IT Decision-Making

for

Managers & Senior Consultants: using

Value Planning Methods

a 1 day course by gilb.com

Slide Version 17 Sept 2015 12:23

Content of the Day

1 Quantification of critical values and qualities in requirements and objectives

2 Specification of background information to help understand risks and priorities

3 Impact Estimation Tables: a tool for comparing complex options, architectures and strategies.

4 Dynamic Decision Making: learning fast, committing late

5 Delegation of Decision Making: to where the action and competence is placed.

6 Agile Contracting: decisions and commitments in smaller increments

7 Evo: a project planning framework for decision making

1 Quantification of critical values and qualities in requirements and objectives

Main Idea:
Go Digital
Drop the
‘Poetry’

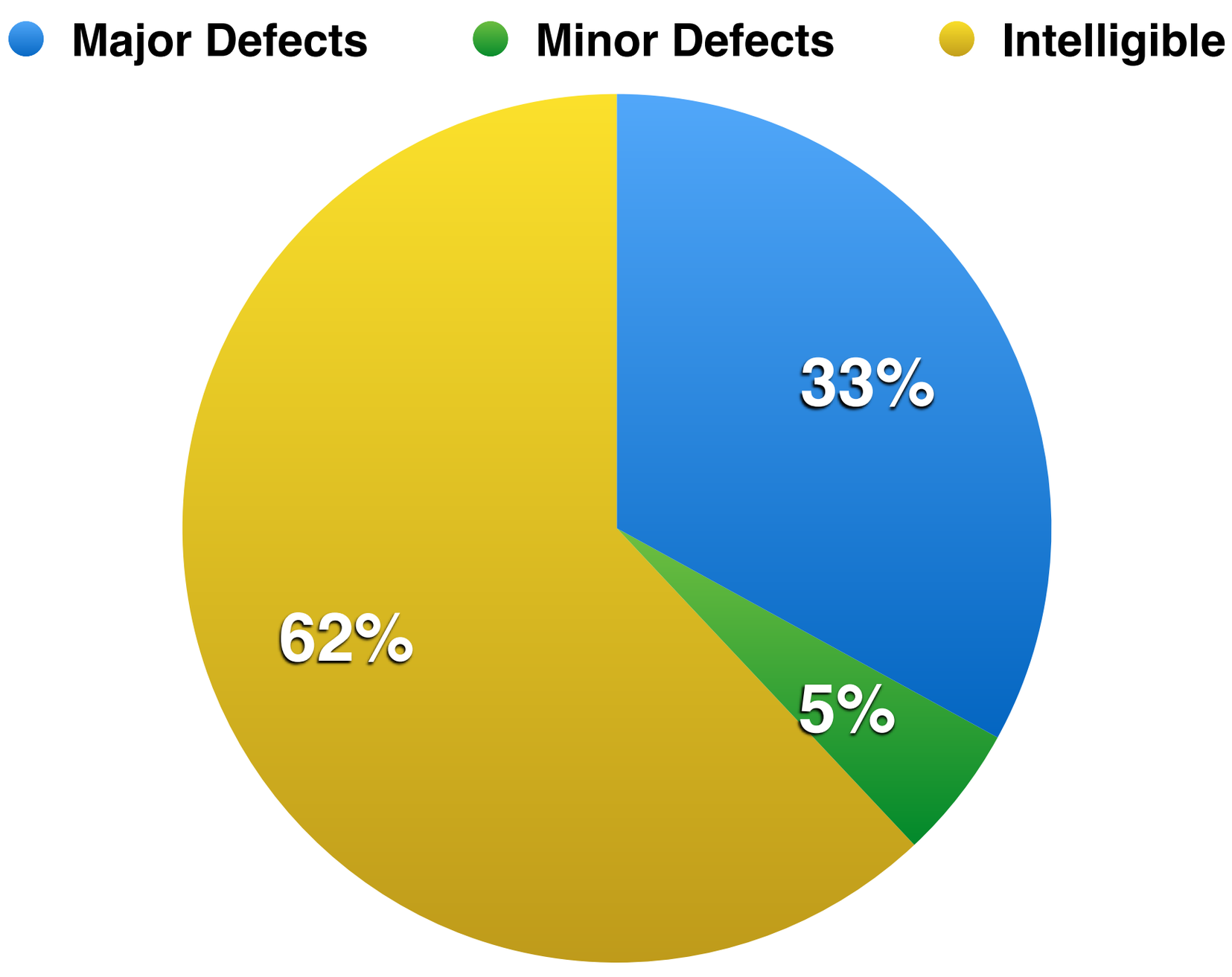
"In physical science the first <u>essential step</u> in the direction of learning any subject is to <u>find principles of numerical reckoning</u> and <u>practicable methods for measuring</u> some <u>quality</u> connected with it.

I often say that when you can <u>measure</u> what you are speaking about, and <u>express it</u> in numbers, you know something about it;
but when you cannot <u>measure</u> it, when you cannot <u>express it in numbers</u>, your knowledge is of a meagre and unsatisfactory kind;

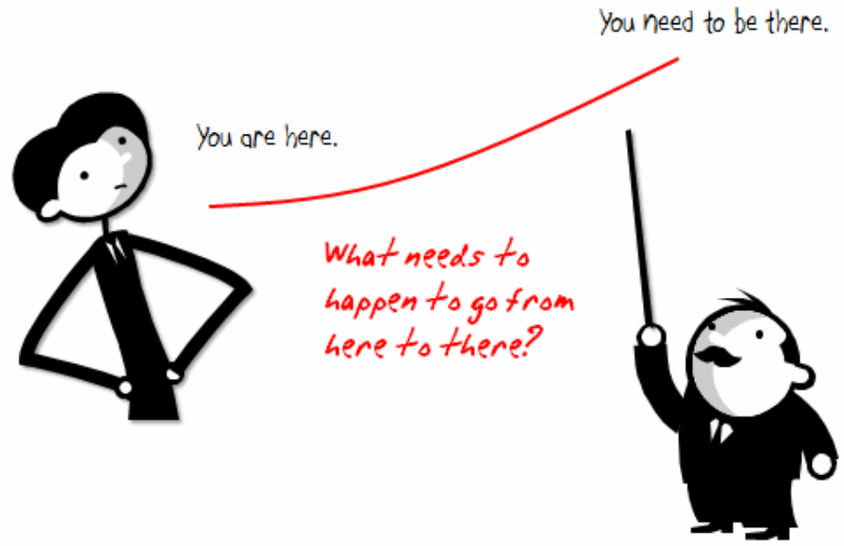
it may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the state of Science, whatever the matter may be.”

Lord Kelvin, 1893, Lecture to the Institution of Civil Engineers, 3 May 1883

% Intelligible Plans



Real Example
"Platform Rationalisation Initiative"
"<u>Main Objectives.</u>"
<u>London Multinational Bank</u>



- **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.**
- ***<u>Improved development environment</u>*, leading to increased capacity to enhance functionality in future.**
- **Removes duplicative spend on two back office platforms in support of mandatory message changes, etc.**

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.

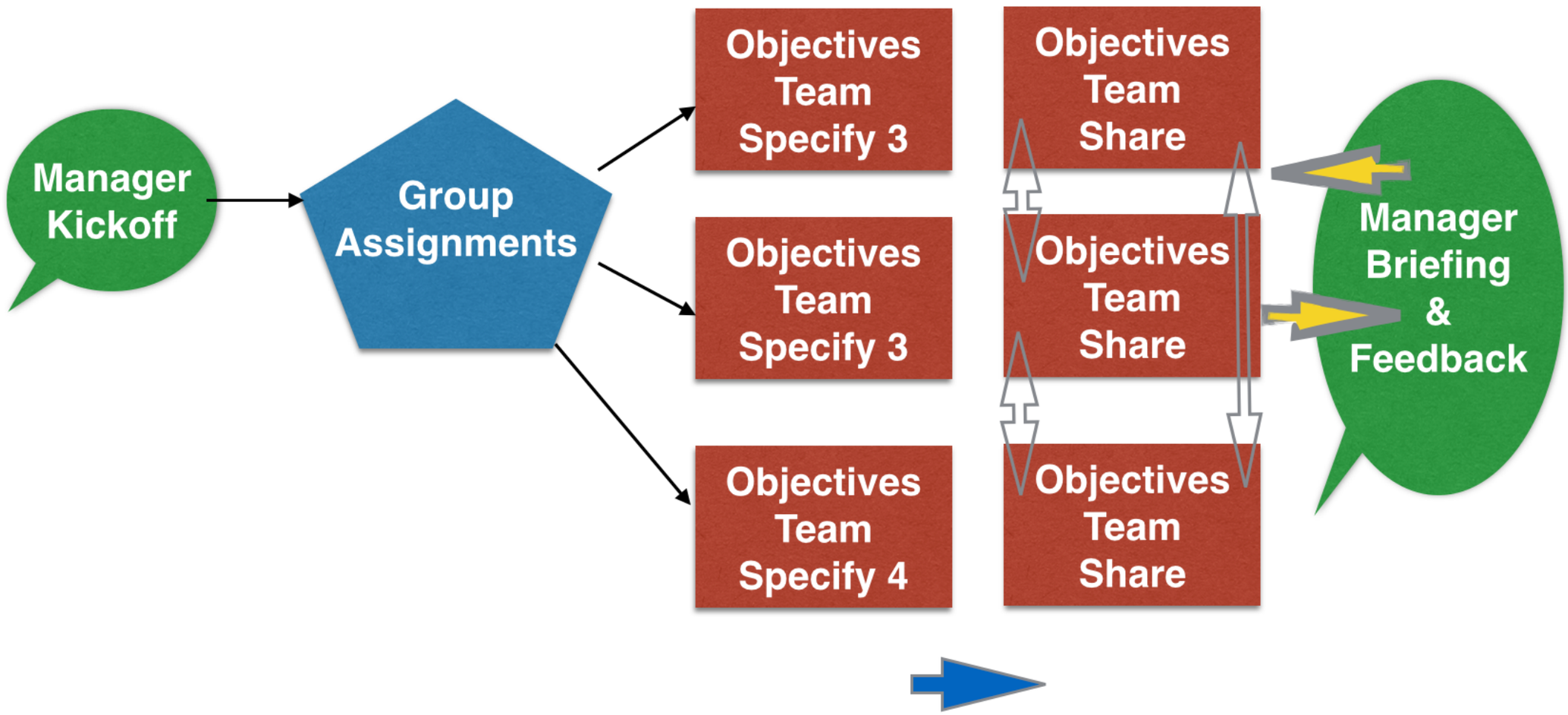
Past [2009, {Bxx, Lxx, Gxx}, If QA Approved Processes used, Developer = Architect, Task = Draft Architecture] **15 days** ±4 ?? <- Rob

