What's Wrong With Requirements Methods?

An analysis of the fundamental failings of conventional thinking about software requirements, and some suggestions for getting it right.

MASTER FULL VERSION

Talk Background Data

- Evening 25-30 minute London SPIN talk 6:15 to 6:45
- _

Paper , Publication

- http://www.coremag.eu/fileadmin/Papers/RQNG_tom_gilb_core_ENG.pdf
- http://www.gilb.com/tiki-download_file.php?fileId=443 (Journal Sw Eng version)
- http://www.testingexperience.com/testingexperience11_09_10.pdf (Test exp version)
  - Registration required

- Presented by Tom Gilb, Independent Consultant, Author, Teacher
  - Tom@Gilb.com
  - www.Gilb.com
  - @ImTomGilb on twitter

What is wrong with Requirements Practice?

- *Lack of*

Critical Value Requirements

Talk Outline
Time permitting, then see slides at www.gilb.com

1. Requirement definition: 'Stakeholder Prioritized End State'
2. Ten Reasons Why Requirements Methods Fail
3. Top Level Critical Objectives: the missing link
4. Don't Mix Ends and Means
5. Requirements are not always 'Required': Intelligent Dynamic Prioritization
6. Stakeholders: not just users and customers!
7. Value Delivery: leading to Systems Thinking, not Software Silos
8. Quantification: not 'Software Poetry' – a basis for real Software Engineering – not mere 'Softcrafting'
9. Rich Specification: Requirement specifications need far more info than the requirement itself!
10. Ten Principles for Successful Requirements Methods.
11. Who or What will Change things?
12. Summary

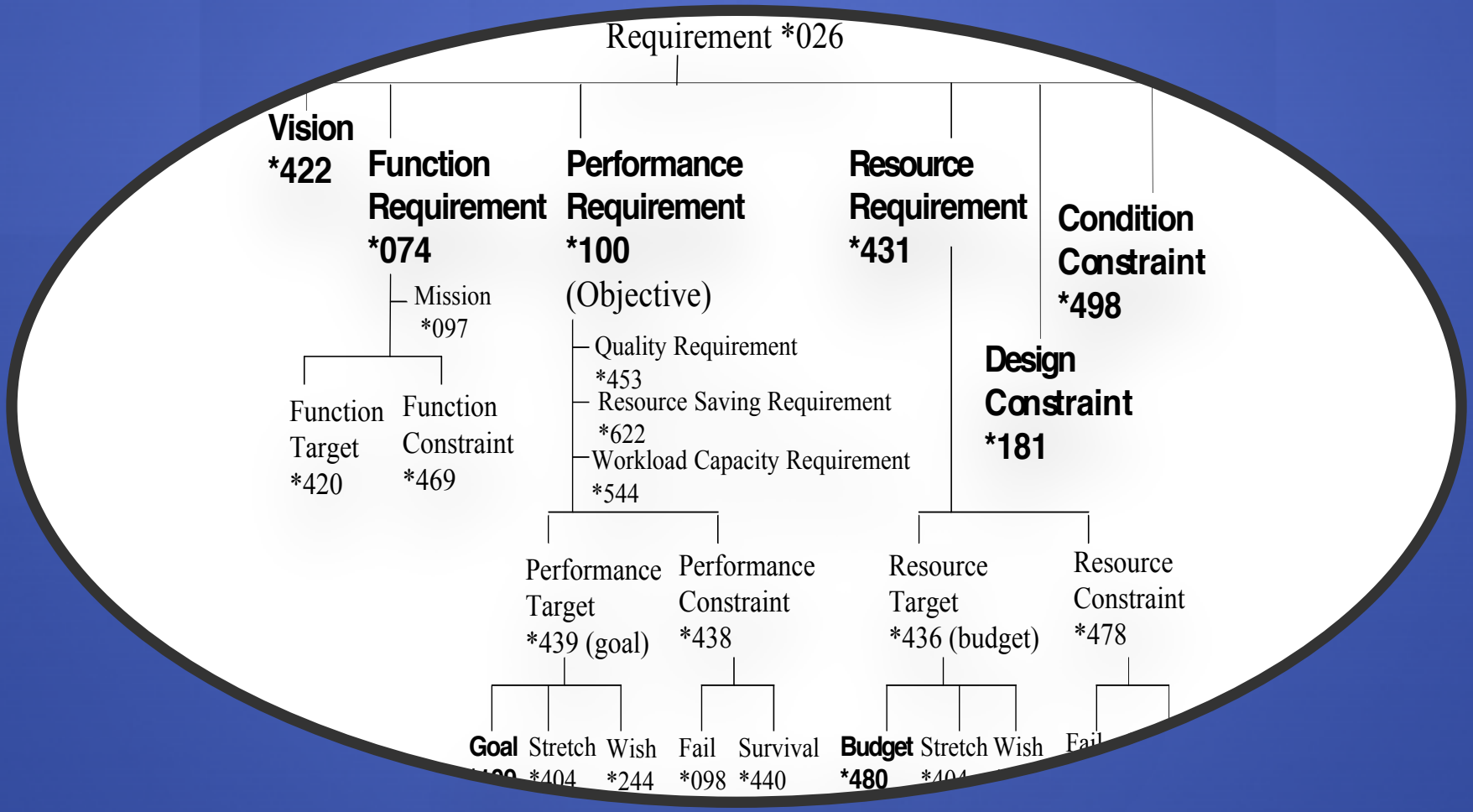
"Requirement" is

"Stakeholder Valued End State"

Source: Gilb, Planguage Concept Glossary 2011 version

http://www.gilb.com/tiki-download_file.php?fileId=386

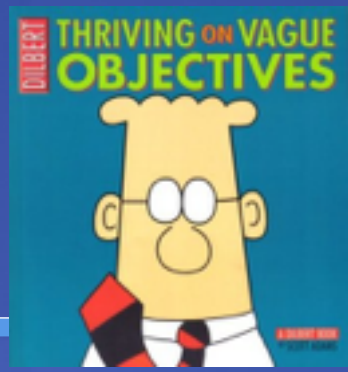
Requirement Types: <-CE, PL



Ten Reasons
Why Requirements Methods Fail

- 1. Focus, not stakeholders
- 2. Designs, not values
- 3. Poetry, not clear
- 4. Function, not quality
- 5. Testable, not constraints
- 6. Requirement, not background
- 7. Single Requirement, not the set
- 8. Assumptions, not rigorous definitions
- 9. Blind acceptance, no real QC
- 10. single level, not multiple levels

The Worst Problem

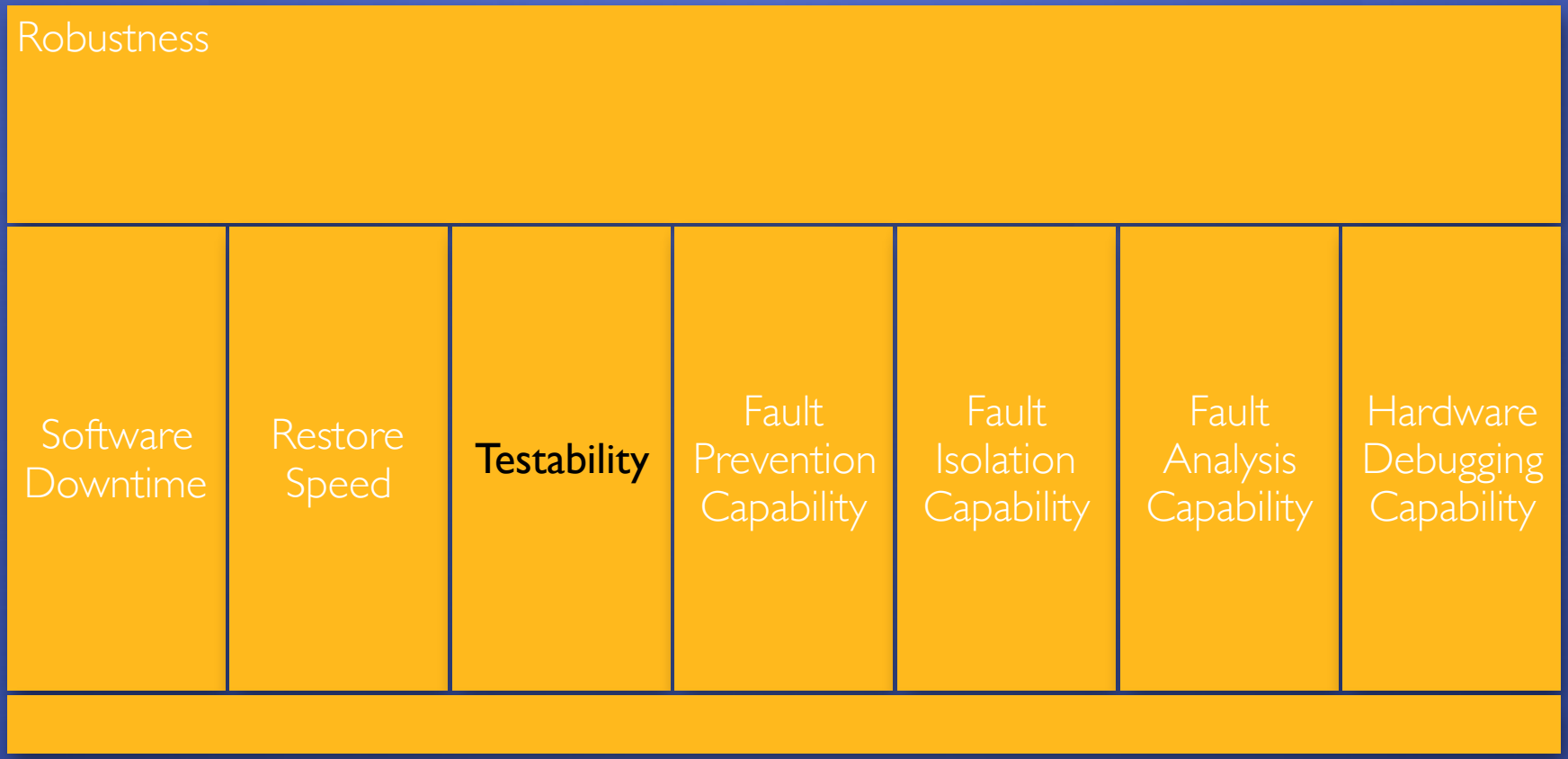


Real Case

Bad Quality
for the
Top Level
Critical
Requirements

- "Make the system much easier to understand and use"
- "Robustness'
  - (See next slide)
- "Richer set of tools for supporting next generation tools and applications"

A Complex Requirement "Robustness"



Testability (part of "Robustness")

Scale: the *duration* of

a defined [Volume] of testing,

of a defined [Type],

by a defined [Skill Level] of system operator,

under defined [Operating Conditions]

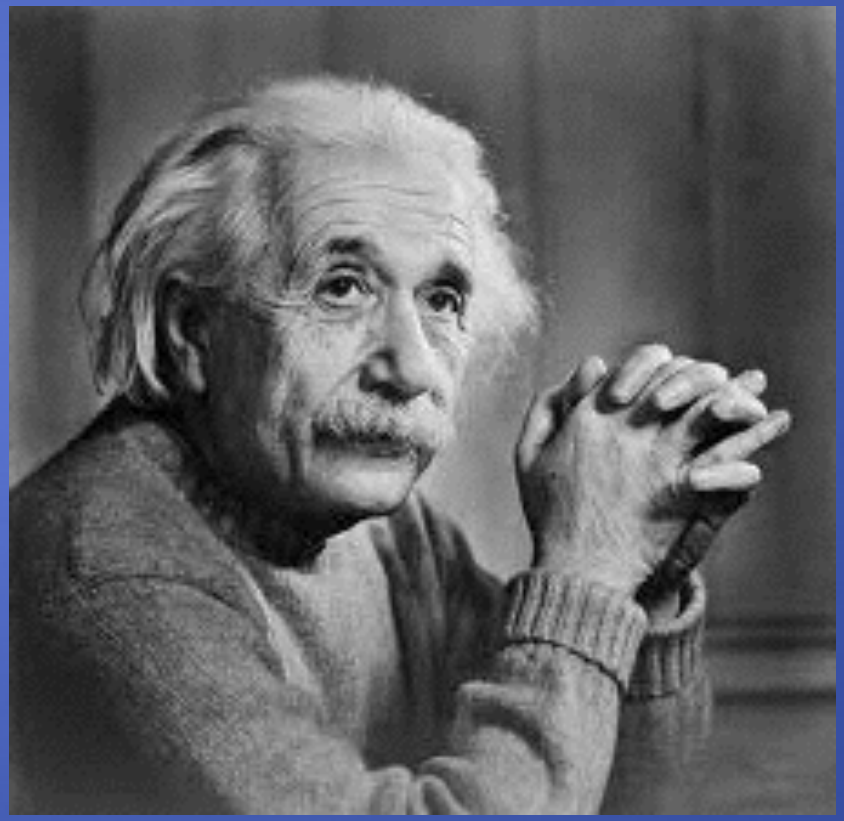
Fail .. 20 minutes → **Goal.... < 10 minutes** → **Stretch ... 1 minute**

Previous Case: Observation

- Management lost over $100 million on this project, and 8 years time,
- Because they failed to clarify (quantify!) critical requirements
- 1 days work

