently.**

**They need to measure 'value delivered' and 'costs incurred' incrementally.**

**They need to contract for incremental value delivery, and pay for value delivered, not just 'work done'.**

**This is in the spirit of both agile and lean processes: but these are just frameworks.**

**They need specific tools to do all this in addition to wanting to do it.**

**Managers have to learn new tools for 'value quantification', and add these skills to the management and technical skills of their organization.**

Quantify

Manage

Values