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

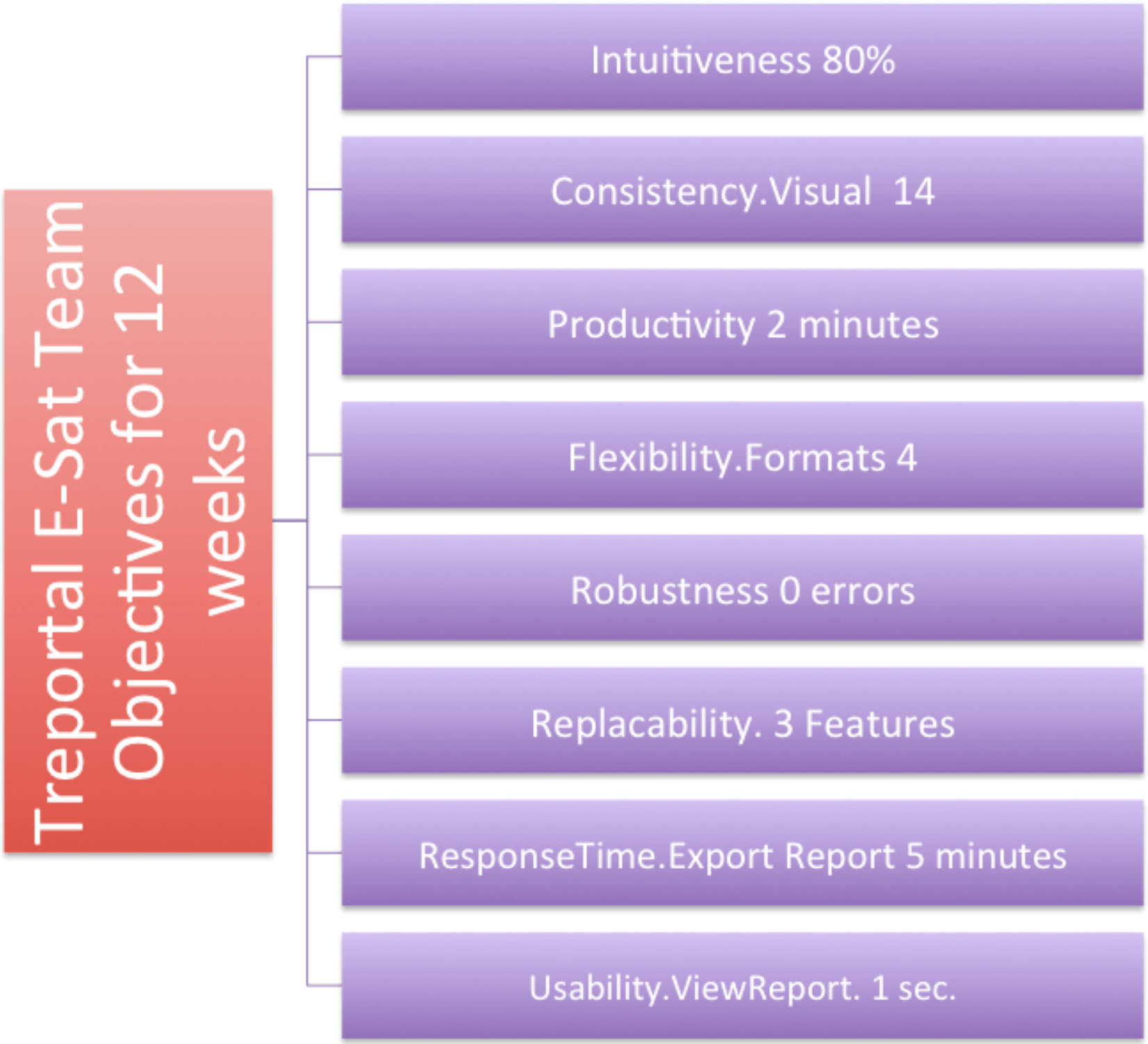
Justification: Really good architects are very scarce so we need to optimize their use.

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

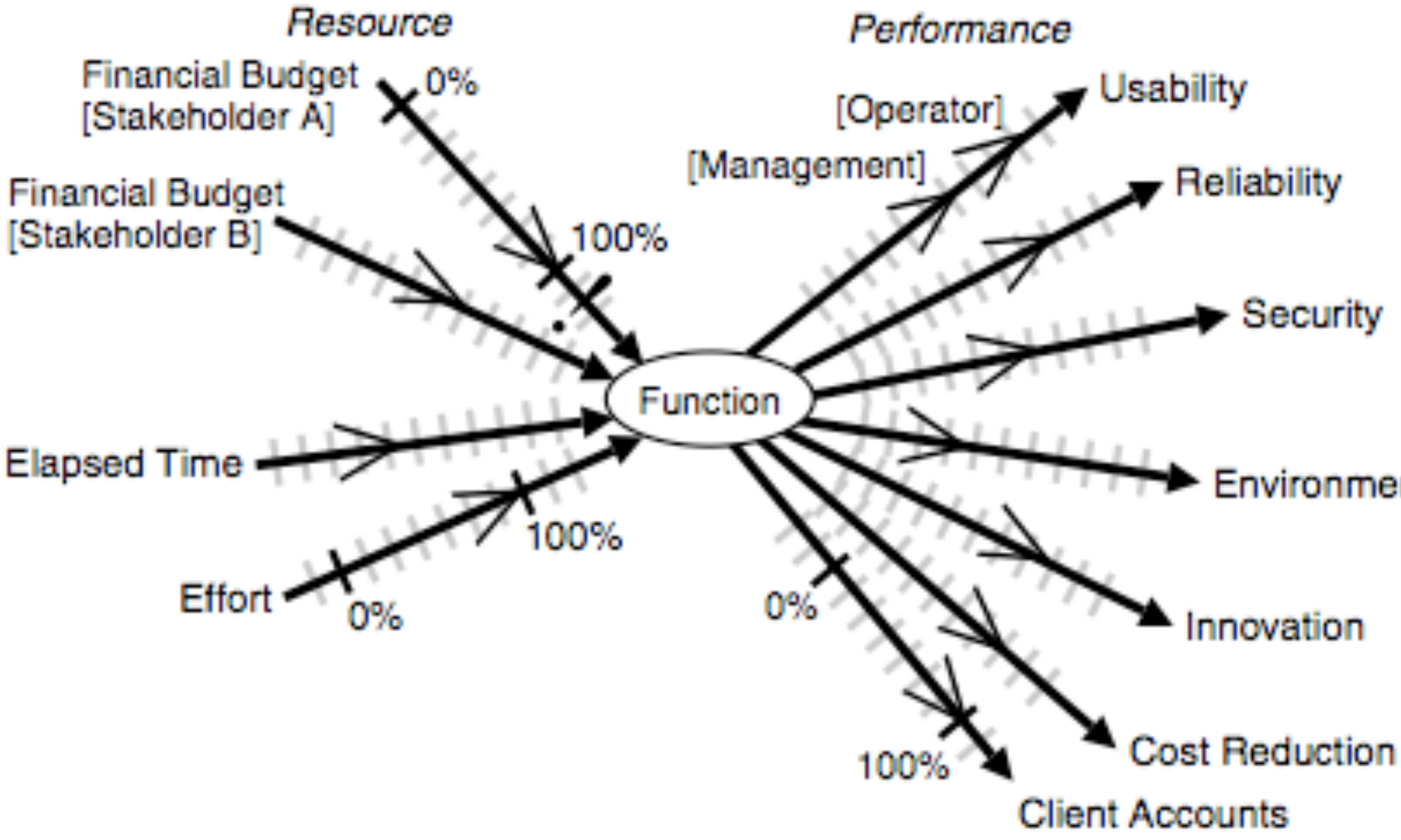
The First Day of the Startup Process
Top Ten Critical Values
a *quantification* process



Example of Top Ten Critical Objectives (Real Set, Confirmit)

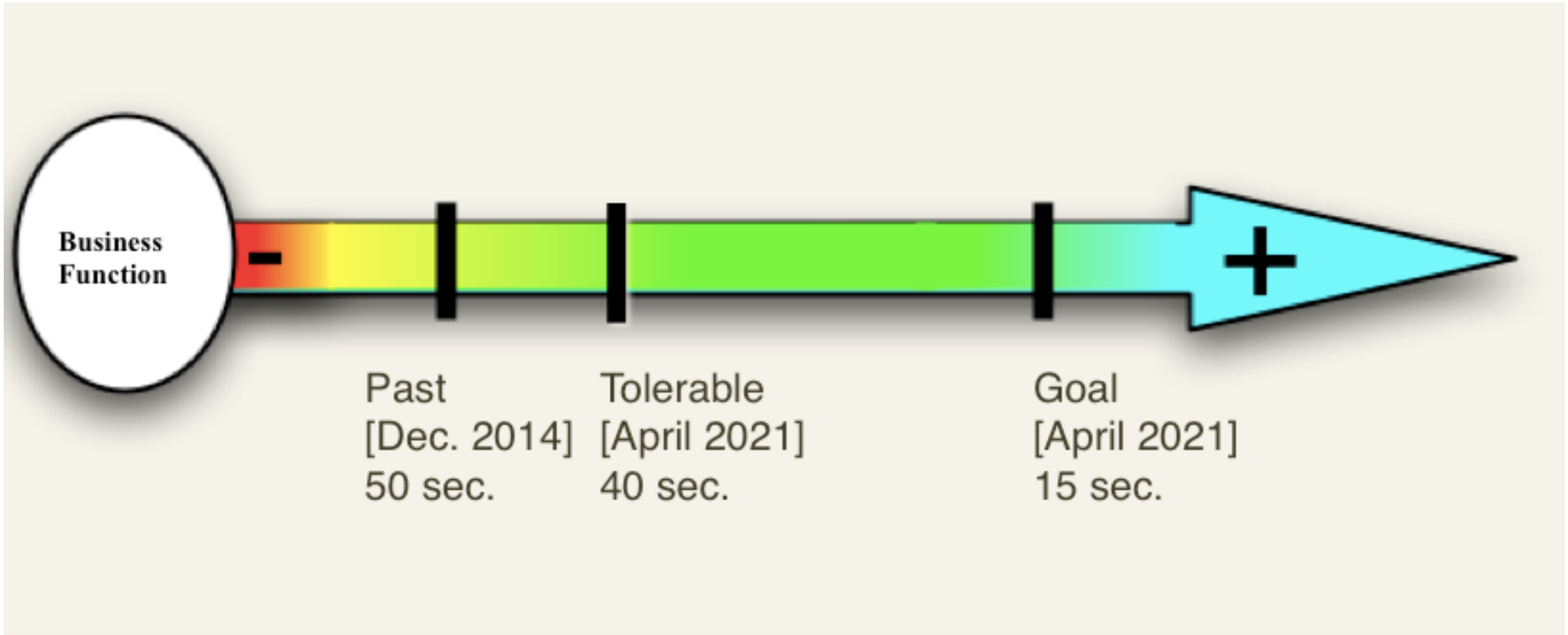


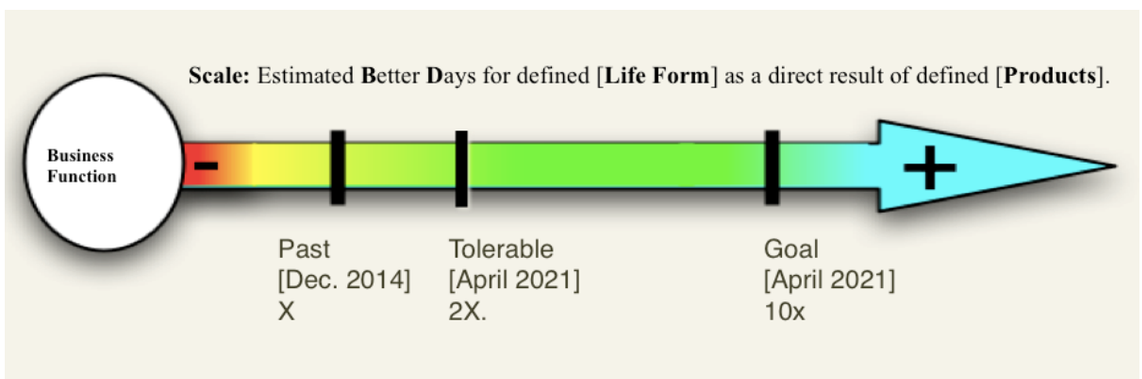
Many variable Critical Values to be managed at once

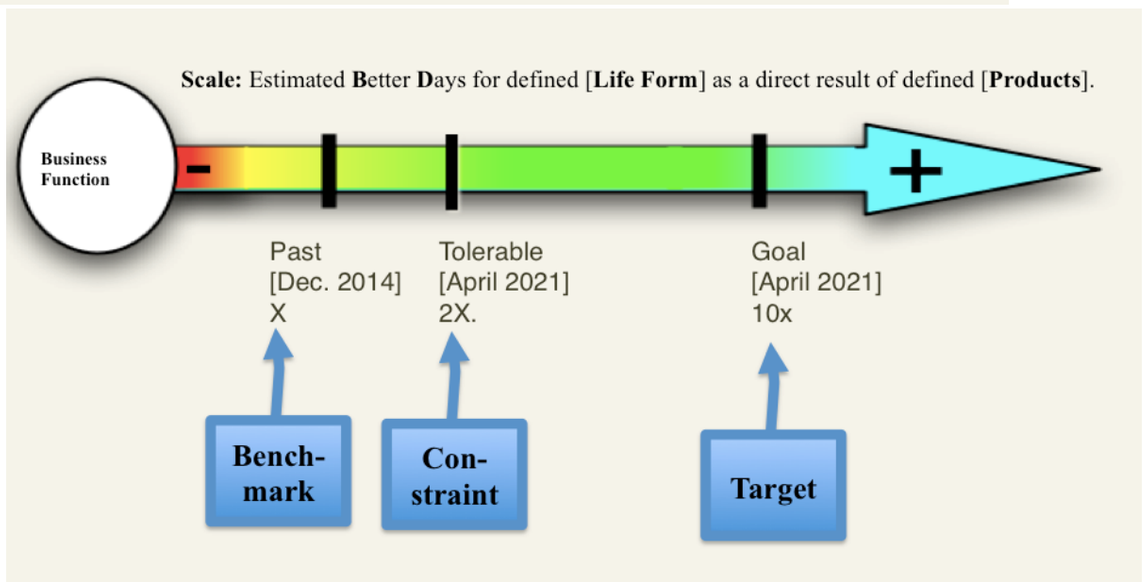


THE QUANTIFICATION PRINCIPLE

Performance objectives,
ranging from *core objectives* to 'any' detailed performance objective
– where 'getting better-and-better in time' is implied –
can *always* be defined using 'scales of measure'.