4. Don't Mix Ends and Means

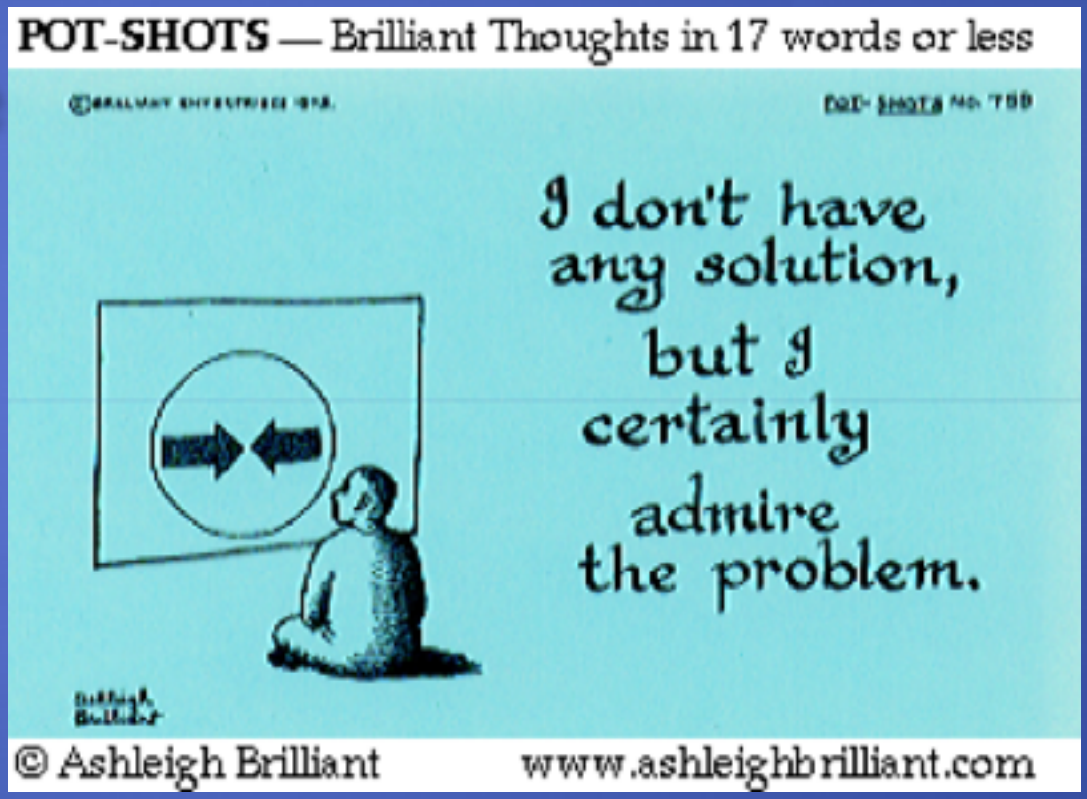
- *"Perfection of means*
- *and confusion of ends*
- *seem to characterize our age."*



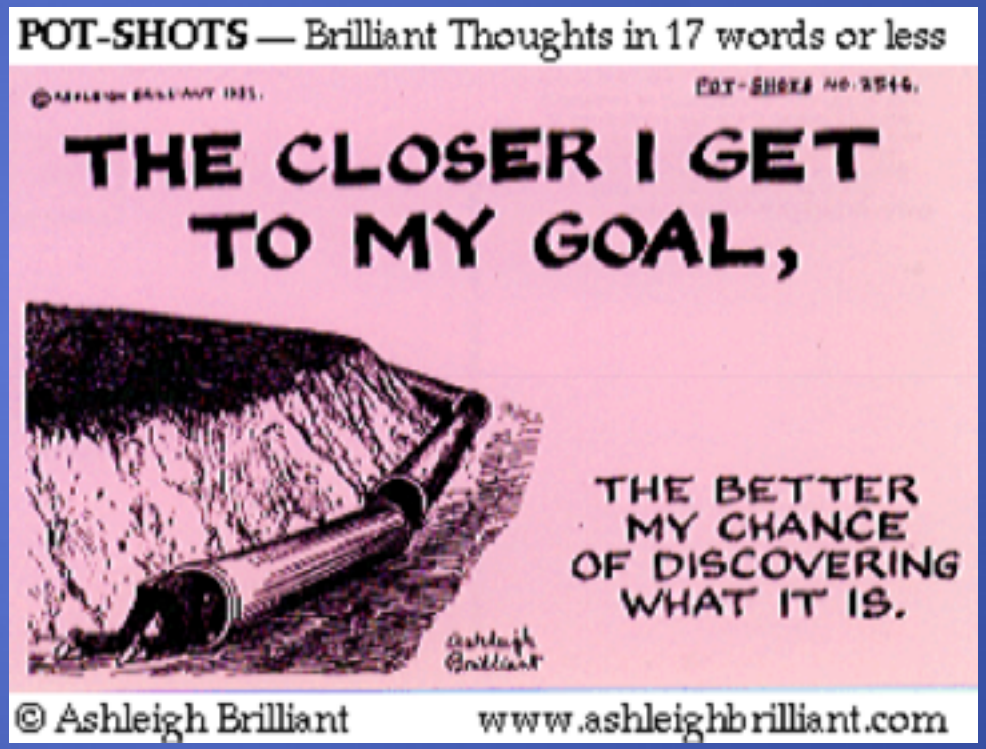
Albert Einstein. 1879-1955

Why do people specify a *Means* as if they were their real *Ends*?

- Means = concrete
- Ends = abstract
- Lack of training/ education
  - Hopper: Puritan Gift



Finding the right *level*



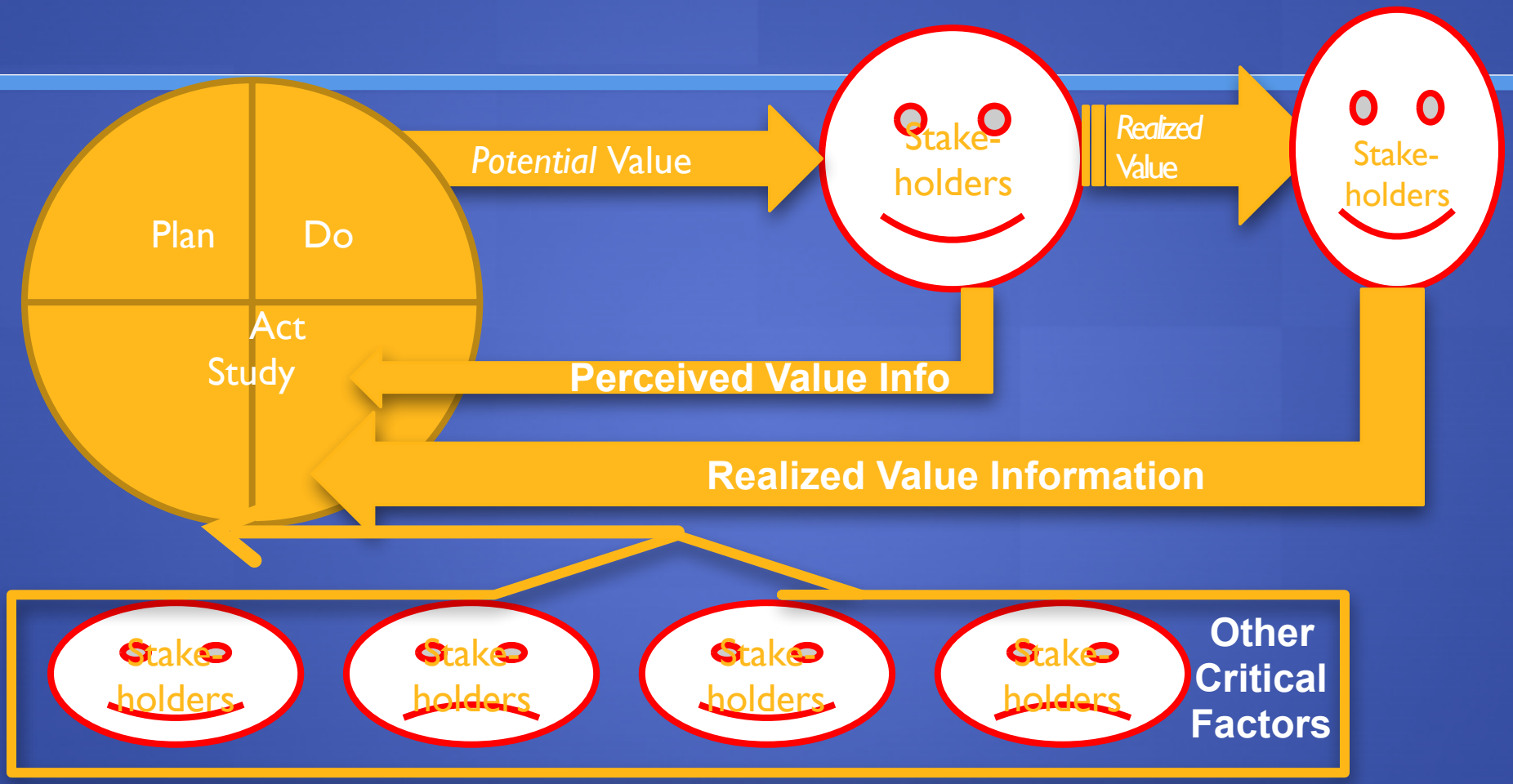
Why? Example


Why do you require a 'password' for Security!
• That's what I asked for!
What kind of security do you want?
• Against stolen information
How strong security against stolen info are you willing to pay for?.
So that is your real requirement ?
• Yep.
Can we make that official?

Our Client, Real Results
Real Immediate Stakeholder value

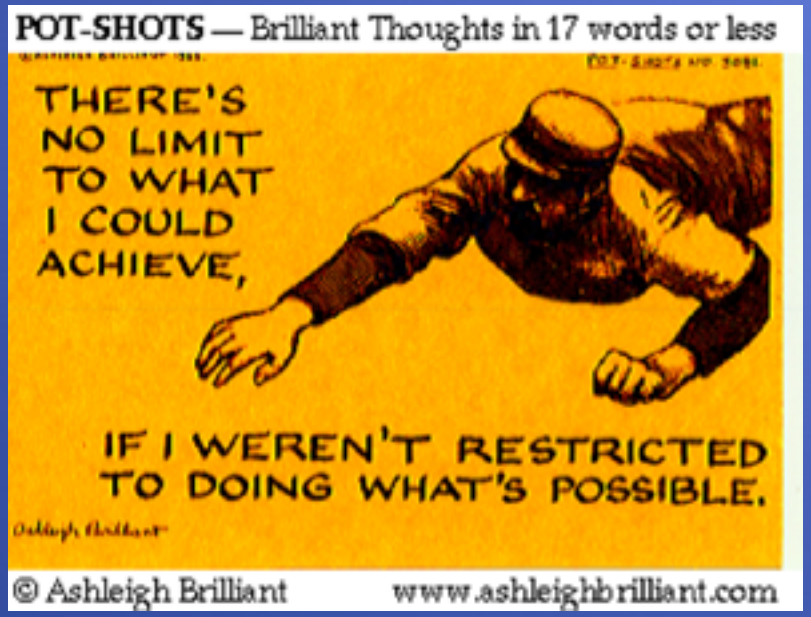
Description of requirement/work task	Past	Status
Usability.Productivity: Time for the system to generate a survey	7200 sec	15 sec
Usability.Productivity: Time to set up a typical specified Market Research-report (MR)	65 min	20 min
Usability.Productivity: Time to grant a set of End-users access to a Report set and distribute report login info.	80 min	5 min
Usability.Intuitiveness: The time in minutes it takes a medium experienced programmer to define a complete and correct data transfer definition with Confirmit Web Services without any user documentation or any other aid	15 min	5 min
Performance.Runtime.Concurrency: Maximum number of simultaneous respondents executing a survey with a click rate of 20 sec and an response time<500 ms, given a defined [Survey-Complexity] and a defined [Server Configuration, Typical]	250 users	6000

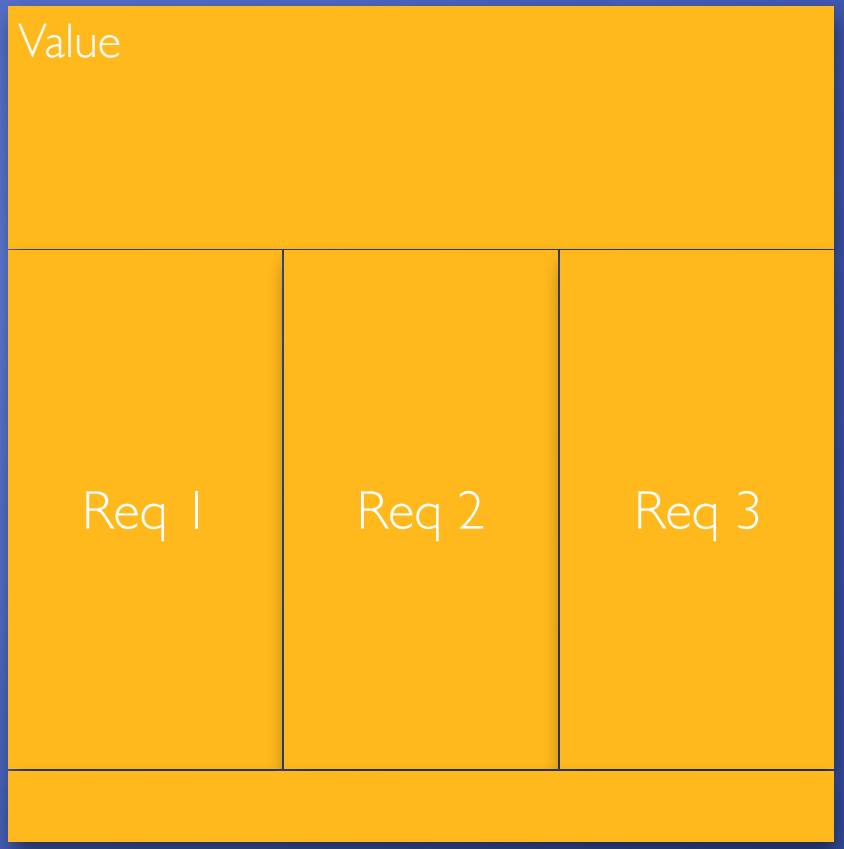
Value Delivery:
leading to *Systems Thinking*, not *Software Silos*



Value Requirements

- If requirements are NOT closely tied to value then:
  - We risk failure to deliver the value expected, even if 'requirements' are satisfied.





How can we articulate and document notions of value
in a requirement specification?
Initial Definitions: to base levels requirement levels on
(this is how the spec looks in 'Planguage')

- Usability.Intuitiveness:
- Type: Marketing Product Requirement.
- Stakeholders: Marketing Director, Support Manager, Training Center
- Impacts: Product Sales, Support Costs, Training Effort, Documentation Design.
- Supports: Corporate Quality Policy 2.3
- Ambition: Any potential user, any age, can immediately discover and correctly use all functions of the product, without training, help from friends, or external documentation
- Scale: % chance that defined [User] can successfully complete defined [Tasks] Immediately, with no External help.
- Meter: [Consumer Reports] tests all tasks for all defined user types, and gives public report.

How can we articulate and document notions of value in a single requirement specification?

Graphic of previous slide: here are some "value relationships"

Ambition: Any potential user, any age, can immediately discover and correctly use all functions of the product, without help from

Usability.Intuitive-ness:						
Type: Marketing Product Requirement.	**Stakeholders: Marketing Director, Support Manager, Training Center**	**Impacts: Product Sales, Support Costs, Training Effort, Documentation Design.**	Supports: Corporate Quality Policy 2.3		**Scale: % chance that defined [User] can successfully complete defined [Tasks] Immediately, with no External help.**	**Meter [Consumer Reports] tests all tasks for all defined user types, and gives public report.**

More Requirement Info? What values are we competing against?