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 %**

TWELVE TOUGH QUESTIONS

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

http://www.gilb.com/tiki-download_file.php?fileId=24

2 Specification of background information to help understand risks and priorities

In addition to ‘Core’ specification,
the Value Driven planning language allows you to specify many other value-related things in a single requirement

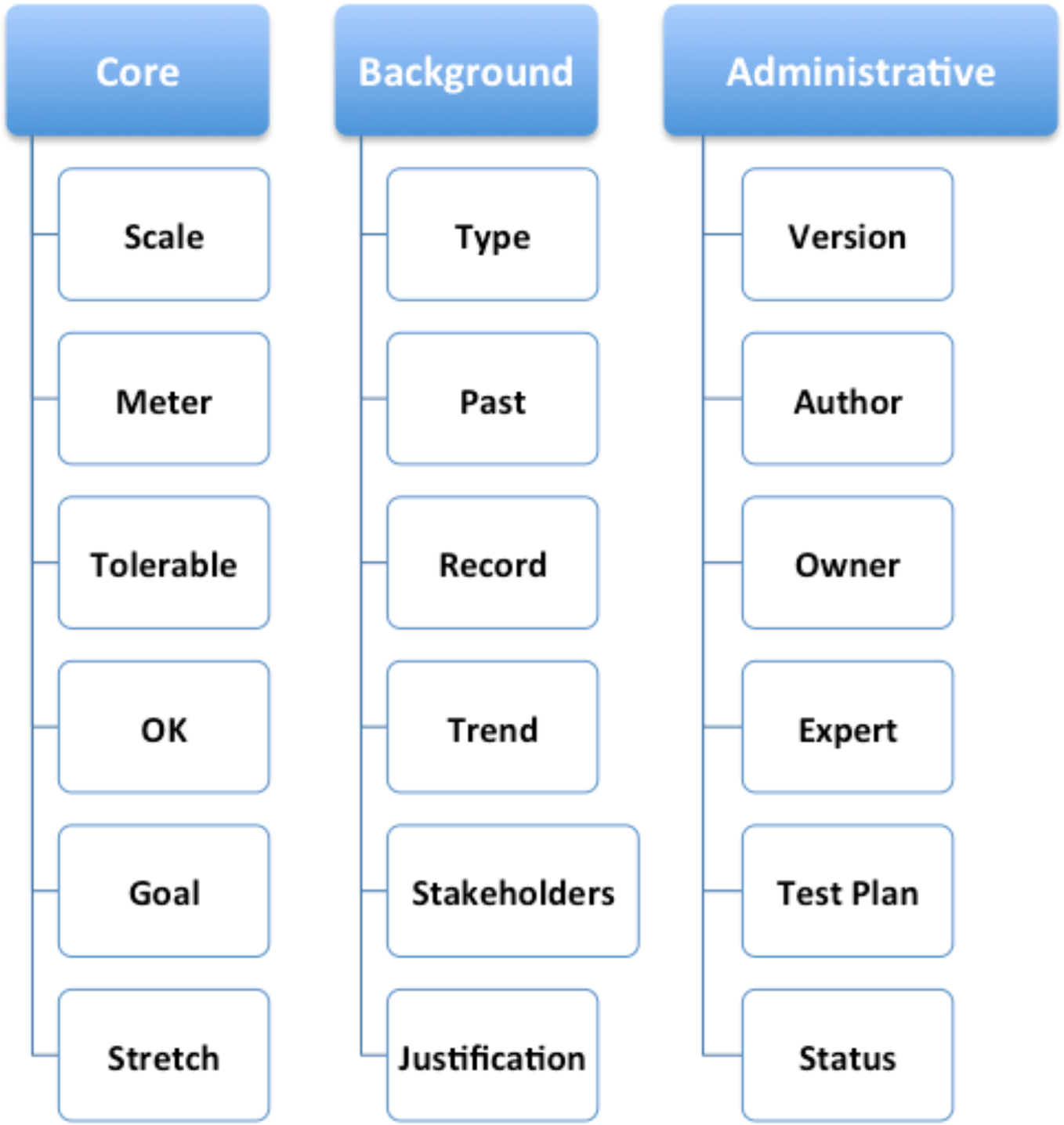
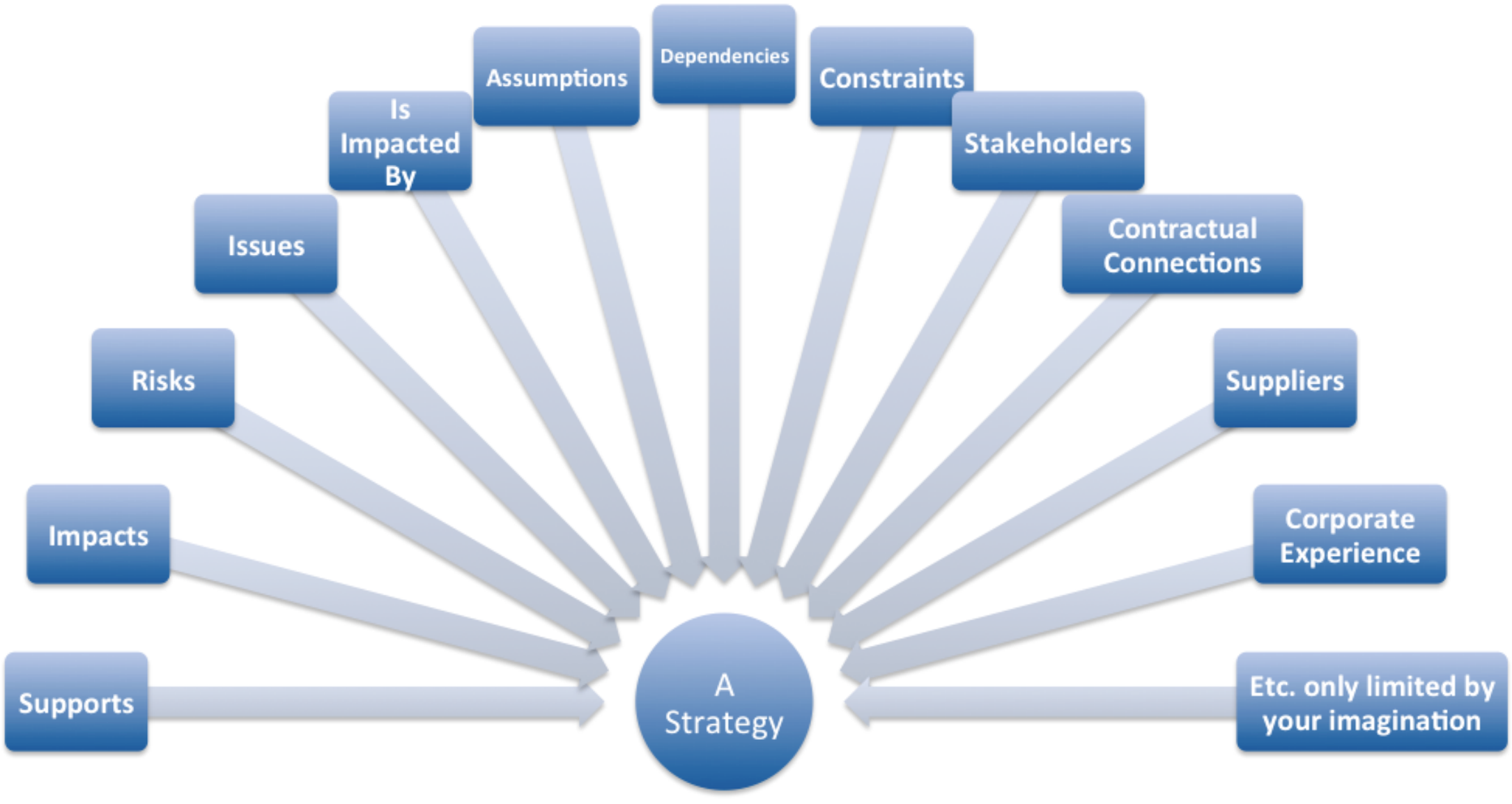


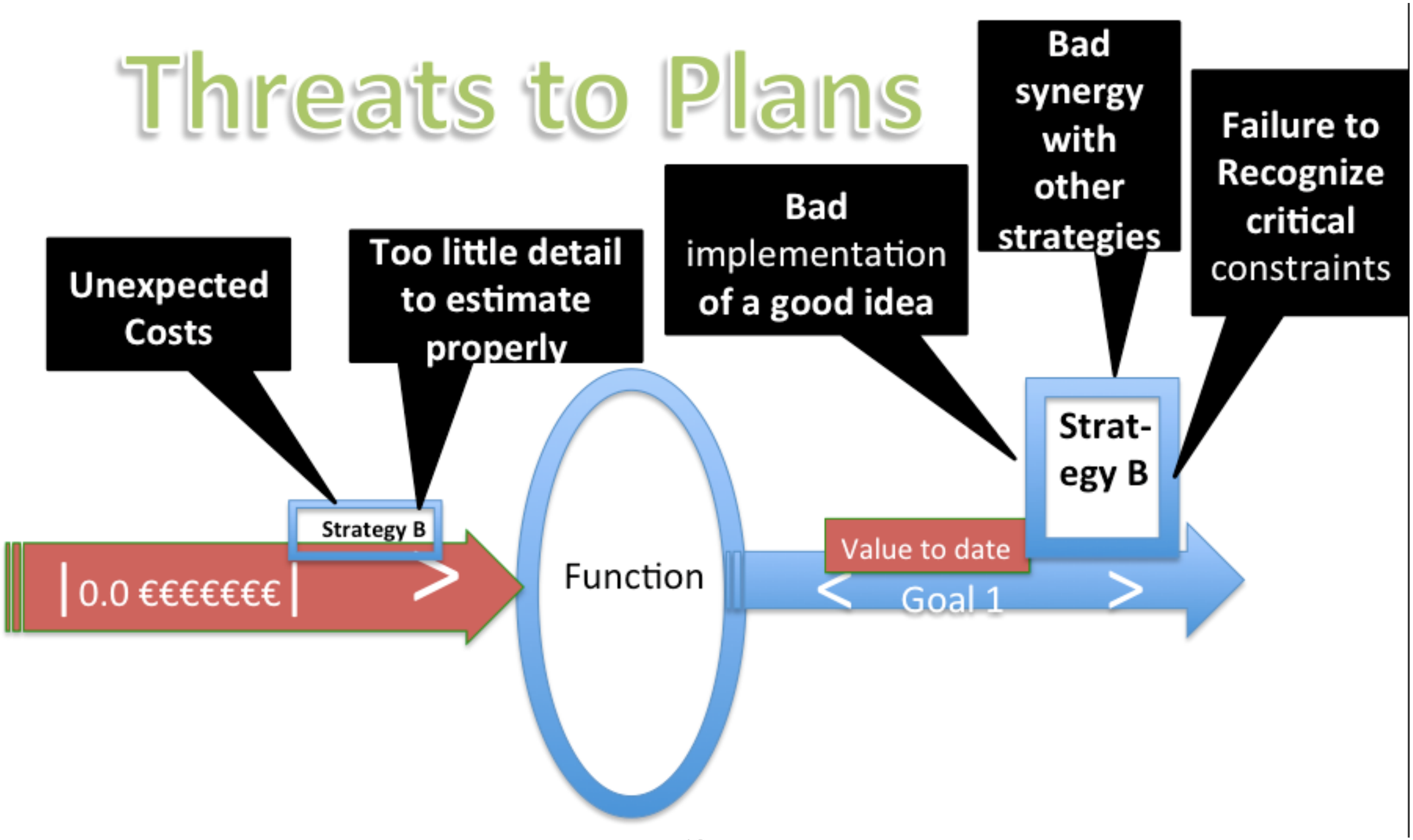
Figure: *682. Some Examples Of Core, Background, And Administrative Parameters. (Source ‘Value Planning’ Diagram 4.3, Aug 2015)