In 'Planguage'

- Analysis
  - Trend [Market = Asia, User = {Teenager, Early Adopters}, Product = Main Competitor, Projection = 2013]
    95%±3% <- Market Analysis
  - Past [Market = USA, User = Seniors, Product = Old Version, Task = Photo Tasks Set, When = 2010]
    70% ±10% <- Our Labs Measures
  - Record [Market = Finland, User = {Android Mobile Phone, Teenagers}, Task = Phone+SMS Task Set, Record Set = January 2010]
    98% ±1% <- Secret Report

Trend

Past

Reco

Graphic of previous slide

"What values are we competing against?" (analysis)

[Market = Asia, User = {Teenager, Early Adopters}, Product = Main Competitor, Projection 2013] 95%

[Market = USA, User = Seniors, Product = Old Version, Task = Photo Tasks Set, When = 2010] 70% ±10% <-

[Mark Finland, User = {Android Mobile Phone, Teenagers}, Task = Phone+SMS Task Set, Record Set = January

Analysis

Goal

Wish

Toler

Requirement-Level Priority Specs

[Market = USA, User = Seniors, Product = New Version, Task = Photo Tasks Set, When = 2012]

[Market = Asia, User = {Teenager, Early Adopters}, Product = Our New Version, Deadline =

[Mark Finland, User = {Android Mobile Phone, Teenagers}, Task = Phone+SMS Task Set,

& who, where] Market = 'Value'

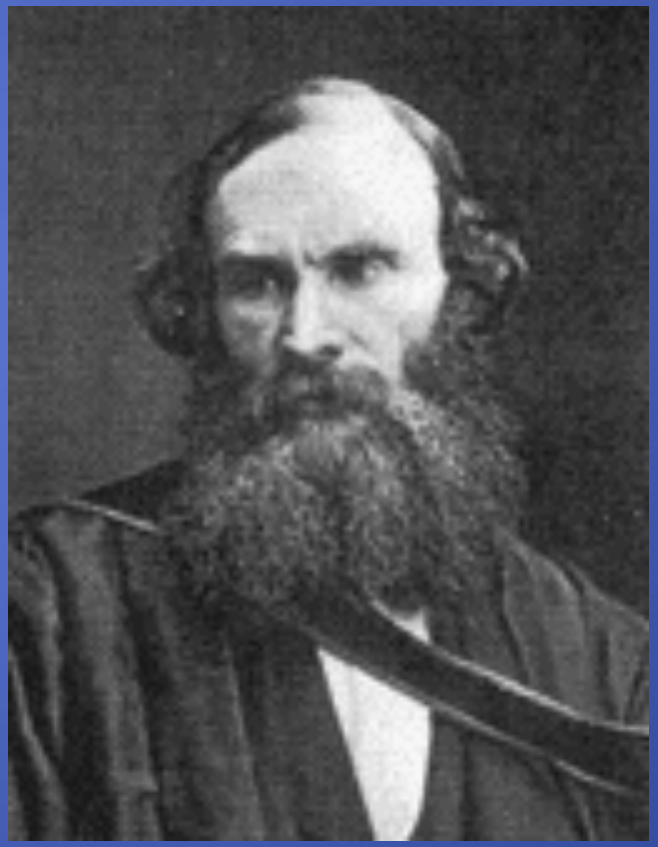
Our Product Plans

Target Levels (Wish, Goal)

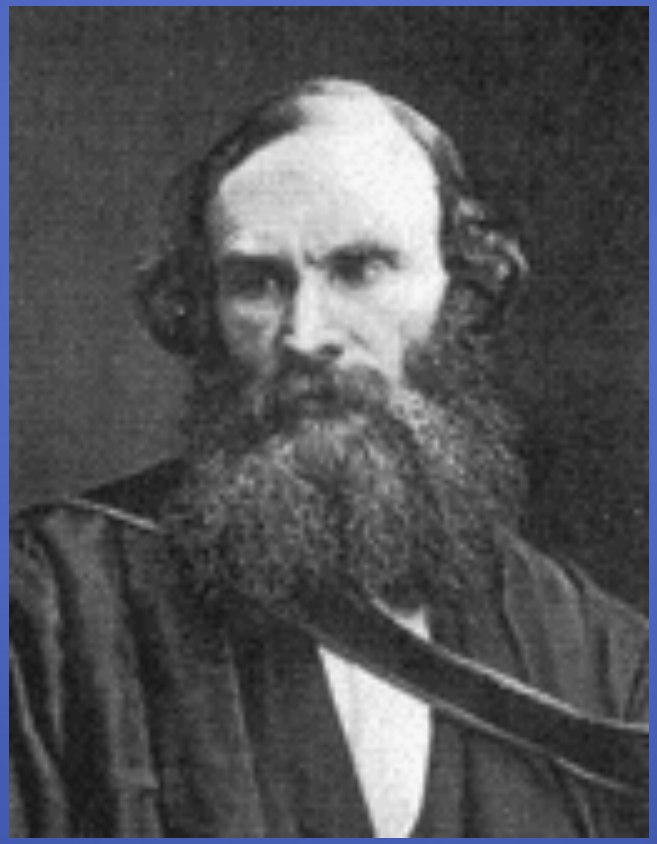
Constraint Level

Quantification: not 'Software Poetry' – a basis for real Software Engineering – not mere 'Softcrafting'

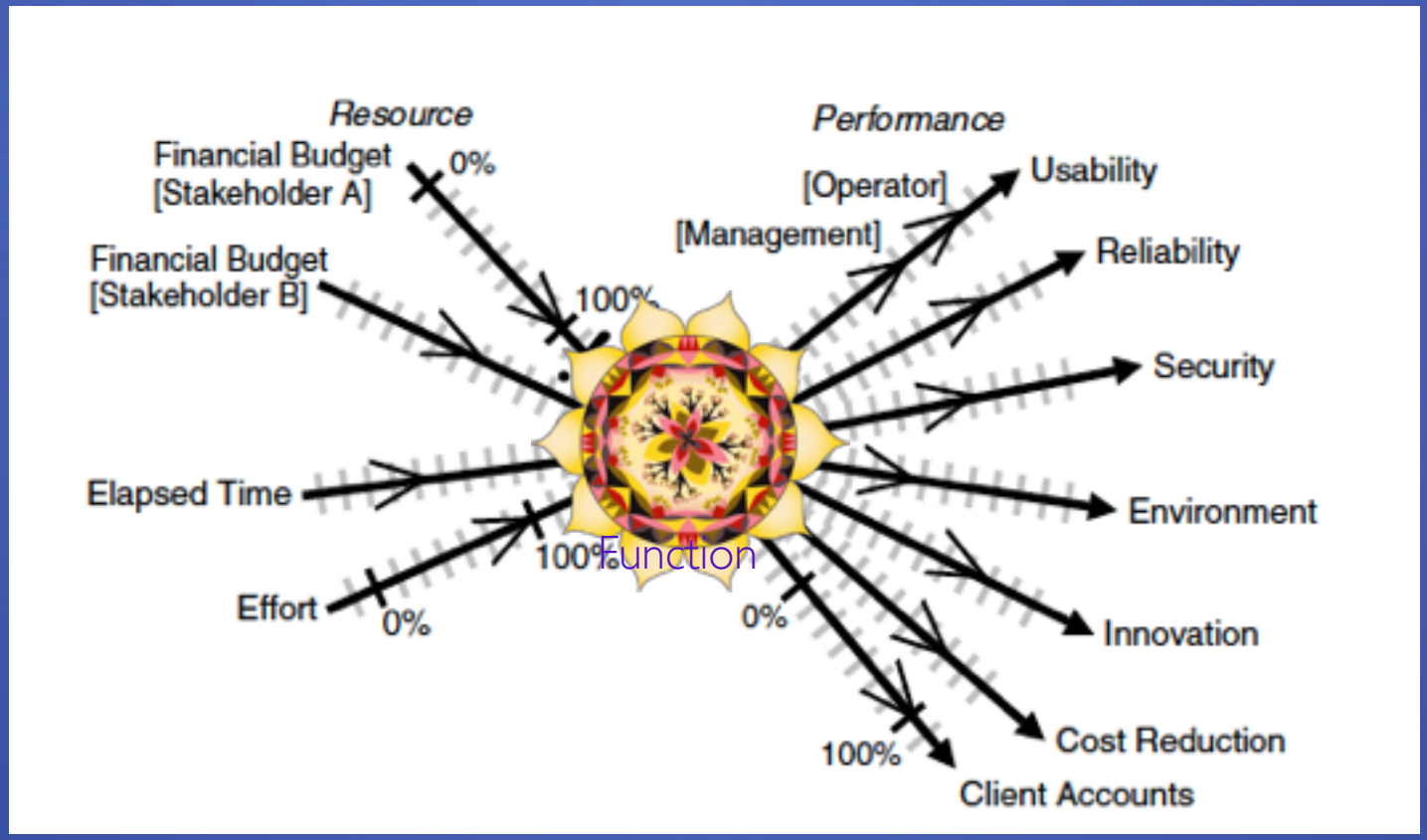
- In physical science the first essential step in the direction of learning any subject is to find principles of numerical reckoning and practicable methods for measuring some quality connected with it.
- I often say that when you can measure what you are speaking about, and express it in numbers, you know something about it;
- but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meagre and unsatisfactory kind;
- it may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the state of *Science*, whatever the matter may be."
- [PLA, vol. 1, "Electrical Units of Measurement", 1883-05-03]
- Lord Kelvin, Sir William Thompson



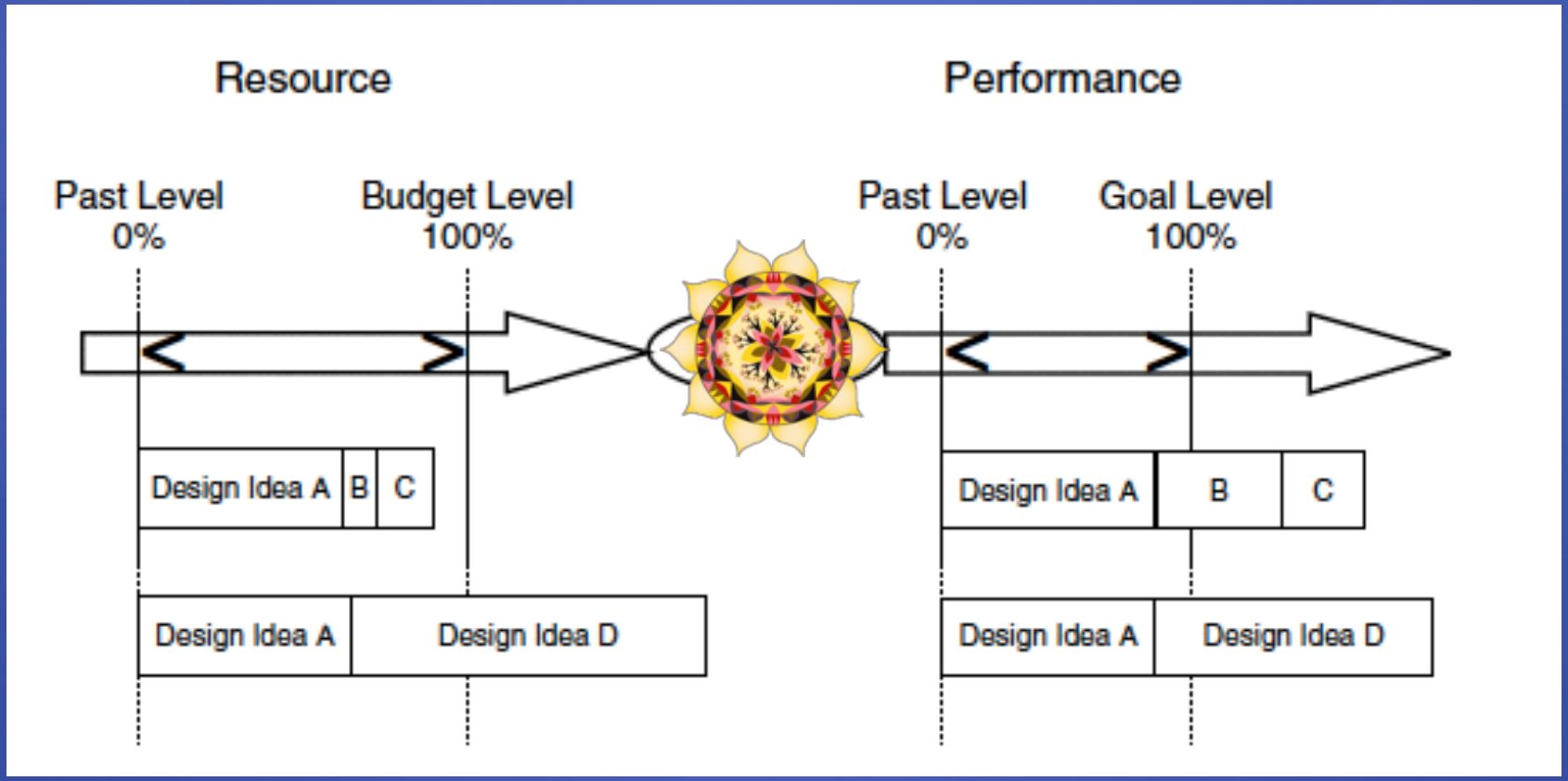
"If you can not measure it, you can not improve it."



Many Qualities and costs



Quantitative Design



Usability.Intuitiveness:

Type: Marketing Product Quality Requirement.

Ambition: Any potential user, any age, can immediately discover and correctly use all functions of the product, without training, help from friends, or external documentation

Scale: % chance that defined [User] can successfully complete defined [Tasks] Immediately, with no External help.

Meter: [Consumer Reports] tests all tasks for all defined user types, and gives public report.