Design Strategy Relationships



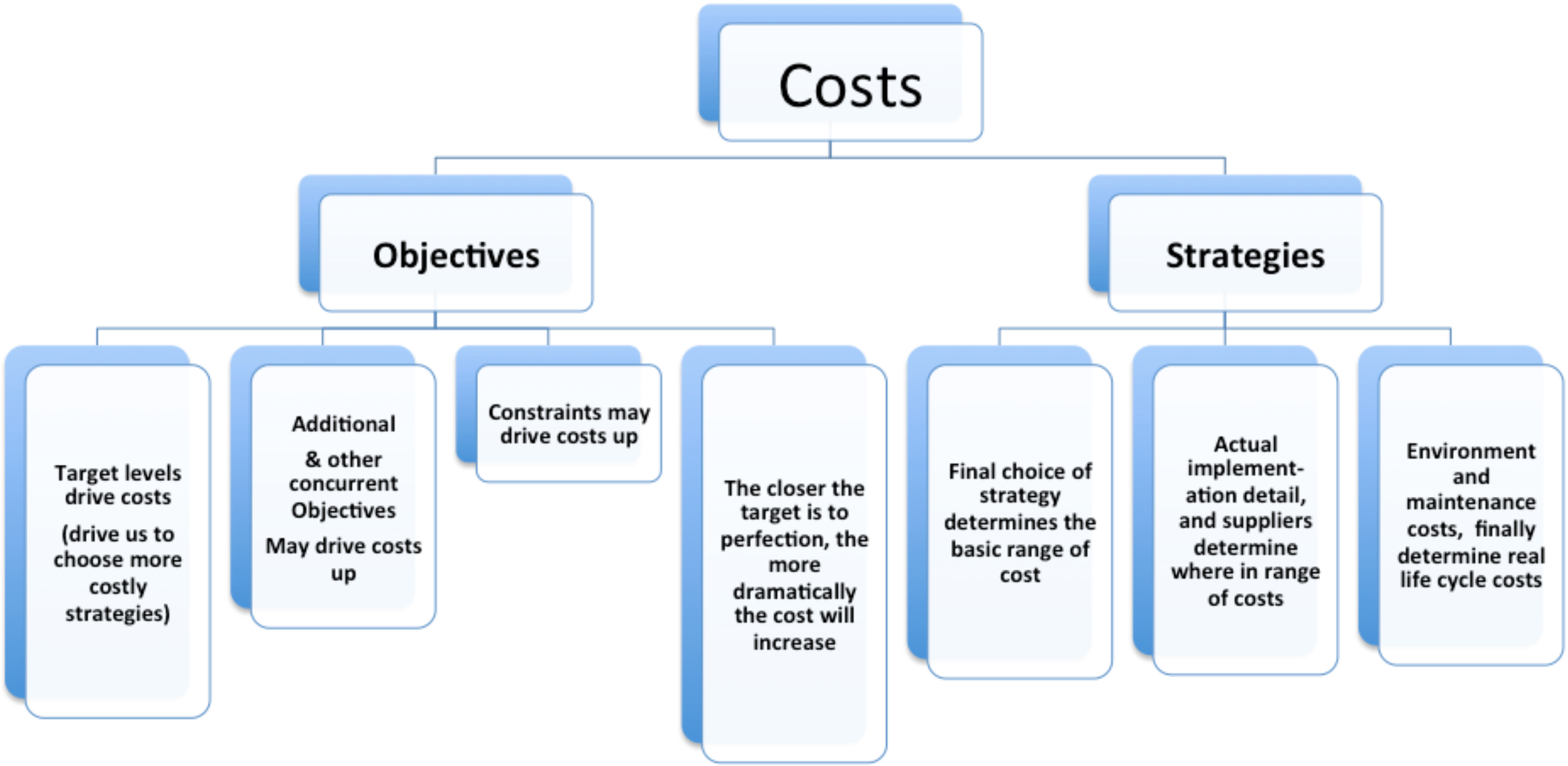
3 Impact Estimation Tables: a tool for comparing complex options, architectures and strategies.

Various Risks to Plans

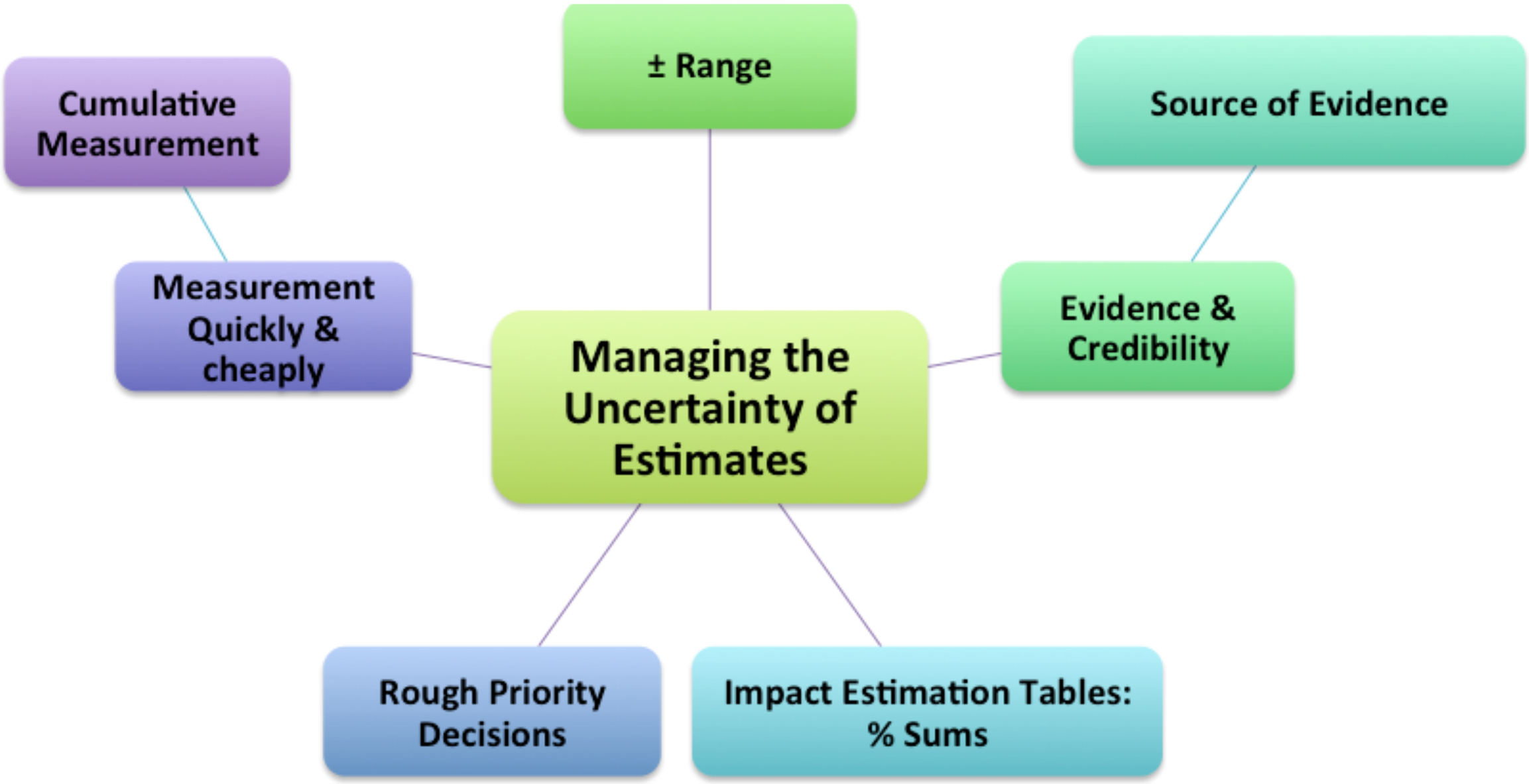


Design Strategy Risks

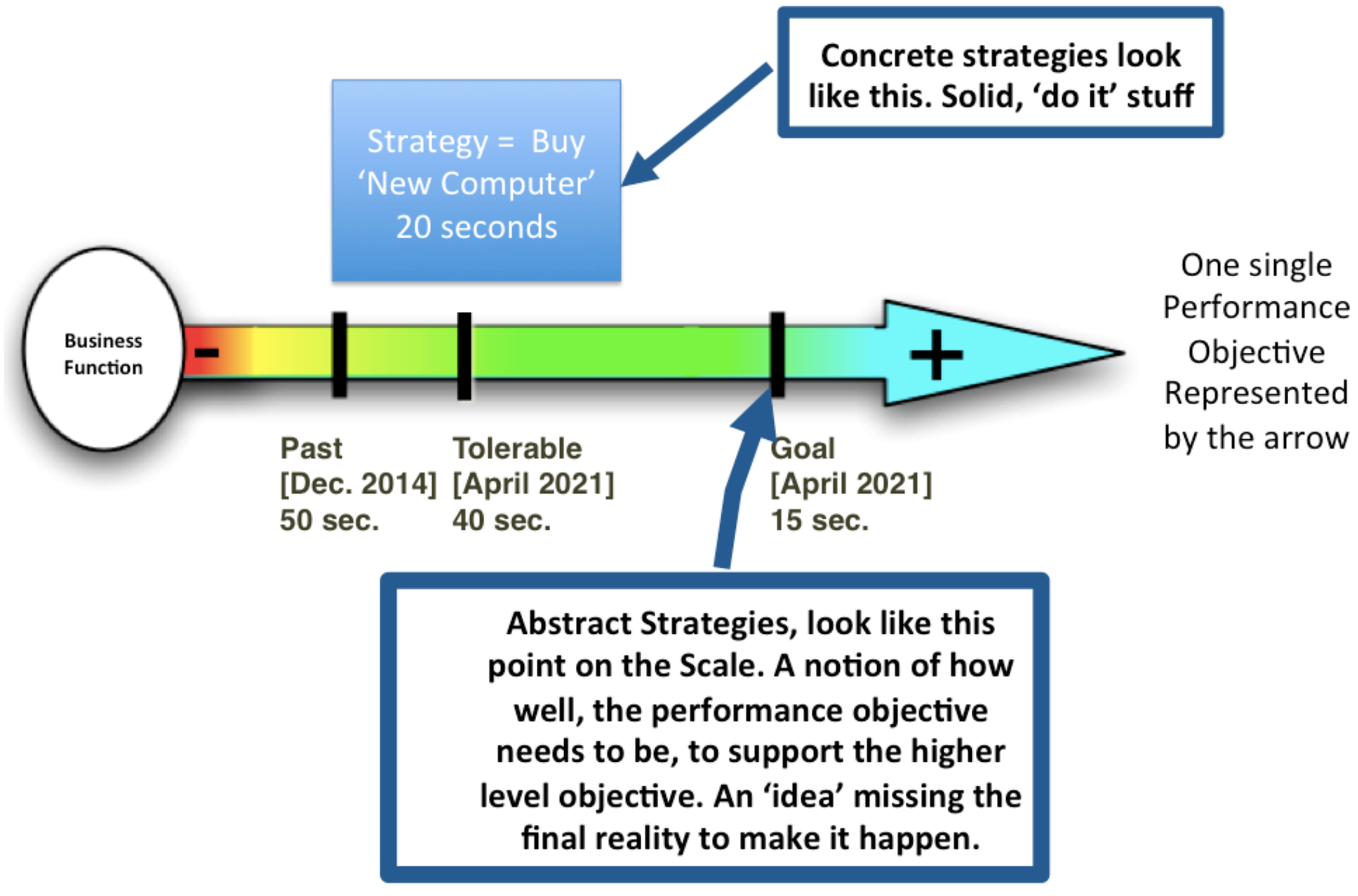
Cost Risks

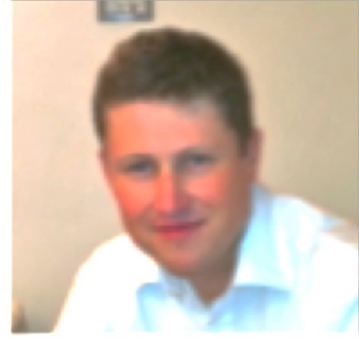

Costs
Objectives
Strategies
Target levels drive costs
(drive us to choose more costly strategies)
Additional
& other concurrent Objectives
May drive costs up
Constraints may drive costs up
The closer the target is to perfection, the more dramatically the cost will increase
Final choice of strategy determines the basic range of cost
Actual implement-ation detail, and suppliers determine where in range of costs
Environment and maintenance costs, finally determine real life cycle costs