Goal: [Market = USA, User = Seniors, Product = New Version, Task = Photo Tasks Set, When = 2012] 80% ±10% <- Draft Marketing Plan

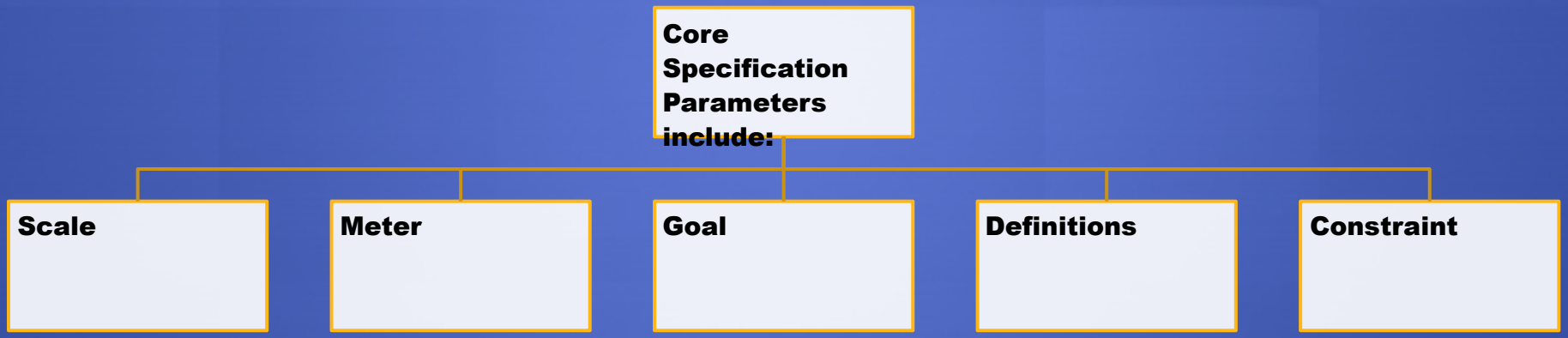
Rich Specification:
Requirement specifications need far more info, than the 'requirement' *itself*!

Rich Specification

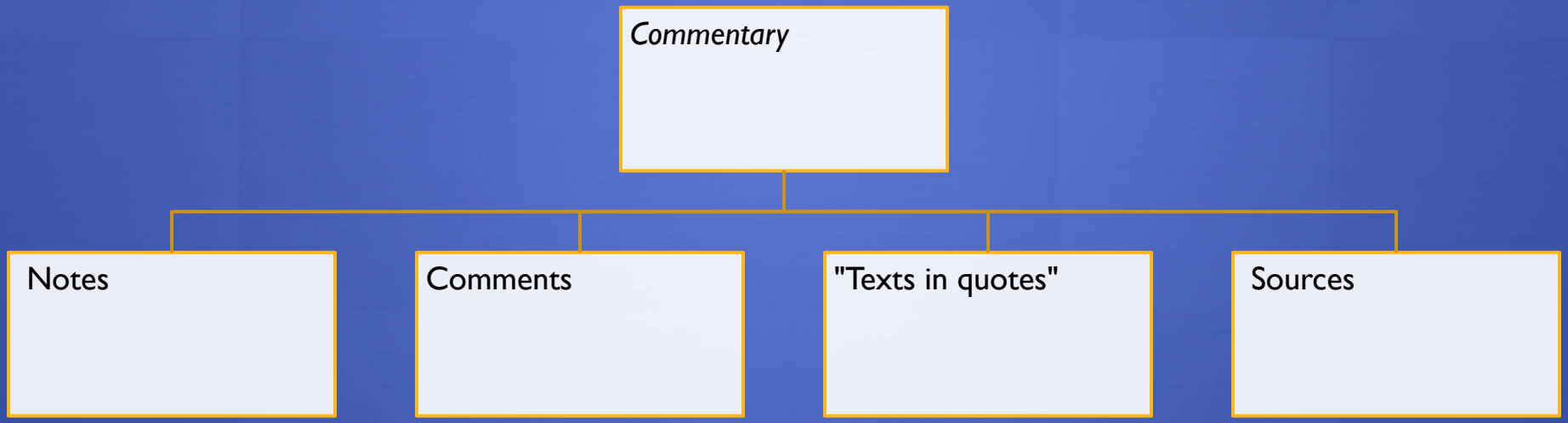
Con

Core

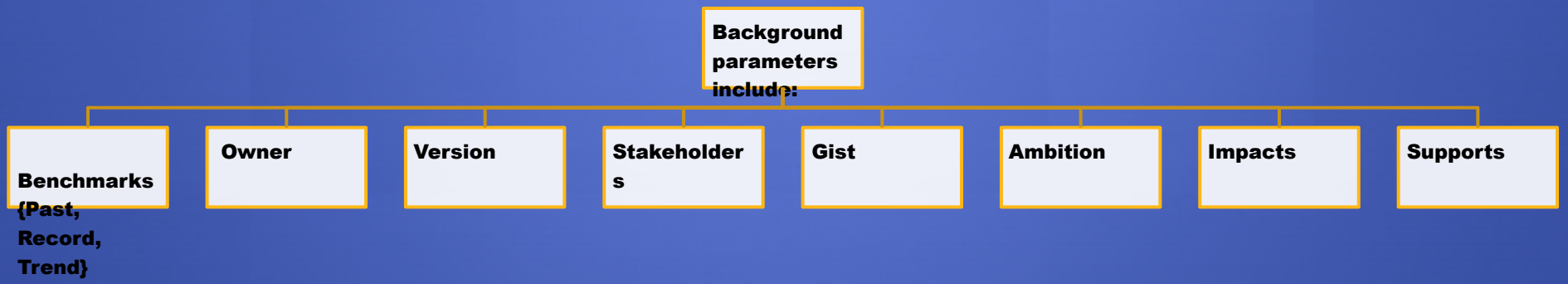
Core Specs


Core Specification Parameters include:
Scale
Meter
Goal
Definitions
Constraint

Commentary Specs



“Background” Specs (often, “Relationships”)



Why do the background specification elements need to be included? Here are some functions of the background information:

1. Value	2. Prioritization	3. Risks
6. Level Synchro	5. Updating	4. Detail Level
7. Quality Control	8. Clarity	9. etc.

Background for Core Specs

Reliability:
Type: Performance.Quality.
Owner: Quality Director
Author: John Engineer
Stakeholders: {Users, Shops, Repair Centers}.
Scale: Mean Time Between Failure.
Goal [Users]: 20,000 hours. <- Customer Survey, 2004
Rationale: anything less would be uncompetitive.
Assumption: our main competitor does not improve more than 10%.
Issues: new competitors might appear.
Risks: the technology for reaching this level might have excessive costs.
Design Suggestion: triple redundant software and database system.
Goal [Shops]: 30,000 hours. <- Dixons' Chain [Quality Director].
Rationale: customer contract specification.
Assumption: this is technically possible today.
Issues: the necessary technology might cause undesired schedule delays.
Risks: the customer might merge with a competitor chain and leave us to foot the costs that they might no longer require.
Design Suggestion: Simplification and reusing known components.

Example: a requirement specification can be embellished with many background specifications that will help us to understand risks associated with one or more other requirement specification elements.

10. Ten Principles for Successful Requirements Methods.

Here is a summary of my advice for more successful requirement methods in the form of some principles, or 'admonishments':

- 1. Quality requirements must be quantified.
- 2. Requirement specifications must be rich with relevant background
- 3. Requirements must be finally developed based on incremental feedback from stakeholders, as to their real value
- 4. Requirements need to be accompanied by many types of signals about their priority, and value
- 5. Requirements must represent the stakeholders' real and core values, not a perceived means of delivering those values

10. Ten Principles for Successful Requirements Methods.

Here is a summary of my advice for more successful requirement methods in the form of some principles, or 'admonishments':

- 6. The top-level most-critical-few project requirements, are the major focus; all others are supporting details
- 7. Requirements are not 'required': they are merely *valued*
- 8. The top ten critical requirements for any project can be quantified and put on a single page.
- 9. A good first draft of the top ten critical requirements for any project can be made in a day's work.
- 10. Requirements will forever change, because our world is changing, so don't ask to get final stable requirements from anyone ever.

The Teachable Details

- Classic Ideas
  - Principles
  - Measures
  - Processes
  - Concept Glossary
- Cases
- Systems Engineering Level
- 60% of book is about Requirements

- Free digital copy?
  - Email Tom@Gilb.com
    - Request "BOOK"
    - And/or request "SLIDES"
    - And/or request "PAPER"

End of SPIN lecture

- The following slides are included to give more realistic detail from recent work we have done in London (2010)
- We have no illusions of presenting them in the SPIN 30 minute time frame , unless a speaker falls away!

Setting and Tracking Project Objectives The Tom Gilb Approach.

Tuesday 7 Dec 20xx
At BCS London
Reused for ACCU 15 April 2011
Included for SPIN 27 June 2011 London
as extra examples

The entire talk, for those who like simple slides

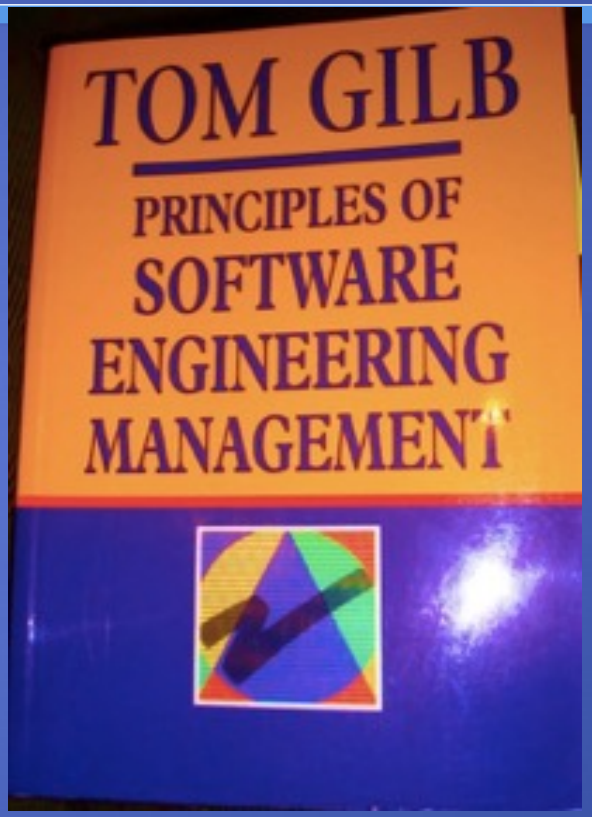
1. Quantify all improvement requirements
2. Estimate quantified impact of all ´means´
3. Do the project in small 2% increments
    1. Highest value for stakeholder first
    2. Measure real value delivered (Goals reached)
    3. Learn from deviations and successes
    4. Modify all requirements and designs as experience and environment dictates

The details

- If you like simplified slides and unfounded generalisations
  - Leave now, or fall asleep, or check messages and news on your phone.
- I personally prefer concrete details, and real examples★
  - So if you choose to stay on, there is going to be a lot of detail
  - In fact – you will not be able to study and get explained all detail
  - But the slides are now at gilb-com/downloads
  - So, if they seem interesting you can study them at your leisure
  - In addition, if you need detailed explanation you will find it in the book Competitive Engineering. If you ask me at tom@gilb.com I´ll be happy to send you a free digital copy.
  - If you are too shy to ask, then copies can be acquired the usual way, and there is plenty of detail free at gilb.com
- Last chance to escape is NOW
- ★ I want to show examples as realistic as possible, but in order to maintain client confidentiality I have:
  - not revealed client names, person names, project names, site location, application names.
  - I have also randomly changed numbers. It is the principles of realistic use I want to share.

The theory and practice of our `Evo´ method for project management

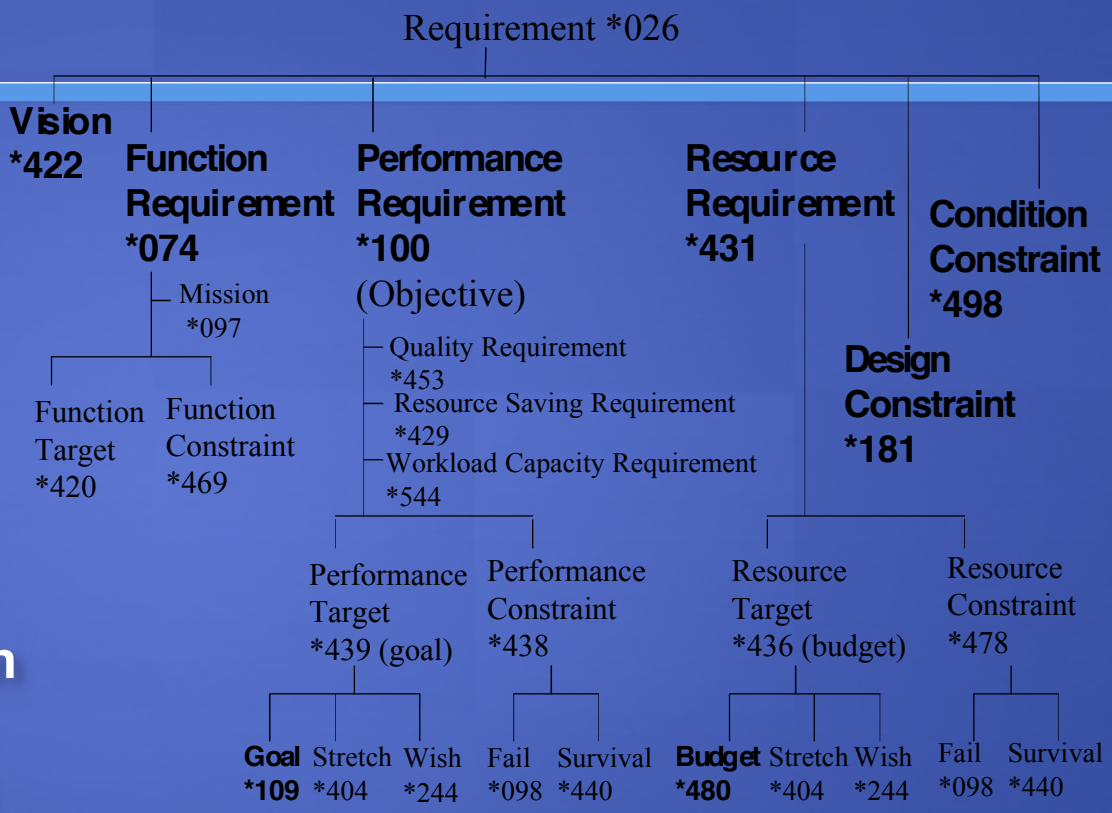
2005



1983

Planguage (Planning Language).

- **A Planning Language - an engineering language**
- **A systems engineering language (software, management)**
- **Concept Glossary**
- **Graphical Language**
- **Control of Multiple dimensions: Performance, Costs, Constraints**
- **Extendible, Tailorable, Open**
- **Rich views, traceability, configuration management**
- **Risk Management**
- **Priority Management**



The Evo method (also known as Value Delivery Method VDM) is a radical simplification (Lean!) from a project management view.

- VALUE CLARITY: Quantify the most-critical project objectives on day 1
- SOLUTION RESPONSIBILITY: Quantify impact of all suggested strategies, architectures, on all critical objectives, deadline, and budget.
- VALUE REPORTING: Measure project progress early, continuously, in terms of top ten objectives
- JUST-IN-TIME PLANNING: Dynamic intelligent do-next prioritisation: Value/ cost based

Original diagram by Kai Gilb

Lack of clear top level project objectives has seen real projects fail for $100+ million: personal experience, real case

Bad Objectives, for 8 years

1. Central to The Corporations business strategy is to be the world's premier integrated_ <domain> service provider.

2. Will provide a much more efficient user experience

3. Dramatically scale back the time frequently needed after the last data is acquired to time align, depth correct, splice, merge, recompute and/or do whatever else is needed to generate the desired products

4. Make the system much easier to understand and use than has been the case for previous system.

5. A primary goal is to provide a much more productive system development environment than was previously the case.

6. Will provide a richer set of functionality for supporting next-generation logging tools and applications.

7. Robustness is an essential system requirement (see partial rewrite in example at right)

8. Major improvements in data quality over current practice

Quantified Objectives (in Planguage),

Robustness.Testability:

Type: Software Quality Requirement.
Version: 20 Oct 2006-10-20
Status: Demo draft,
Stakeholder: {Operator, Tester}.
Ambition: Rapid-duration automatic testing of <critical complex tests>, with extreme operator setup and initiation.

Scale: the duration of a defined [Volume] of testing, or a defined [Type], by a defined [Skill Level] of system operator, under defined [Operating Conditions].

Goal [All Customer Use, Volume = 1,000,000 data items, Type = WireXXXX Vs DXX, Skill = First Time Novice, Operating Conditions = Field, {Sea Or Desert}. <10 mins.

VALUE CLARITY:
Quantify the most-critical project objectives on day 1

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 %

Example of Estimating the Value of a Technical IT System Improvement (20xx)

TIME.HEDGE - Time for hedge execution of average-sized trade	
Ambition:	Reduce the average time taken from verbal agreement ("done") to hedge execution of an <average-sized> trade
Scale:	Seconds
Past:	[2Q10; Region=NA] 30 seconds
Goal:	[2Q12; Region=ALL] 3 seconds
Business Value:	**[Type=Revenue; Reason=Improved Hedging P&L; Goal Scale=3 seconds; Region=Global] Revenue= +$1mm to +$2mm**

SPEED.CODE – Mean elapsed time for code changes	
Ambition:	Reduce the mean elapsed time for code changes from business request to end-user go live
Scale:	Mean time in calendar days over <three> months
Past:	[2009; Market=Eurex; Task=Bond execution] <60 - 90> days
Goal:	[2Q12; Market=Eurex; Task=Bond execution] 5 days
Business Value:	**[Type=Revenue; Reason=Earlier P&L from faster time to Market; Goal Scale=5 days; Region=Global] Revenue= +$2mm to +$5mm**

This is an example made to reason about specification standards and is not supposed to be a real spec. Just realistic.

Quantified Objective in Planguage Tool: notice Stakeholders

[redacted]

Timeliness

Version *1.12.*
Type: *Top Level Business Goal*
Quality
Owner: *Sam, Andy*
Stakeho **Primary:** Front Office, Middle Office; **Secondary:** Senior Management, Product Control, Financial Control, Internal Audit
Ambitio *Consistently meet timeliness SLAs for the daily business process. E. g. Availability of SOD risk*

Scale: **average number of days per year that defined [SLA] is exceeded, due to the [System], for defined [Scope]**

		Day & Time	*Conditions (Place, Defined, Stakeholder, etc.)*	]	*number*	
Past	[at			]		±
Status	[at		Sum	]	**0**	±
Tolerable	[by	2014	Sum	]	**3**	±
Goal	[by	2014 - j...	Sum	]	**100**	±
Past	[		SLA=SOD risk by 7.30am, Scope=Exxxx Exxxxx, System=OXXXX	]	6	±
Status	[		SLA=SOD risk by 7.30am, Scope=Exxxx Exxxxx, System=OXXXX	]	**6**	±
Tolerable	[	2014 - j...	SLA=SOD risk by 7.30am, Scope=Exxxx Exxxxx, System=TBD	]	**2**	±
Goal	[	2014 - j...	SLA=SOD risk by 7.30am, Scope=Exxxx Exxxxx, System=TBD	]	**0**	±
	[			]		±
Past			SLA=Initial EOD P/L within 5 mins of being avail. in Kxxxx, Scope=Exxxx Flo		252	
Status			SLA=Initial EOD P/L within 5 mins of being avail. in Kxxxx, Scope=Exxxx Flo		**252**	
Tolerable		2014 - j...	SLA=Initial EOD P/L within 5 mins of being avail. in Kxxxx, Scope=Exxxx Flo		**15**	
Goal		2014 - j...	SLA=Initial EOD P/L within 5 mins of being avail. in Kxxxx, Scope=Exxxx Flo		**0**	
Past			SLA=SOD risk by 7am, Scope=Exxxx Flow Options, System=Txxxx		1	
Status			SLA=SOD risk by 7am, Scope=Exxxx Flow Options, System=Txxxx		1	
Tolerable		2014 - j...	SLA=SOD risk by 7am, Scope=Exxxx Flow Options, System=TBD		1	
Goal		2014 - j...	SLA=SOD risk by 7am, Scope=Exxxx Flow Options, System=TBD		0	

SOLUTION RESPONSIBILITY:
Quantify impact of all suggested strategies, architectures, on all critical objectives, deadline, and budget.

NOT ☹

- Just *name* an idea/design
- Assert the design is good
- Fail to explain how you know
- Fail to take responsibility
- Fail to measure results
- Fail to consider all requirements
- Fail to even estimate costs
- ***"Tool Simulators, Reverse Cracking Tool, Generation of simulated telemetry frames entirely in software, Application specific sophistication, for <our domain>– recorded mode simulation by playing back the dump file, Application test harness console" <-6.2.1 HFA***

YES ! ☺

- Describe detail for estimation
- Estimate the impact on Goals
- Estimate the ± uncertainty
- Specify the estimate evidence
- Estimate all objectives
- Estimate all resources

Don´t we need more detail to estimate costs and other attributes of a design?

Simple design description

- Design Spec:
  - Risk and P/L aggregation service

Ask the following questions about such brief design descriptions

- What will it cost to develop?
- What will it cost to operate?
- Will we deliver any or all of the quality and performance Goal levels on time?
- What are the critical assumptions, that might fail or be untrue?
- What are the known risks?
- Do we actually understand anything of consequence from such a short design specification?

See enlarged view of this slide in following slides. This is a 1-page overview

Defining a Design/Solution/Architecture/Strategy (Planguage, CE Design Template)
1. enough detail to estimate, 2. some impact assertion, 3. Assumptions, Risks, Issues

Orbit Application Base: (formal Cross reference Tag)

Type: Primary Architecture Option

========== Basic Information ==========

Version: Nov. 30 20xx 16:49, updated 2.Dec by telephone and in meeting. 14:34

Status: Draft

Owner: Brent Barclays

Expert: Raj Shell, London

Authority: for differentiating business environment characteristics, Raj Shell, Brent Barclays(for overview)

Source: <Source references for the information in this specification. Could include people>. Various, can be done later BB

Gist: risk and P/L aggregation service, which also provides work flow/adjustment and outbound and inbound feed support. Currently used by Rates ExtraBusiness, Front Office and Middle Office, USA & UK.

Description: <Describe the design idea in sufficient detail to support the estimated impacts and costs given below>.

D1: ETL Layer. Rules based highly configurable implementation of the ETL Pattern, which allows the data to be onboarded more quickly. Load and persist new data very quickly. With minimal development required. -> Business-Capability-Time-To-Market, Business Scalability

D2: high performance risk and P/L aggregation processing (Cube Building). -> Timeliness, P/L Explanation, Risk & P/L Understanding, Decision Support, Business Scalability, Responsiveness.

D3: Orbit supports BOTH Risk and P/L -> P/L Explanation, Risk & P/L Consistency, Risk & P/L Understanding, Decision Support.

D4: a flexible configurable workflow tool, which can be used to easily define new workflow processes -> Books/Records Consistency, Business Process Effectiveness, Business Capability Time to Market.

D5: a report definition language, which provides 90+% of the business logic contained with Orbit, allows a quick turnaround of new and enhanced reports with minimal regression testing and release procedure impact. -> P/L Explanation, Risk & P/L Understanding, Business Capability Time to Market, Business Scalability.

D6: Orbit GUI. Utilizes an Outlook Explorer metaphor for ease of use, and the Dxx Express Grid Control, to provide high performance Cube Interrogation Capability. -> Responsiveness, People Interchangeability, Decision Support, Risk & P/L Understanding.

D7: downstream feeds. A configurable event-driven data export service, which is used to generate feeds . -> Business Process Effectiveness, Business Capability Time to Market.

========== Priority and Risk Management ==========

Assumptions: <Any assumptions that have been made>.

A1: FCCP is assumed to be a part of Orbit. FCxx does not currently exist and is Dec 20xx 6 months into Requirements Spec. <- Picked up by TsG from dec 2 discussions AH MA JH EC.

Consequence: FCxx must be a part of the impact estimation and costs rating.

A2: Costs, the development costs will not be different. All will base on a budget of say $nn mm and 3 years. The o+

costs may differ slightly, like $n mm for hardware. MA AH 3 dec

A3:Boss X will continue to own Orbit. TSG DEC 2

A4: the schedule, 3 years, will constrained to a scope we can in fact deliver, OR we will be given additional budget. If not "I would have a problem" <- BB

A5: the cost of expanding Orbit will not be prohibitive. <- BB 2 dec

A6: we have made the assumption that we can integrate Oribit with PX+ in a sensible way, even in the short term <- BB

Dependencies: <State any dependencies for this design idea>.

D1: FCxx replaces Px+ in time. ? tsg 2.12

Risks: <Name or refer to tags of any factors, which could threaten your estimated impacts>.

R1. FCxx is delayed. Mitigation: continue to use Pxx <- tsg 2.12

R2: the technical integration of Px+ is not as easy as thought & we must redevelop Oribit

R3: the and or scalability and cost of coherence will not allow us to meet the delivery.

R4: scalability of Orbit team and infrastructure, first year especially <- BB. People, environments, etc.

R5: re Cross Desk reporting Requirement, major impact on technical design. Solution not currently known. Risk no solution allowing us to report all P/L

Issues: <Unresolved concerns or problems in the specification or the system>.

I1: Do we need to put the fact that we own Orbit into the objectives (Ownership). MA said, other agreed this is a huge differentiator. Dec 2.

I2: what are the time scales and scope now? Unclear now BB

I3: what will the success factors be? We don't know what we are actually being asked to do. BB 2 dec 20xx

I4: for the business other than flow options, there is still a lack of clarity as to what the requirements are and how they might differ from Extra and Flow Options. BB

I5: the degree to which this option will be seen to be useful without Intra Day. BB 2 dec

Spec Headers

Orbit Application Base: (formal Cross reference Tag)

Type: Primary Architecture Option

==== Basic Information ==========

Version: Nov. 30 20xx 16:49, updated 2.Dec by telephone and in meeting. 14:34

Status: Draft (PUBLIC EXAMPLE EDIT)

Owner: Brent Barclays

Expert: Raj Shell, London

Authority: for differentiating business environment characteristics, Raj Shell, Brent Barclays(for overview)

Source: <Source references for the information in this specification. Could include people>. Various, can be done later BB

Gist: risk and P/L aggregation service, which also provides work flow/ adjustment and outbound and inbound feed support. Currently used by Rates Extra Business, Front Office and Middle Office, USA & UK.

Detailed Description and -> Impacted Objectives

Description: <Describe the design idea in sufficient detail to support the estimated impacts and costs given below>.

D1: ETL Layer. Rules based highly configurable implementation of the ETL Pattern, which allows the data to be onboarded more quickly. Load and persist new data very quickly. With minimal development required. -> Business-Capability-Time-To-Market, Business Scalability

D2: high performance risk and P/L aggregation processing (Cube Building). -> Timeliness, P/L Explanation, Risk & P/L Understanding, Decision Support, Business Scalability, Responsiveness.

D3: Orbit supports BOTH Risk and P/L -> P/L Explanation, Risk & P/L Consistency, Risk & P/L Understanding, Decision Support.

D4: a flexible configurable workflow tool, which can be used to easily define new workflow processes -> Books/Records Consistency, Business Process Effectiveness, Business Capability Time to Market.

D5: a report definition language, which provides 90+% of the business logic contained with Orbit, allows a quick turnaround of new and enhanced reports with minimal regression testing and release procedure impact. -> P/L Explanation, Risk & P/L Understanding, Business Capability Time to Market, Business Scalability.

D6: Orbit GUI. Utilizes an Outlook Explorer metaphor for ease of use, and the Dxx Express Grid Control, to provide high performance Cube Interrogation Capability. -> Responsiveness, People Interchangeability, Decision Support, Risk & P/L Understanding.

D7: downstream feeds. A configurable event-driven data export service, which is used to generate feeds . -> Business Process Effectiveness, Business Capability Time to Market.

Design Spec Enlarged 2 of 2

==== Priority & Risk Management ========

Assumptions: <*Any assumptions that have been made*>.

A1: FCCP is assumed to be a part of Orbit. FCxx does not currently exist and is Dec 20xx 6 months into Requirements Spec. <- Picked up by TsG from dec 2 discussions AH MA JH EC.

Consequence: FCxx must be a part of the impact estimation and costs rating.

A2: Costs, the development costs will not be different. All will base on a budget of say $ nn mm and 3 years. The ops costs may differ slightly, like $n mm for hardware. MA AH 3 dec

A3:Boss X will continue to own Orbit. TSG DEC 2

A4: the schedule, 3 years, will constrained to a scope we can in fact deliver, OR we will be given additional budget. If not "I would have a problem" <- BB

A5: the cost of expanding Orbit will not be prohibitive. <- BB 2 dec

A6: we have made the assumption that we can integrate Oribit with PX+ in a sensible way, even in the short term <- BB

Dependencies: <*State any dependencies for this design idea*>.

D1: FCxx replaces Px+ in time. ? tsg 2.12

Risks: <*Name or refer to tags of any factors, which could threaten your estimated impacts*>.

R1. FCxx is delayed. Mitigation: continue to use Pxx<- tsg 2.12

R2: the technical integration of Px+ is not as easy as thought & we must redevelop Oribit

R3: the and or scalability and cost of coherence will not allow us to meet the delivery.