Risk Tools in Impact Estimation

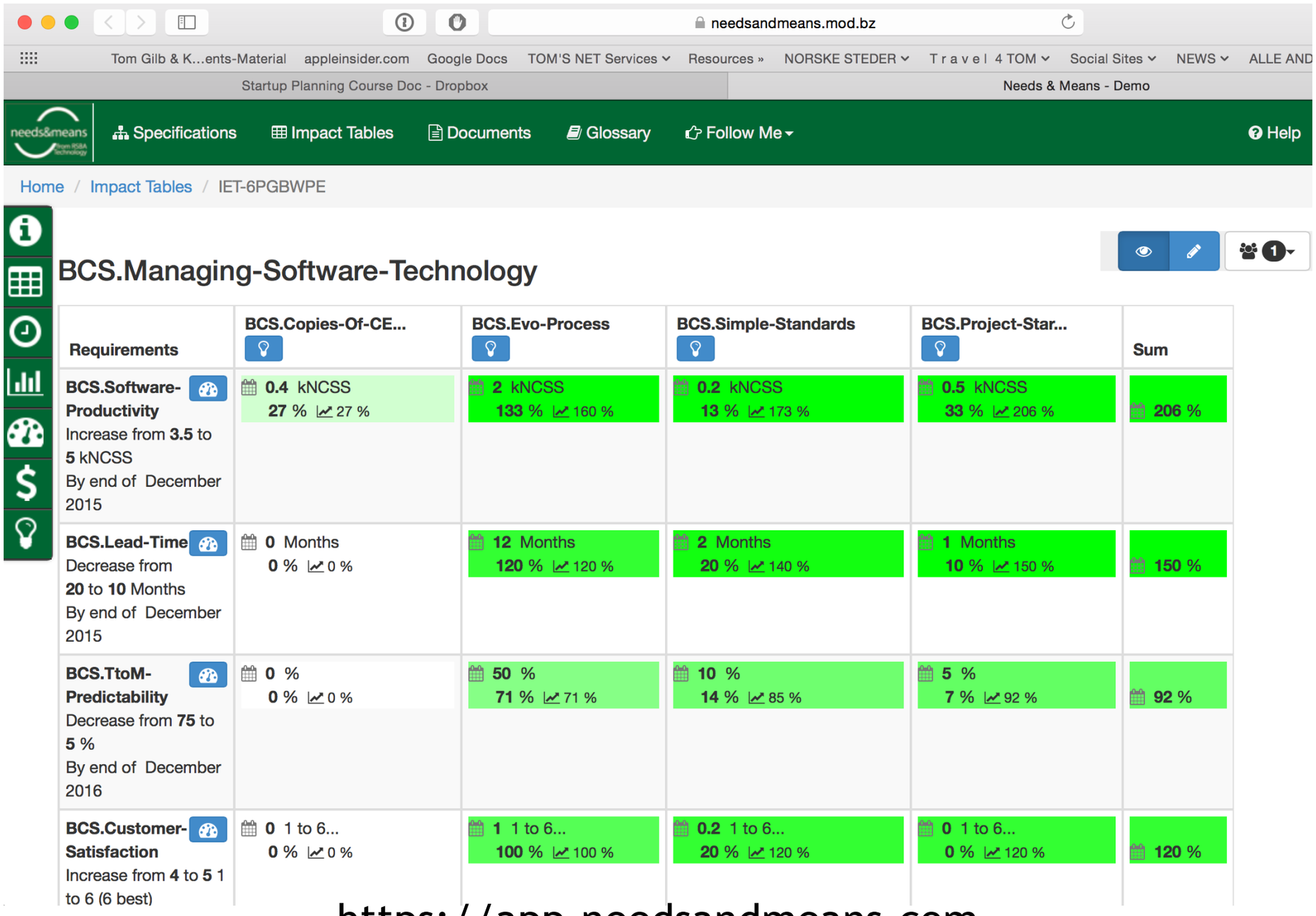


Abstract and Concrete Value Strategies





Richard Smith's Planning Tool which we are using on BCS Courses Great for 'First Week' and all later weeks followup



https://app.needsandmeans.com

Day 3 of Project Startup

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

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

Citigroup, London

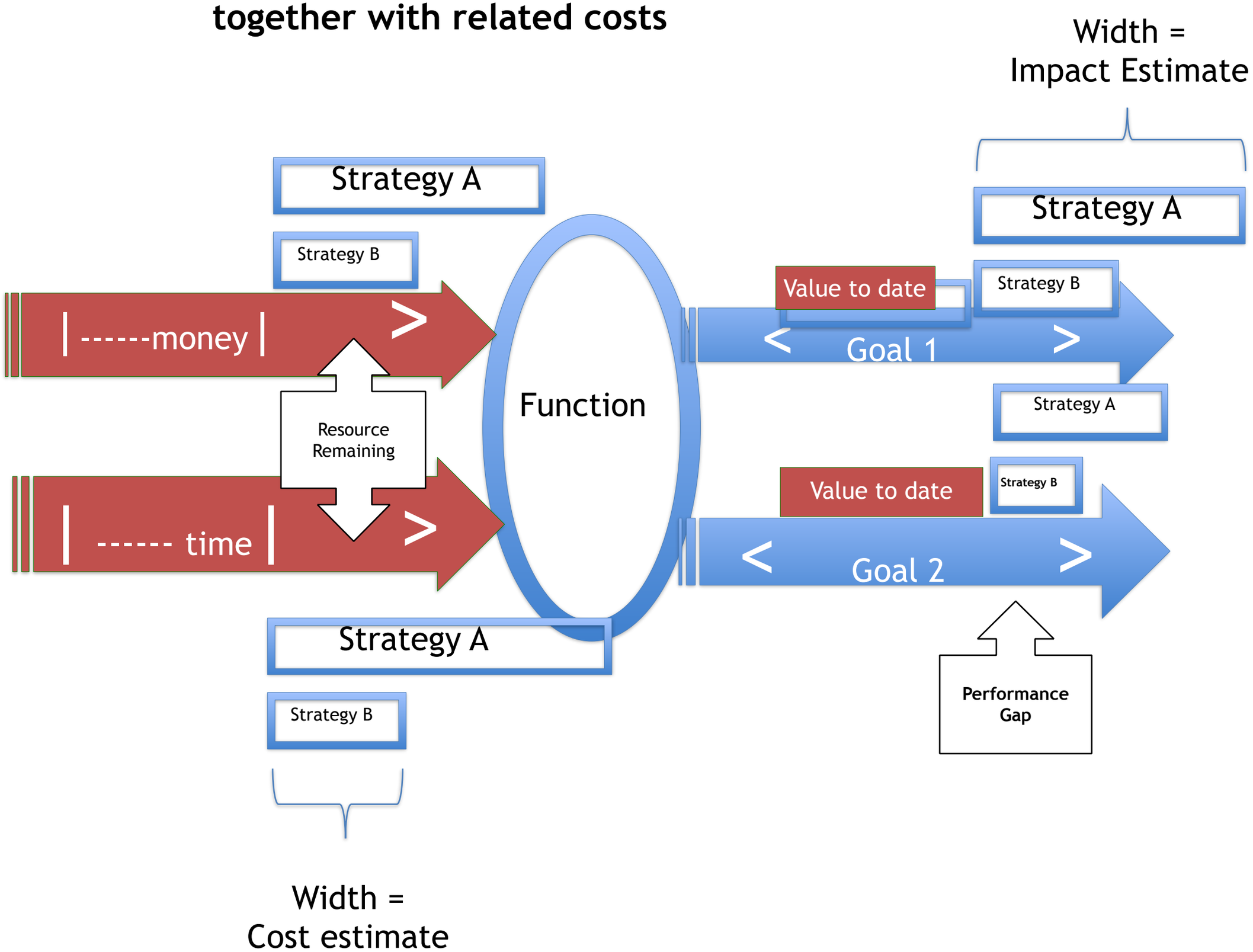
Figure 4. Acer Project: Impact Estimation Table.

A Real London Impact Estimation Table
Made one day, to get £50,000,000 next day

		.....**Deliverables**						
		Telephony	Modularity	Tools	User Experience	GUI & Graphics	Security	Enterprise
Business Objective								
Time to Market		10%	10%	15%	0%	0%	0%	5%
Product Range		0%	30%	5%	10%	5%	5%	0%
Platform Technology		10%	0%	0%	5%	0%	10%	5%
Units		15%	5%	5%	0%	0%	10%	10%
Operator Preference		10%	5%	5%	10%	10%	20%	10%
Commoditization		10%	-20%	15%	0%	0%	5%	5%
Duplication		10%	0%	0%	0%	0%	5%	5%
Competitiveness		15%	10%	10%	10%	20%	10%	10%
User Experience		0%	20%	0%	30%	10%	0%	0%
Downstream Cost Saving		5%	10%	0%	10%	0%	0%	5%
Other Country		5%	10%	0%	10%	5%	0%	0%
Total Contribution		90%	80%	55%	85%	50%	65%	55%
Cost (£M)		0.49	1.92	0.81	1.21	2.68	0.79	0.60
Contribution to Cost Ratio		**184**	42	68	70	19	82	92