R4: scalability of Orbit team and infrastructure, first year especially <- BB. People, environments, etc.

R5: re Cross Desk reporting Requirement, major impact on technical design. Solution not currently known. Risk no solution allowing us to report all P/L

Issues: <*Unresolved concerns or problems in the specification or the system*>.

I1: Do we need to put the fact that we own Orbit into the objectives (Ownership). MA said, other agreed this is a huge differentiator. Dec 2.

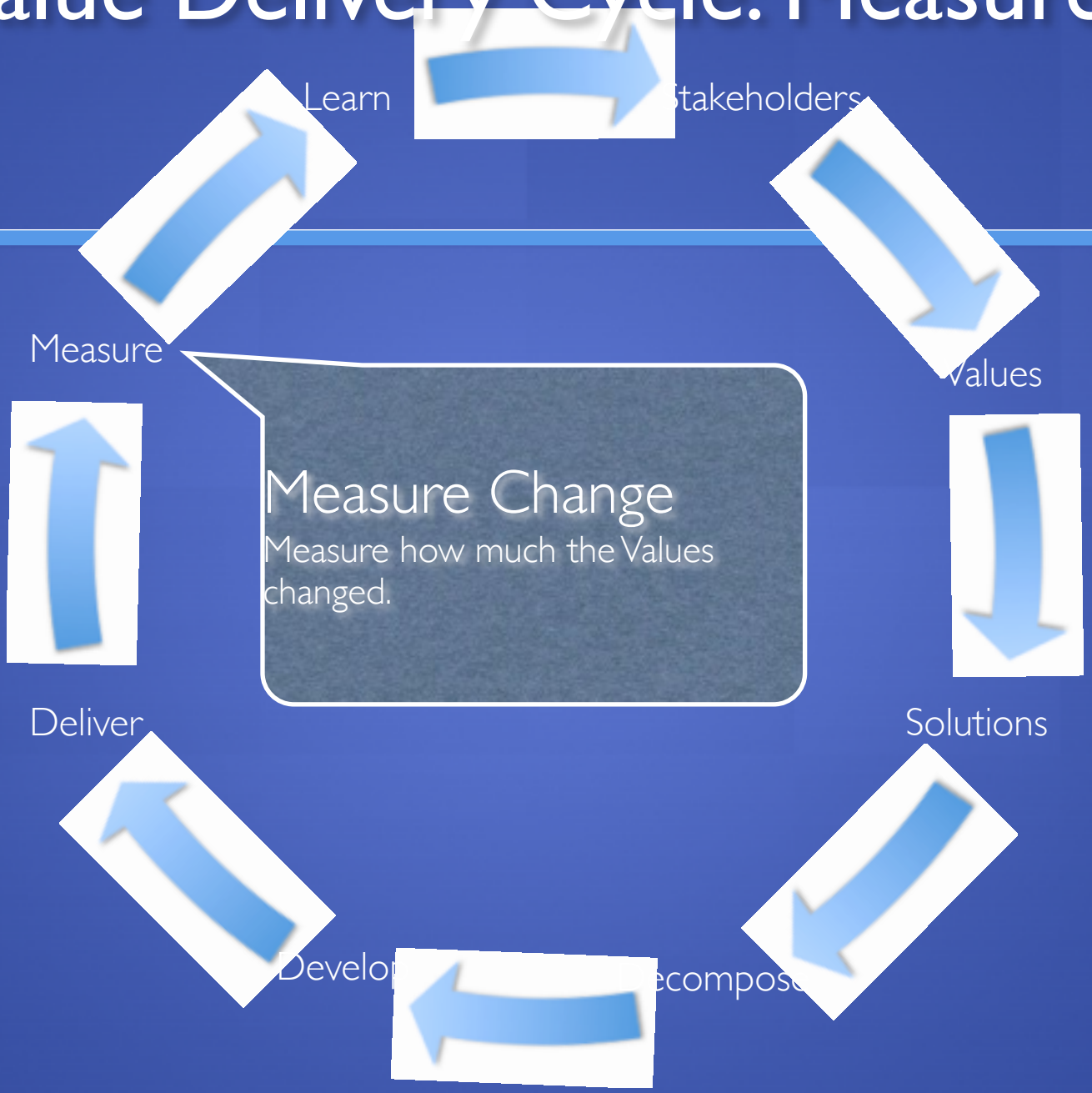
I2: what are the time scales and scope now? Unclear now BB

I3: what will the success factors be? We don't know what we are actually being asked to do. BB 2 dec 20xx

I4: for the business other than flow options, there is still a lack of clarity as to what the requirements are and how they might differ from Extra and Flow Options. BB

I5: the degree to which this option will be seen to be useful without Intra Day. BB 2 dec

Value Delivery Cycle: Measure



The impact of a solution,
on a single improvement objective

Solution ABC

20 seconds

Function

-

Attribute

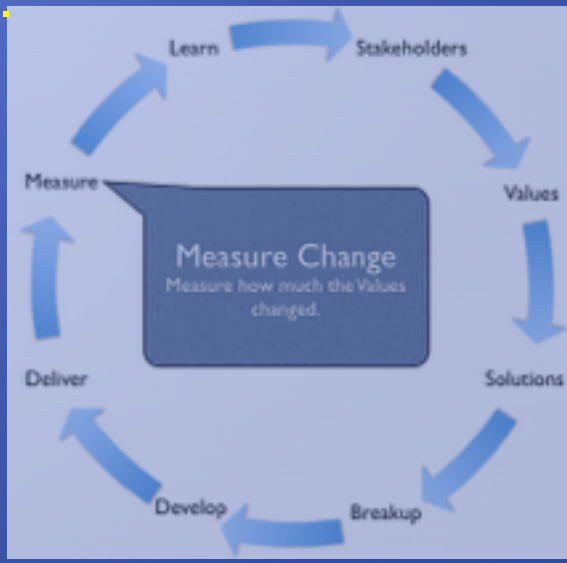
+

?

Past
[Dec. 2010]
50 sec.

Tolerable
[April 2015]
40 sec.

Goal
[April 2016]
15 sec.



Impact Estimation Tables

Improvement

Value Requirements Status when	Tolerable when	Goal when	Operating Model Consistency units	% of Goal
P&L-Consistency&T P&L			-20	**44%**
60	0	15	-10	22%
0	0	0	0.1	4%
Speed-To-Deliver			-20	**29%**
75	30	5	-7	10%
0	0	0	0.1	3%
Operational-Control.Accurate			5	**50%**
90	99	100	5	50%
0	0	0	0.1	5%
Operational-Control.Consistent			1	**50%**
97	0	99	0.2	10%
0	0	0	0.2	10%
Operational-Control.Timely.End&Overnigh			-1	**200%**
1	1	0.5	-0.5	100%
0	0	0	0.2	40%
Operational-Control.Timely.IntradayP&L				
1	2	3		
0	0	0		
Operational-Control.Timely.Trade-Booking			-15	**75%**

Estimate Units & %

± Uncertainty Worst Case range

Credibility Adjustment 0.0 to 1.0

Based on tool built by Kai Gilb

Impact Estimation Concepts

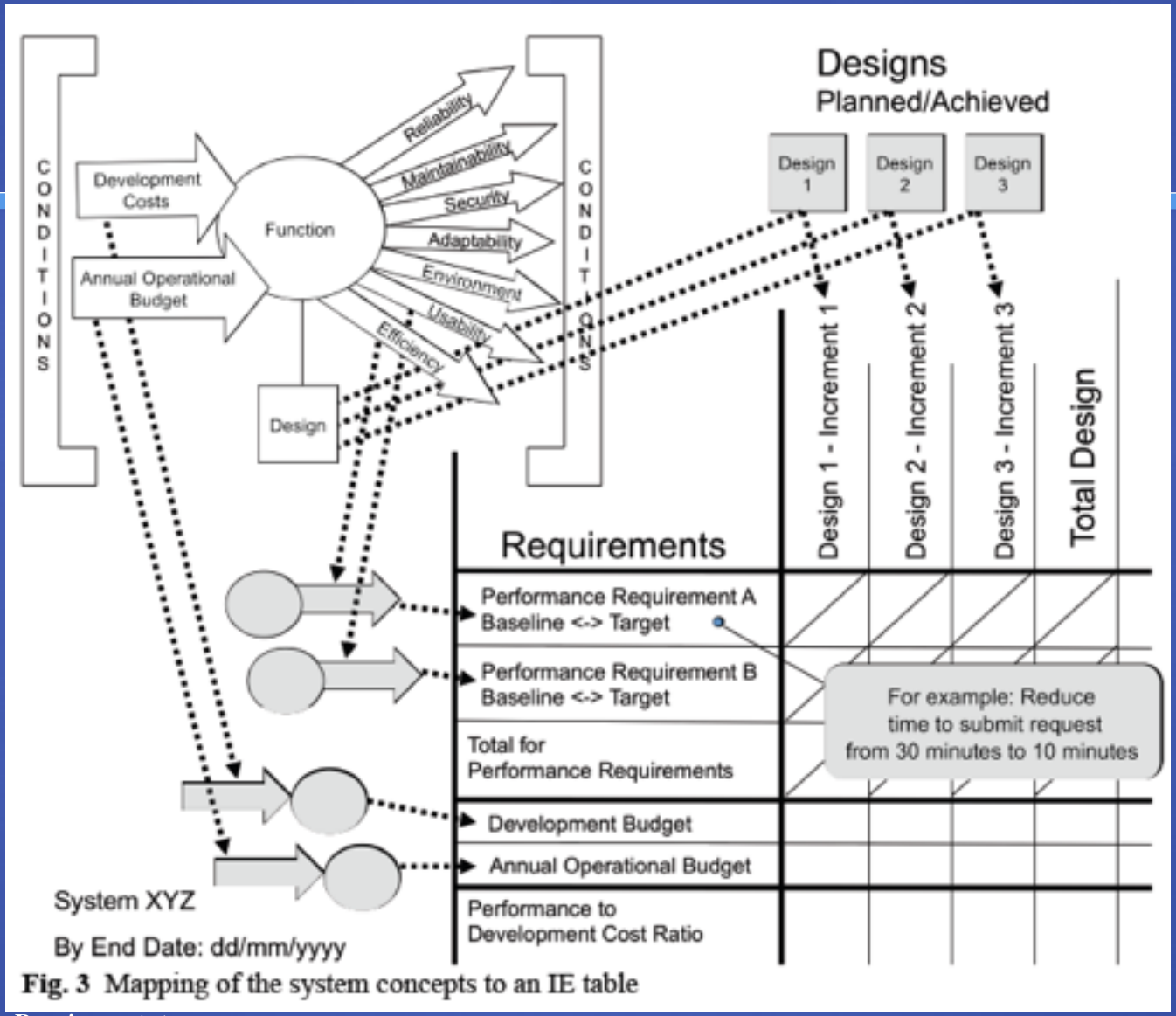


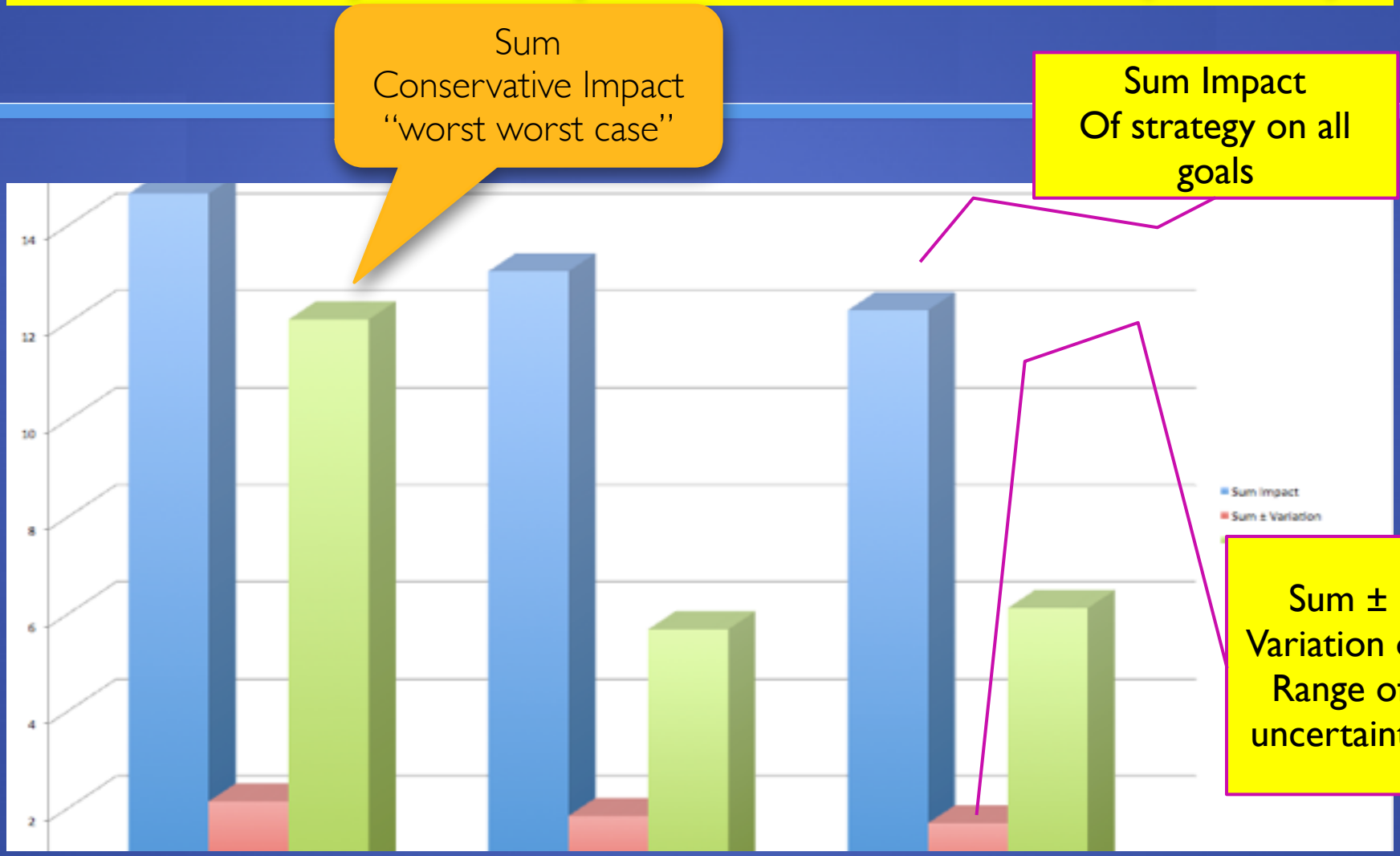
Fig. 3 Mapping of the system concepts to an IE table

SOURCE

Using Metrics within System Requirements to Express Quality and Derive Stakeholder Value

Lindsey Brodie

Summary of Options wrt Risk (2010)



Based on work done by Kai Gilb

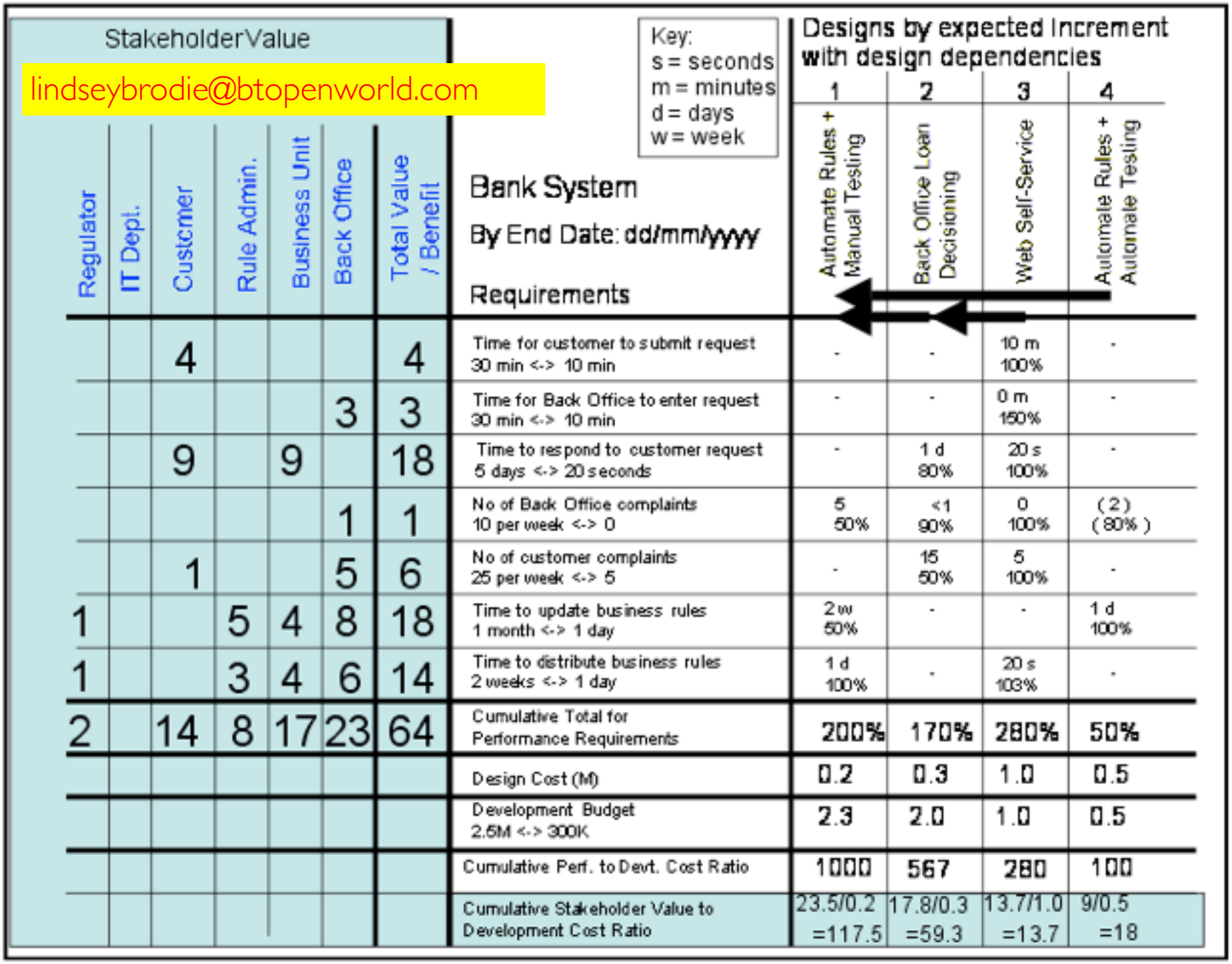
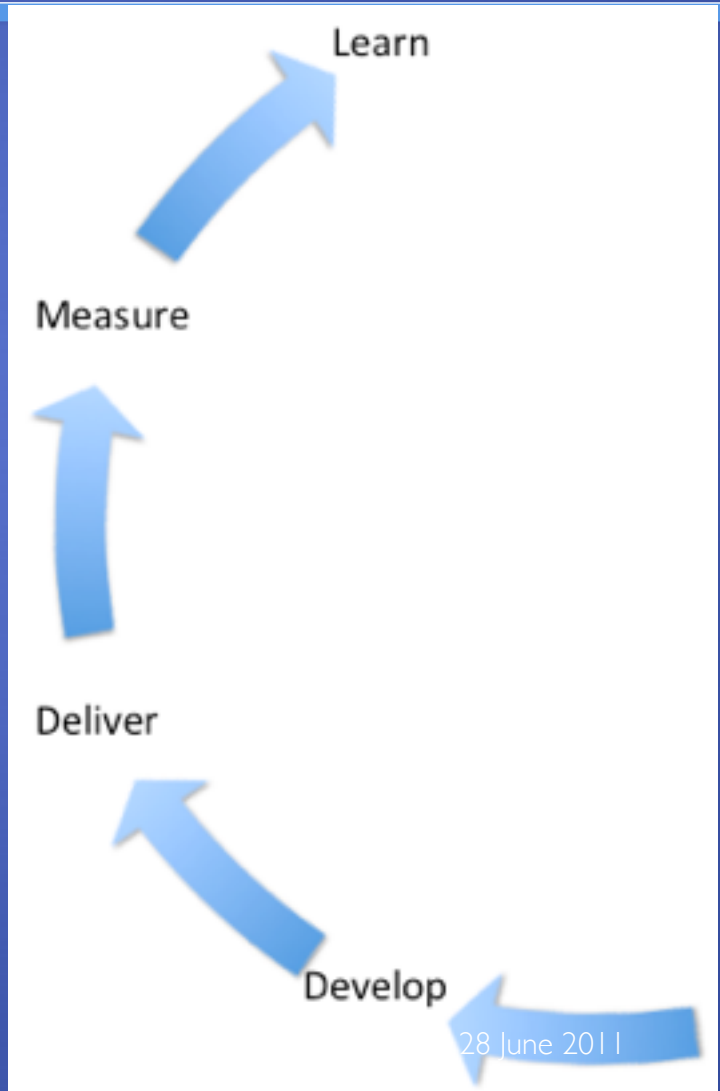


Figure 4: An IE table for the bank system. The shaded area represents the extensions to IE

VALUE REPORTING:
Measure project progress early, continuously, in terms of top ten objectives

- Basic idea
  - Estimate expected value next cycle
    - Based on a specific design for that increment
    - Design Hypothesis
  - Measure the actual effect, roughly, pilot,
    - Confirm or deny the effect hypothesis
  - If reasonable result compared to need and expectation, then take another cumulative cycle
  - Measure the cumulated value later, and better, before scaling up and major release
  - If bad result: learn change, try again



Real client example (Confirmit): weekly design impact estimates, and same-week measurement,
Weekly Feedback to the development team
about cumulative progress toward critical numeric performance and quality targets

	A	B	C	D	E	F	G	BX	BY	BZ	CA
1											
2		Current Status	Improvements		Goals			Step9			
3								Recoding			
4								Estimated impact		Actual impact	
5		Units	Units	%	Past	Tolerable	Goal	[illegible]	%	Units	
6					Usability.Replacability (feature count)						
7		1,00	1,0	50,0	2	1	0				
8					Usability.Speed.NewFeaturesImpact (%)						
9		5,00	5,0	100,0	0	15	5				
10		10,00	10,0	200,0	0	15	5				
11		0,00	0,0	0,0	0	30	10				
12					Usability.Intuitiveness (%)						
13		0,00	0,0	0,0	0	60	80				
14					Usability.Productivity (minutes)						
19		20,00	45,0	112,5	65	35	25	20,00	50,00	38,00	95,00
20					Development resources						
21			101,0	91,8	0		110	4,00	3,64	4,00	3,64

Weekly Testing

Estimates

Priority Next week Warning metrics based

Cumulative weekly progress metric

Constraint

Target

Evo Plan Confirmit 8.5

4 product areas were attacked in all: 25 Qualities concurrently, one quarter of a year. Total development staff = 13

Sum impact after 9 deivery cycles

Impact Estimation Table: Reportal codename "Hyggen"

Current Status	Improvements		Reportal - E-SAT features		
Units	Units	%	Past	Tolerable	Goal
			Usability.Intuitivness (%)		
75,0	25,0	62,5		75	90
			Usability.Consistency.Visual (Elements)		
14,0	14,0	100,0	0	11	14
			Usability.Consistency.Interaction (Components)		
15,0	15,0	107,1	0	11	14
			Usability.Productivity (minutes)		
5,0	75,0	96,2		5	2
5,0	45,0	95,7		5	1
			Usability.Flexibility.OfflineReport.ExportFormats		
3,0	2,0	66,7		3	4
			Usability.Robustness (errors)		
1,0	22,0	95,7		1	0
			Usability.Replacability (nr of features)		
4,0	5,0	100,0		5	3
			Usability.ResponseTime.ExportRep (min		es
1,0	12,0	150,0		13	5
			Usability.ResponseTime.ViewRepo seco		s)
1,0	14,0	100,0	15		1
			evelopment resources		
203,0					91

Current Status	Improvements		Survey Engine .NET		
Units	Units	%	Past	Tolerable	Goal
			Backwards.Compatibility (%)		
83,0	48,0	80,0	40	85	95
0,0	67,0	100,0	67	0	0
			Generate.WI.Time (small/medium/large seconds)		
4,0	59,0	100,0	63	8	4
10,0	397,0	100,0	407	100	10
94,0	2290,0	103,9	2384	500	180
			Testability (%)		
10,0	10,0	13,3	0	100	100
			Usability.Speed (seconds/user rating 1-10)		
774,0	507,0	51,7	1281	600	300
5,0	3,0	60,0	2	5	7
			Runtime.ResourceUsage.Memory		
0,0	0,0	0,0		?	?
			Runtime.ResourceUsage.CPU		
3,0	35	97,2	38	3	2
			Runtime.ResourceUsage.MemoryLeak		
0	800	100,0	800	0	0
			Runtime.Concurrency (number of users)		
350	1100	146,7	150	500	1000
			Development resources		
64			0		84

Current Status	Improvements		Reportal - MR Features		
Units	Units	%	Past	Tolerable	Goal
			Usability.Replacability (feature count)		
1,0	1,0	50,0	14	13	12
			Usability.Productivity (minutes)		
20,0	45,0	112,5	65	35	25
			Usability.ClientAcceptance (features count)		
4,4	4,4	36,7	0	4	12
			Development resources		
101,0			0		86

Current Status	Improvements		XML Web Services		
Units	Units	%	Past	Tolerable	Goal
			TransferDefinition.Usability.Efficiency		
7,0	9,0	81,8	16	10	5
17,0	8,0	53,3	25	15	10
			TransferDefinition.Usability.Response		
943,0	-186,0	######	170	60	30
			TransferDefinition.Usability.Intuitiveness		
5,0	10,0	95,2	15	7,5	4,5
			Development resources		
2,0			0		48

Trond Johansen

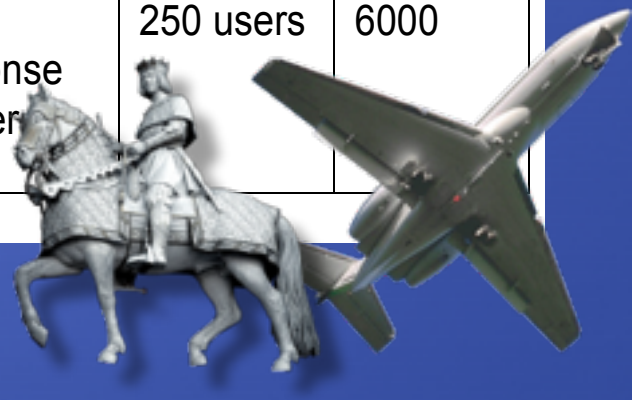
Confirmit Evo-week cycle: Measure Progress Weekly

	Development Team	Users (PMT, Pros, Doc writer, other)	CTO (Sys Arch, Process Mgr)	QA (Configuration Manager & Test Manager)
Friday	✓ PM: Send Version N detail plan to CTO + prior to Project Mgmt meeting ✓ PM: Attend Project Mgmt meeting: 12.00-15.00 ✓ Developers: Focus on genereal maintenance work, documentation.		✓ Approve/reject design & Step N ✓ Attend Project Mgmt meeting: 12-15	✓ Run final build and create setup for Version N-1. ✓ Install setup on test servers (external and internal) ✓ Perform initial crash test and then release Version N-1
Monday	✓ Develop test code & code for Version N	✓ Use Version N-1		✓ Follow up CI ✓ Review test plans, tests
Tuesday	✓ Develop Test Code & Code for Version N ✓ Meet with users to Discuss Action Taken Regarding Feedback From Version N-1	✓ Meet with developers to give Feedback and Discuss Action Taken from previous actions	✓ System Architect to review code and test code	✓ Follow up CI ✓ Review test plans, tests
Wednesday	✓ Develop test code & code for Version N			✓ Review test plans, tests ✓ Follow up CI
Thursday	✓ Complete Test Code & Code for Version N ✓ Complete GUI tests for Version N-2			✓ Review test plans, tests ✓ Follow up CI

Evo's impact on Confirmit product qualities

- **Only 5 highlights of the 25 impacts are listed here**

Description of requirement/work task	Past	Status
Usability.Productivity: Time for the system to generate a survey	7200 sec	15 sec
Usability.Productivity: Time to set up a typical specified Market Research-report (MR)	65 min	20 min
Usability.Productivity: Time to grant a set of End-users access to a Report set and distribute report login info.	80 min	5 min
Usability.Intuitiveness: The time in minutes it takes a medium experienced programmer to define a complete and correct data transfer definition with Confirmit Web Services without any user documentation or any other aid	15 min	5 min
Performance.Runtime.Concurrency: Maximum number of simultaneous respondents executing a survey with a click rate of 20 sec and an response time<500 ms, given a defined [Survey-Complexity] and a defined [Server Configuration, Typical]	250 users	6000



confirmit®

Release 8.5

JUST-IN-TIME PLANNING:
Dynamic intelligent do-next prioritisation:
Value/cost based