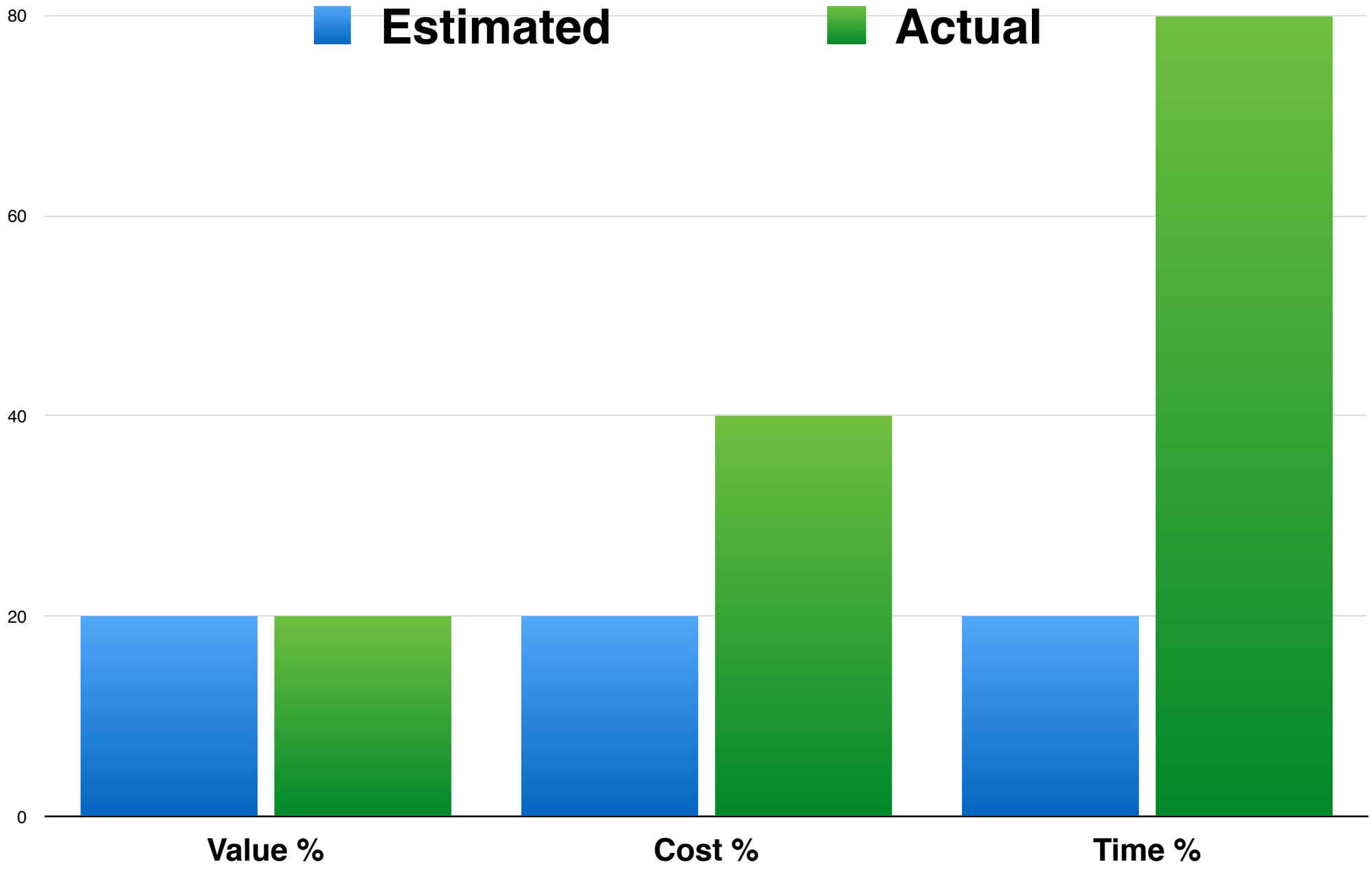
4 Dynamic Decision Making: learning fast, committing late

Estimating the Power of suggested architecture together with related costs



20% Snapshot:
Design to Cost Dynamically.

The point being that unexpected residual resources may force you to choose unexpectedly different architecture, in order to achieve deadline and budget



25 Balls in The Air: Concurrent Engineering

EVO Plan Confirmit 8.5 in **Evo Step Impact *Measurement***

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

9

8

3

3

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	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
			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		7	7
			Runtime.ResourceUsage.CPU		
3,0	35,0	97,2	38	3	2
			Runtime.ResourceUsage.MemoryLeak		
0,0	800,0	100,0	800	0	0
			Runtime.Concurrency (number of users)		
1350,0	1100,0	146,7	150	500	1000
			Development resources		
64,0			0		84

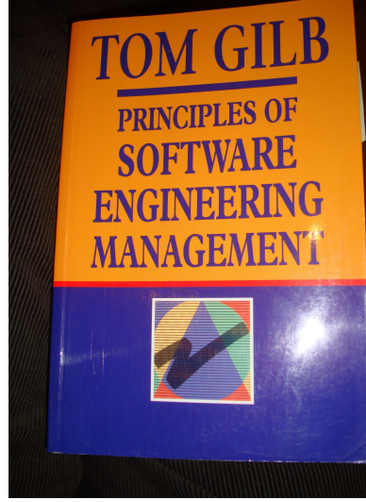
Current Status	Improvements		Reportal - MR Features		
Units	Units	%	Past	Tolerable	Goal
			Usability.Replacability (feature count)		
1,0	1,0	50,0	14	13	12
			Usability.Productivity (minutes)		
20,0	45,0	112,5	65	35	25
			Usability.ClientAcceptance (features count)		
4,4	4,4	36,7	0	4	12
			Development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

Computing Real Time Priority

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	35,0	97,2	38	3	2
			Runtime.ResourceUsage.MemoryLeak		
0,0	800,0	100,0	800	0	0
			Runtime.Concurrency (number of users)		
1350,0	1100,0	146,7	150	500	1000
			Development resources		
64,0			0		84

Quinnan: IBM FSD Cleanroom
Dynamic Design to Cost



Quinnan describes the process control loop used by IBM FSD to ensure that cost targets are met.

'Cost management. . . yields valid cost plans linked to technical performance. Our practice carries cost management farther by introducing design-to-cost guidance. Design, development, and managerial practices are applied in an integrated way to ensure that software technical management is consistent with cost management. The method [illustrated in this book by Figure 7.10] consists of developing a design, estimating its cost, and ensuring that the design is cost-effective.' (p. 473)

He goes on to describe a design iteration process trying to meet cost targets by either redesign or by sacrificing 'planned capability.' When a satisfactory design at cost target is achieved for a single increment, the 'development of each increment can proceed concurrently with the program design of the others.'

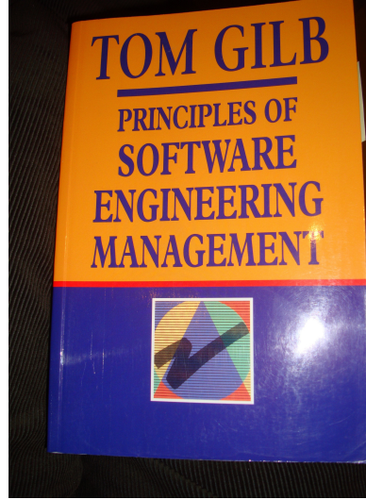
'Design is an iterative process in which each design level is a refinement of the previous level.' (p. 474)

It is clear from this that they avoid the big bang cost estimation approach. Not only do they iterate in seeking the appropriate balance between cost and design for a single increment, but they iterate through a series of increments, thus reducing the complexity of the task, and increasing the probability of learning from experience, won as each increment develops, and as the true cost of the increment becomes a fact.

'When the development and test of an increment are complete, an estimate to complete the remaining increments is computed.' (p. 474)

Source: Robert E. Quinnan, 'Software Engineering Management Practices', IBM Systems Journal, Vol. 19, No. 4, 1980, pp. 466~77
This text is cut from Gilb: The Principles of Software Engineering Management, 1988

Quinnan: IBM FSD Cleanroom
Dynamic Design to Cost



Quinnan describes the process control loop used by IBM FSD to ensure that cost targets are met.

'Cost management. ...s cost management farther by introducing desi... lied in an integrated way to ensure that softwar... lustrated in this book by Figure 7.10] consis... cost-effective.' (p. 473)

He goes on to ...sign or by sacrificing 'planned capability. ... the 'development of each increment can proc...

'Design is an iterati... p. 474)

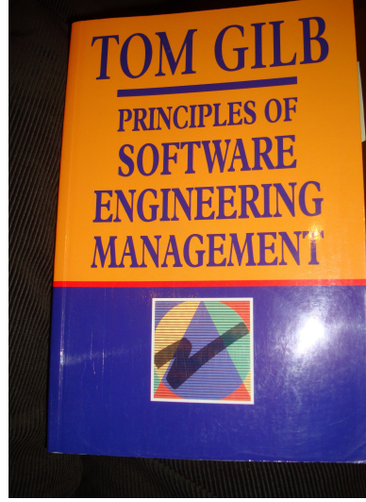
of developing a design, estimating its cost, and ensuring that the design is cost-effective

It is clear from this that they avoid the big bang cost estimation approach. Not only do they iterate in seeking the appropriate balance between cost and design for a single increment, but they iterate through a series of increments, thus reducing the complexity of the task, and increasing the probability of learning from experience, won as each increment develops, and as the true cost of the increment becomes a fact.

'When the development and test of an increment are complete, an estimate to complete the remaining increments is computed.' (p. 474)

Source: Robert E. Quinnan, 'Software Engineering Management Practices', IBM Systems Journal, Vol. 19, No. 4, 1980, pp. 466~77
This text is cut from Gilb: The Principles of Software Engineering Management, 1988

Quinnan: IBM FSD Cleanroom
Dynamic Design to Cost



Quinnan describes the process control loop used by IBM FSD to ensure that cost targets are met.

'Cost management. . . yields valid cost plans linked to technical performance. Our practice carries cost management farther by introducing design-to-cost guidance. Design, development, and managerial practices are applied in an integrated way to ensure that software technical management is consistent with cost management. The method [illustrated in this book by Figure 7.10] consists of developing a design, estimating its cost, and ensuring that the design is cost-effective.' (p. 473)

He goes on to describe a design iteration process trying to meet cost targets by either redesign or by sacrificing 'planned capability.' When a satisfactory design at cost target is achieved for a single increment, the 'development of each increment can proc

'Design is an iterati **p. 474)**

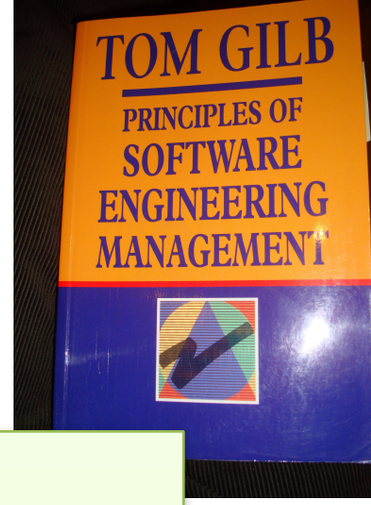
It is clear from **iterate in seeking the appropriate balanc** **eries of increments, thus reducing the compl** **won as each increment develops, and as th**

'When the developr **ining increments is computed.' (p. 474)**

Source: Robert E. Qu **9, No. 4, 1980, pp. 466~77**
This text is cut from

iteration process trying to meet cost targets by either *redesign* or by *sacrificing* 'planned capability'

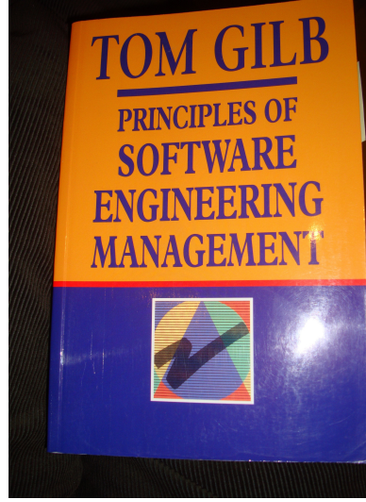
Quinnan: IBM FSD Cleanroom
Dynamic Design to Cost



Design is an

iterative process

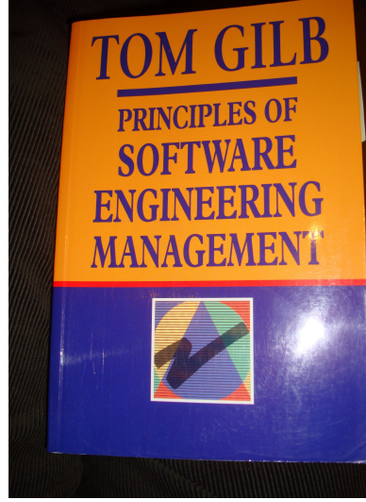
Quinnan: IBM FSD Cleanroom
Dynamic Design to Cost



Quinnan describes the process control loop used by IBM FSD to ensure that cost targets are met.

but they iterate through a series of increments,
thus *reducing the complexity of the task*,
and *increasing the probability of learning from experience*

Quinnan: IBM FSD Cleanroom
Dynamic Design to Cost



Quinnan describes the process control loop used by IBM FSD to ensure that cost targets are met.

an estimate to complete the remaining increments is computed.

This text is cut from Gilb: The Principles of Software Engineering Management, 1988

A story of devs
refusing to be told how to design
by Bank IT architects. Focussing
on a few critical value measurable
Objectives;
and delivering **on time** for **full**
user satisfaction: 100%
success
Using Agile Evo: The Engineering
Agile Method

Richard Smith

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

Previous IT Project Management Methods: No ‘Value delivery tracking’. No change reaction ability

Richard Smith

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

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

Richard Smith

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

The proof is in the pudding;

Richard Smith

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

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

Richard Smith

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

" I attended a 3-day course with you and Kai whilst at Citigroup in 2006", Richard Smith

Dynamic (Agile, Evo) design testing: not unlike 'Lean Startup'

Richard Smith

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

" I attended a 3-day course with you and Kai whilst at Citigroup in 2006", Richard Smith

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

Richard Smith

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

" I attended a 3-day course with you and Kai whilst at Citigroup in 2006" , Richard Smith

5 Delegation of Decision Making: to where the action and competence is placed.

How?

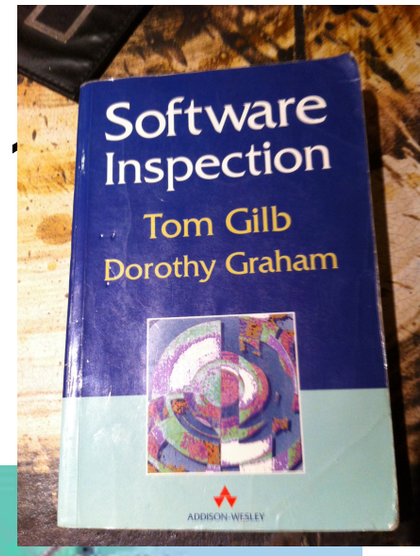
- Make *<u>developers</u>* **responsible**
  - **for delivery of the 'quantified' critical requirements**
    - (Performance, Qualities, cost, deadline)
- Give them the **freedom to decide the right designs**
  - With immediate responsibility to *measure* that they are delivering the results
- Get the 'unprofessional' users and **customers 'off their backs'**
  - Avoid receiving features and stories
    - which are usually amateur design, by people who have no overview or responsibility or design ability (users and customers, and managers)
- **Elevate your talent** by becoming a **real 'software ENGINEER'**
  - With coding-expert craftsmanship, as your basic talent

Background 1970-1980
MANAGERS FAIL

- **Michael Fagan and Ron Radice co-invent 'Software Inspection'**
  - **The intent was to collect data on bugs and defects**
  - **Use it to find frequent common causes**
  - **To improve development processes**
  - **The attitude was explicitly**
    - **'managers should manage' (MEF to TsG)**
  - **THEY FAILED TO GET REAL PROCESS IMPROVEMENT**

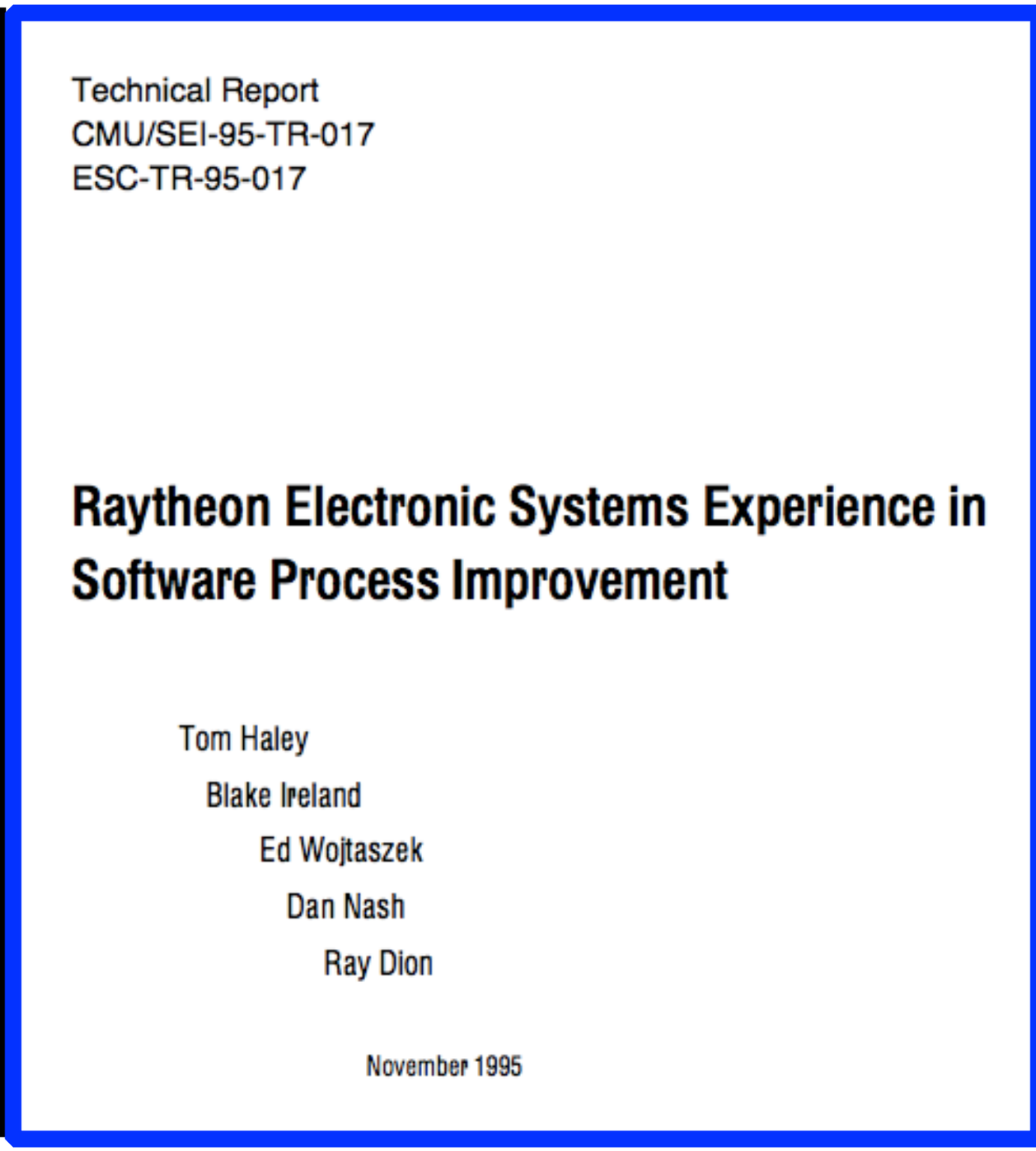
1980
The ‘Troops’ succeed, where the Generals Failed

- Robert Mays and Carol L. Jones, at IBM Research Triangle Park, NC
- Invent ‘Defect Prevention Process’ → Ch
- Major idea:
  - Delegate power to devs to
    - Analyze their OWN defects
    - And fix their OWN process
- THAT *WORKED*



Software Process Improvement at Raytheon

- Source : Raytheon Report 1995
  - http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=12403 (this is a header to the download) Tested May 2014
  - Search “Dion & Raytheon” (Dion is Florida retired in 2014)
  - http://resources.sei.cmu.edu/asset_files/TechnicalReport/1995_005_001_16415.pdf
- An excellent example of process improvement driven by measurement of improvement
- Main Motor:
  - “Document Inspection”, Defect Detection
- Main Driver:
  - “Defect Prevention Process” (DPP)



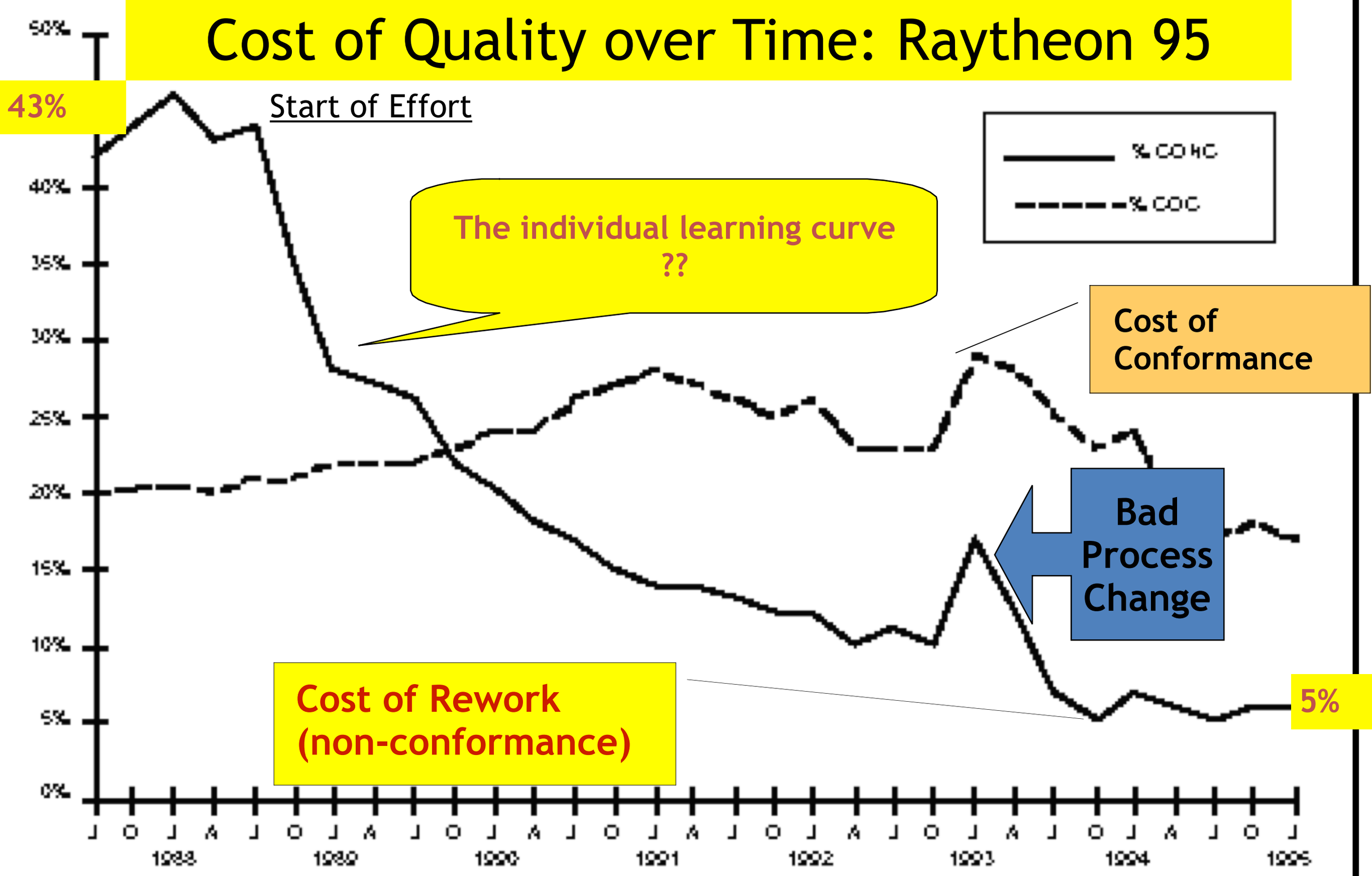
Technical Report
CMU/SEI-95-TR-017
ESC-TR-95-017