- Can you buy into this planning policy?
  - Do, in the next value delivery cycle, that which is estimated to give most value, to all objectives, with regard to risk

Notice the automatically computed priority colours, after each delivery and measurement cycle

Sum impact after NINE deivery cycles

Impact Estimation Table: Reportal c… "Hyggen"

Current Status			Improvements			Reportal - E-SAT features
Units	Units	%	Past	Tolerable	Goal	
			Usability.Intuitivness (%)			
75,0	25,0	62,5	50	75	90	
			Usability.Consistency.Visual (Elements)			
14,0	14,0	100,0	0	11	14	
			Usability.Consistency.Interaction (Components)			
15,0	15,0	107,1	0	11	14	
			Usability.Productivity (minutes)			
5,0	75,0	96,2	80	5	2	
5,0	45,0	95,7	50	5	1	
			Usability.Flexibility.OfflineReport.ExportFormats			
3,0	2,0	66,7	1	3	4	
			Usability.Robustness (errors)			
1,0	22,0	95,7	7	1	0	
			Usability.Replacability (nr of features)			
4,0	5,0	100,0	8	5	3	
			Usability.ResponseTime.ExportReport (minutes			
1,0	12,0	150,0	13	13	5	
			Usability.ResponseTime.ViewReport (seconds)			
1,0	14,0	100,0	15	3	1	
			Development resources			
203,0			0		191	

Current Status			Improvements			Survey Engine .NET
Units	Units	%	Past	Tolerable	Goal	
83,0	48,0	80,0	Backwards.Compatibility (%)			
			40	85	95	
0,0	67,0	100,0	67	0	0	
			Generate.WI.Time (small/medium/large seconds)			
4,0	59,0	100,0	63	8	4	
10,0	397,0	100,0	407	100	10	
94,0	2290,0	103,9	2384	500	180	
			Testability (%)			
10,0	10,0	13,3	0	100	100	
			Usability.Speed (seconds/user rating 1-10)			
774,0	507,0	51,7	1281	600	300	
5,0	3,0	60,0	2	5	7	
			Runtime.ResourceUsage.Memory			
0,0	0,0	0,0		?	?	
			Runtime.ResourceUsage.CPU			
3,0	35,0	97,2	38	3	2	
			Runtime.ResourceUsage.MemoryLeak			
0,0	800,0	100,0	800	0	0	
			Runtime.Concurrency (number of users)			
850,0	1100,0	146,7	150	500	1000	
			Development resources			
64,0			0		84	

Current Status			Improvements			Reportal - MR Features
Units	Units	%	Past	Tolerable	Goal	
			Usability.Replacability (feature count)			
1,0	1,0	50,0	14	13	12	
			Usability.Productivity (minutes)			
20,0	45,0	112,5	65	35	25	
			Usability.ClientAcceptance (features count)			
4,4	4,4	36,7	0	4	12	
			Development resources			
101,0			0		86	

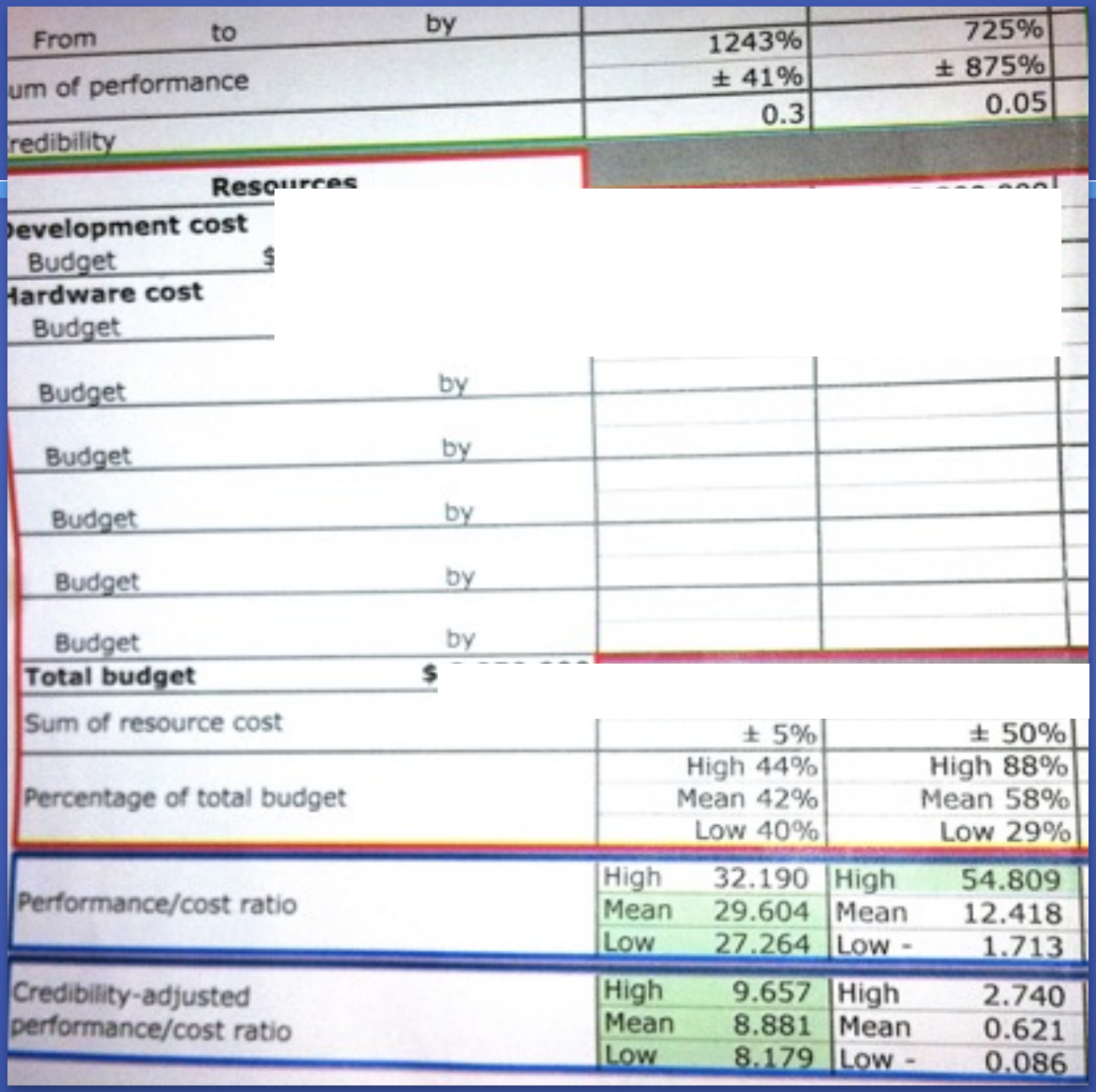
Current Status			Improvements			XML Web Services
Units	Units	%	Past	Tolerable	Goal	
7,0	9,0	81,8	TransferDefinition.Usability.Efficiency			
			16	10	5	
17,0	8,0	53,3	25	15	10	
			TransferDefinition.Usability.Response			
943,0	-186,0	######	170	60	30	
			TransferDefinition.Usability.Intuitiveness			
5,0	10,0	95,2	15	7,5	4,5	
			Development resources			
2,0			0		48	

Trond Johansen

Example: Impact Estimations
B was, as you see, done with *great* uncertainty

Requirements	Opt A	Opt B
TIME.TRADE From 10 to 4 by Dec-11	100%	± 60%
TIME.HEDGE From 30 to 3 by Dec-11	100%	75% ± 60%
SPEED.CODE From 60 to 5 by Dec-11	100%	65% ± 10%
PNL.ADJUST From 60 to 15 by Dec-11	90% ± 10%	85% ± 100%
CAP.TXNS From 62000 to 500000 by Dec-11	100%	50% ± 100%
CAP.PEAK From 6000 to 100000 by Dec-11	100%	25% ± 100%
CAP.BURST From 20 to 200 by Dec-11	75% ± 10%	0% ± 100%
CAP.POSNS From 4000 to 40000 by Dec-11	100%	100% ± 40%
CAP.TRADERS From 180 to 270 by Dec-11	90% ± 10%	100% ± 30%
AVAIL.P1 From 100 to 20 by Dec-11	90% ± 10%	50% ± 75%
RISK.MANAGE From 0 to 100 by Dec-11	100%	50% ± 50%
RISK.TIME From 99 to 100 by Dec-11	100%	0% ± 100%
RISK.REFRESH From 500 to 200 by Dec-11	98% ± 1%	50% ± 50%

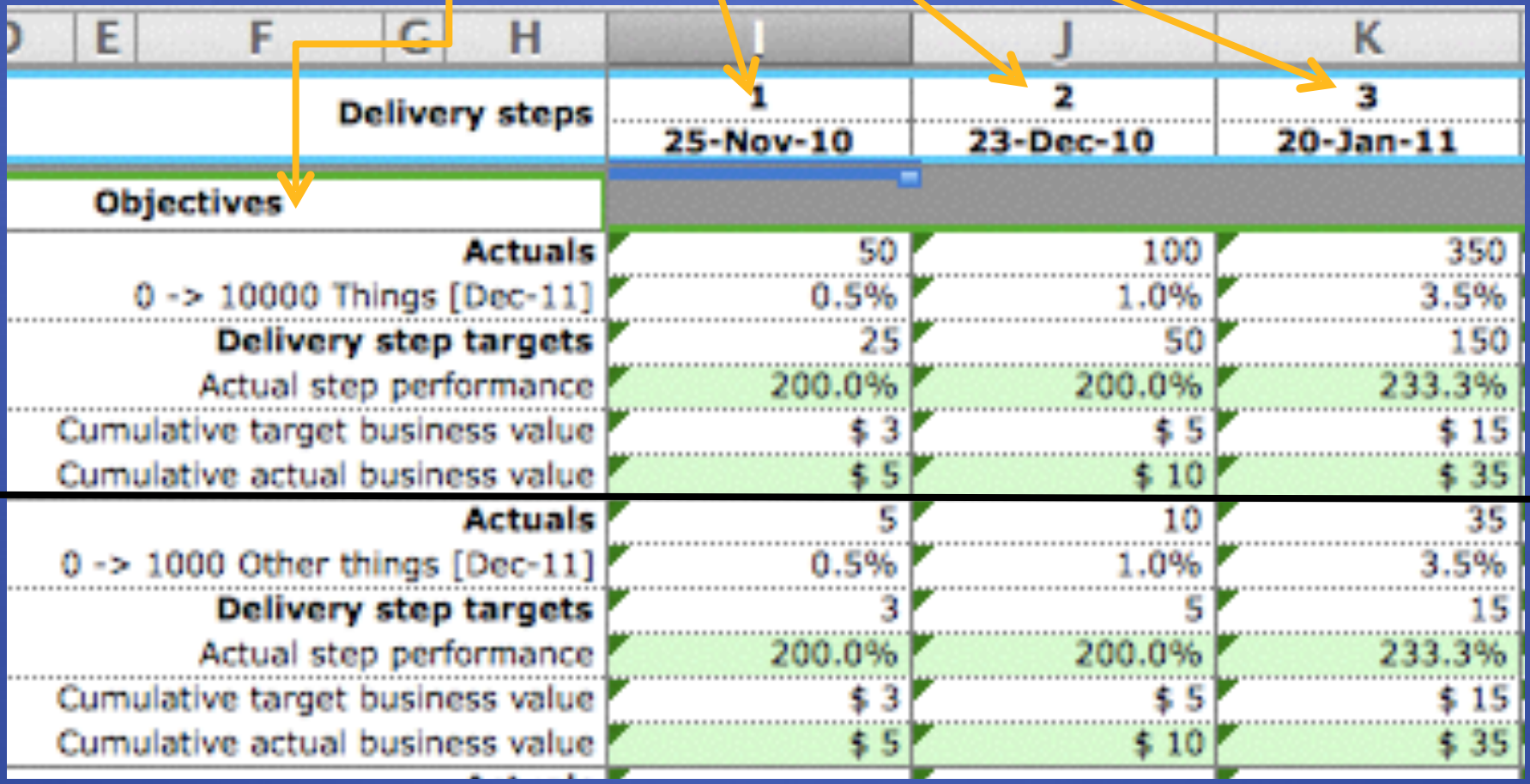
The Bottom Line



From	to	by		
um of performance			1243%	725%
redibility			± 41%	± 875%
			0.3	0.05
Resources				
evelopment cost				
Budget	$			
Hardware cost				
Budget				
Budget		by		
Budget		by		
Budget		by		
Budget		by		
Budget		by		
Total budget	$			
Sum of resource cost			± 5%	± 50%
Percentage of total budget			High 44% Mean 42% Low 40%	High 88% Mean 58% Low 29%
Performance/cost ratio			High 32.190 Mean 29.604 Low 27.264	High 54.809 Mean 12.418 Low - 1.713
Credibility-adjusted performance/cost ratio			High 9.657 Mean 8.881 Low 8.179	High 2.740 Mean 0.621 Low - 0.086

Tracking 3 delivery-steps, for 2 Objectives

(teaching example, not real)



	E	F	G	H	I	J	K
				Delivery steps	**1**	**2**	**3**
					25-Nov-10	**23-Dec-10**	**20-Jan-11**
		Objectives					
				Actuals	50	100	350
				0 -> 10000 Things [Dec-11]	0.5%	1.0%	3.5%
				Delivery step targets	25	50	150
				Actual step performance	200.0%	200.0%	233.3%
				Cumulative target business value	$ 3	$ 5	$ 15
				Cumulative actual business value	$ 5	$ 10	$ 35
				Actuals	5	10	35
				0 -> 1000 Other things [Dec-11]	0.5%	1.0%	3.5%
				Delivery step targets	3	5	15
				Actual step performance	200.0%	200.0%	233.3%
				Cumulative target business value	$ 3	$ 5	$ 15
				Cumulative actual business value	$ 5	$ 10	$ 35

It is fascinating how focused and creative the dialogue becomes between domain experts when they are guided by quantified goal sets, the need to estimate , give evidence, state uncertainty and assign credibility.
All culminating in decision documentation which is auditable reviewable.
Improvable and transparent! <- TG 12-20xx

Make friends by delivering results.

- Get out of the Nerd Mode of delivering functions/stories to a user
- Get into the mode of delivering real measurable results, the highest value, to stakeholders

Shock your boss!

- Insist on being stakeholder-value oriented, rather than IT oriented

The end

- What is wrong with requirements