Raytheon Electronic Systems Experience in Software Process Improvement

Tom Haley
Blake Ireland
Ed Wojtaszek
Dan Nash
Ray Dion

November 1995

Cost of Quality over Time: Raytheon 95



End 1988

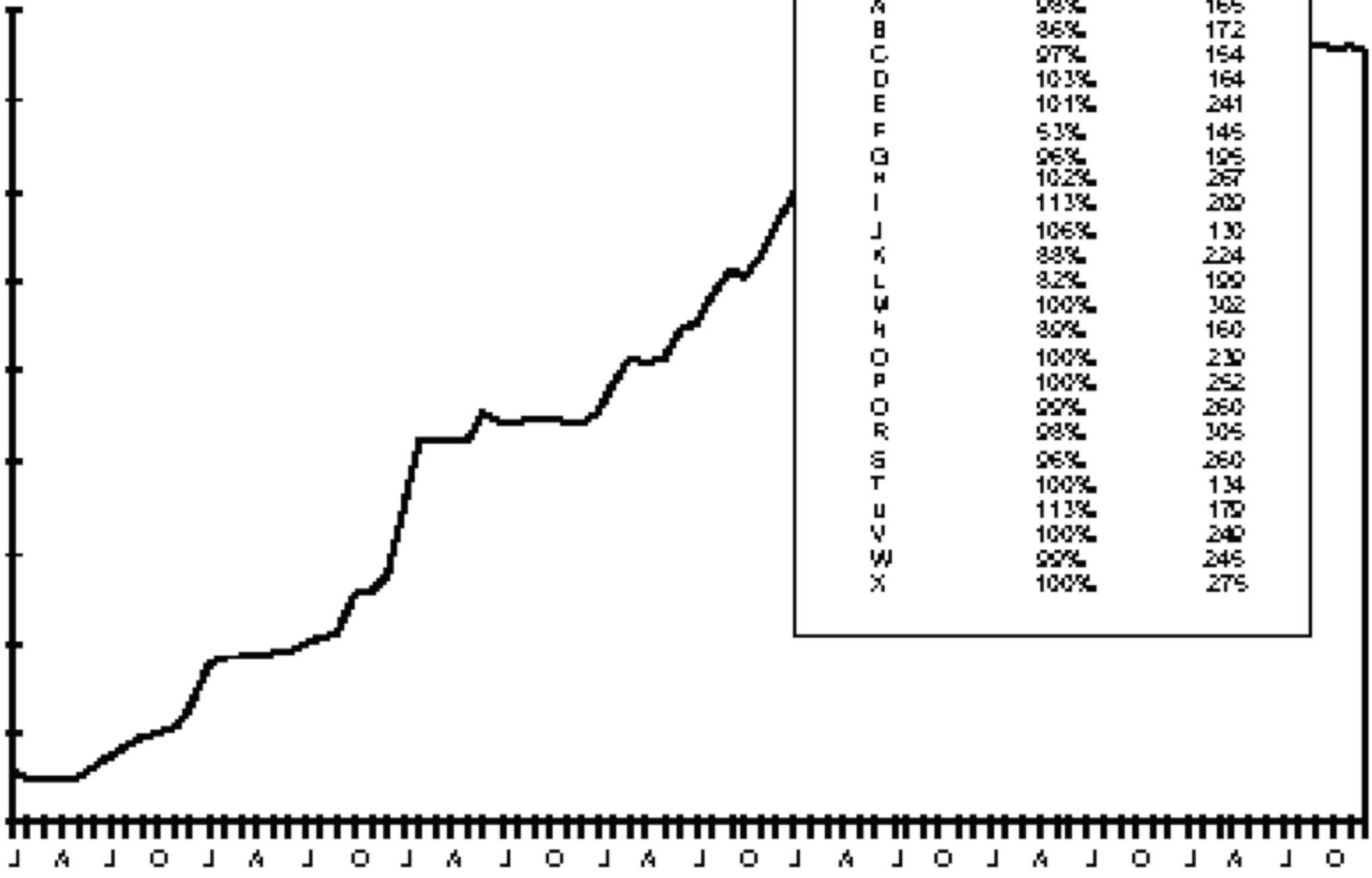
End 1994

Figure 8: Cost of Quality Versus Time

Raytheon 95 Software Productivity 2.7X better

Productivity

Project	CA/CA'Bu D	Productivity
A	98%	165
B	86%	172
C	97%	154
D	103%	164
E	101%	241
F	53%	145
G	96%	195
H	102%	267
I	113%	200
J	106%	130
K	88%	224
L	82%	199
M	100%	302
N	80%	160
O	100%	230
P	100%	252
Q	99%	260
R	98%	305
S	96%	260
T	100%	134
U	113%	170
V	100%	240
W	99%	245
X	100%	275



+

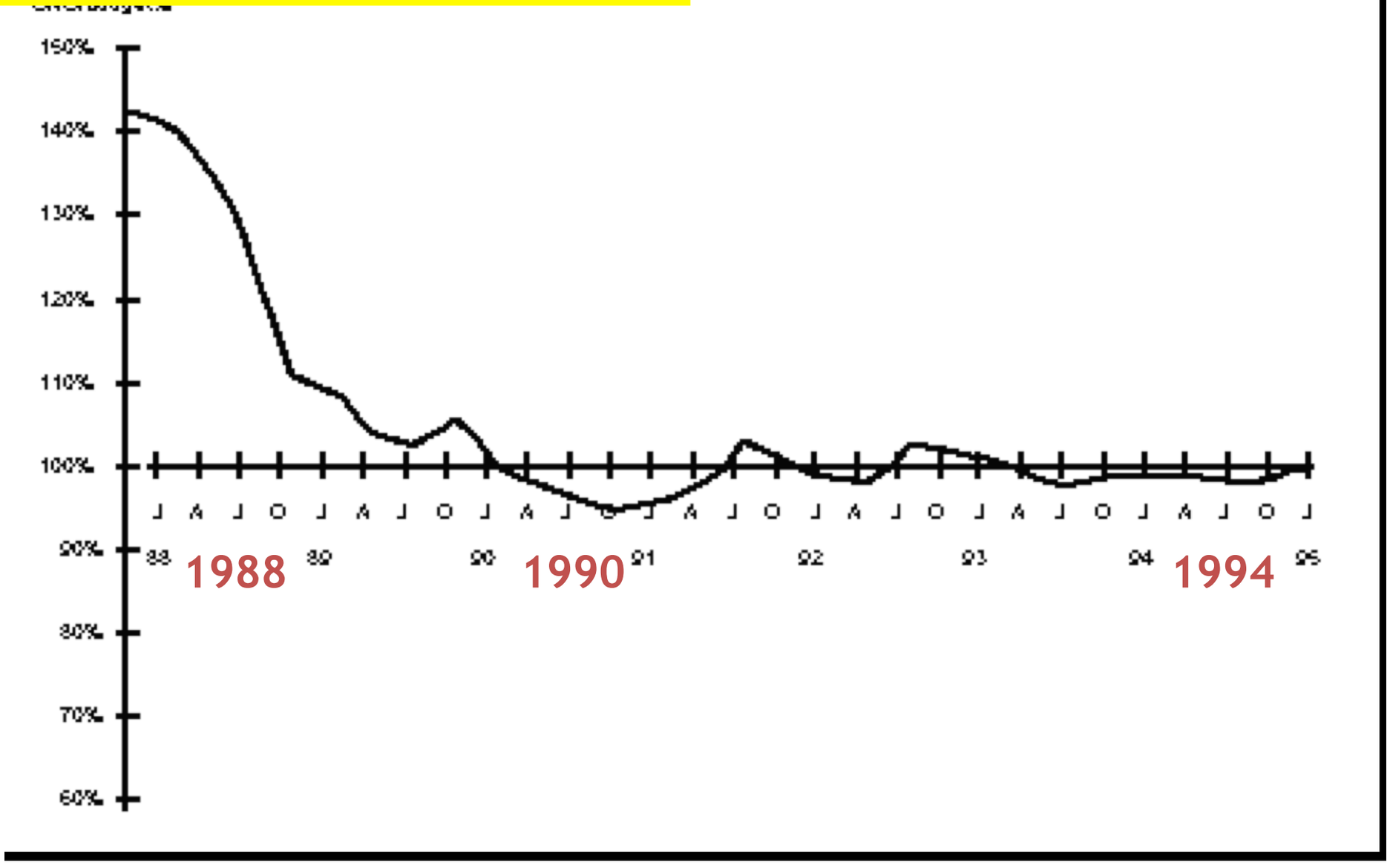
170%

Achieving Project Predictability: Raytheon 95

Cost At Completion / Budget %

140%

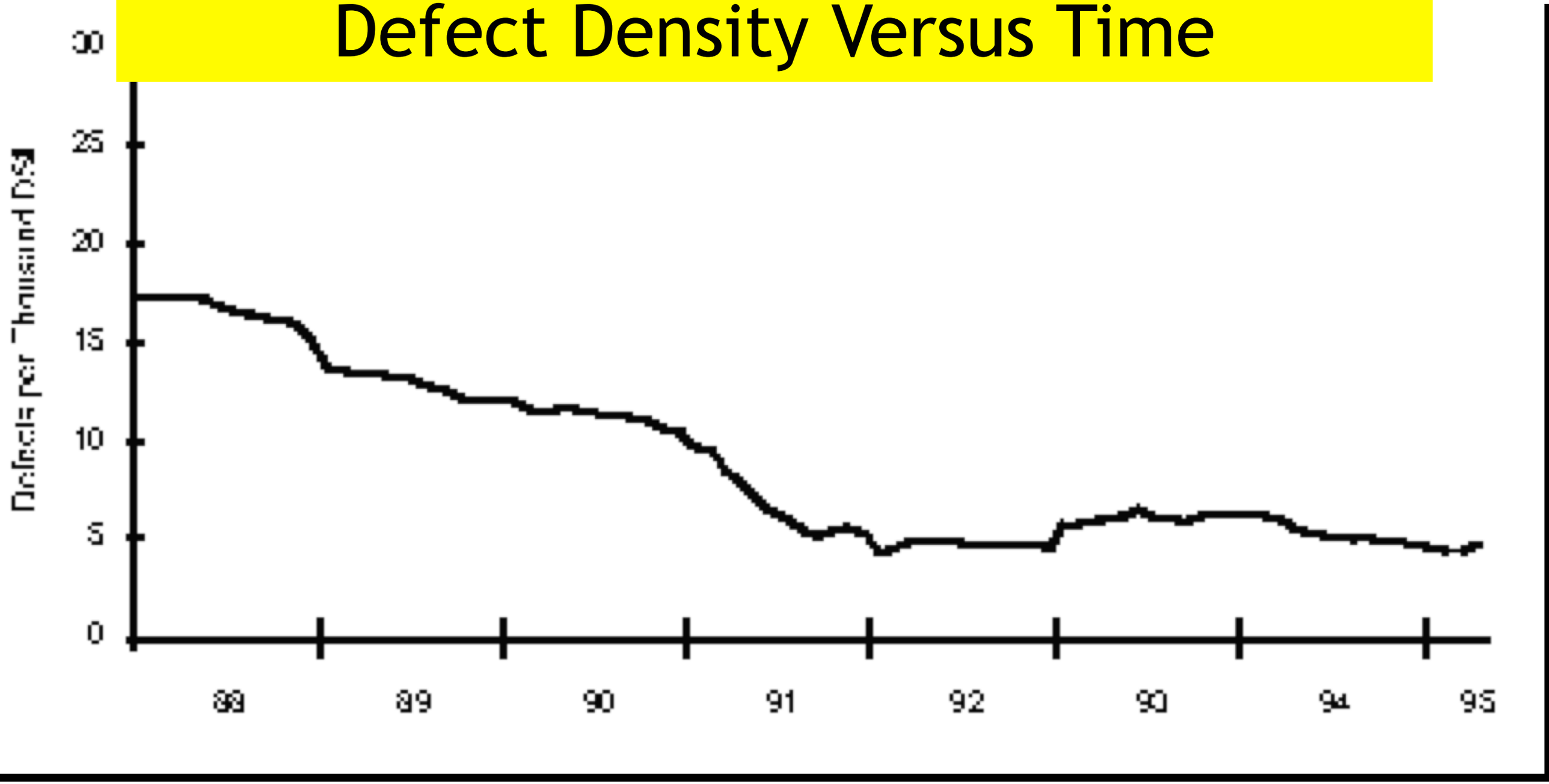
100%



Examples of Process Improvements: Raytheon 95

- **Process Improvements Made**
- **Erroneous interfaces during integration and test** -
  - Increased the detail required for interface design during the requirements analysis phase and preliminary design phase - Increased thoroughness of inspections of interface specifications
- **Lack of regression test repeatability -**
  - Automated testing - Standardized the tool set for automated testing - Increased frequency of regression testing
- **Inconsistent inspection process -**
  - Established control limits that are monitored by project teams - Trained project teams in the use of statistical process control - Continually analyze the inspection data for trends at the organisation level
- **Late requirements up-dates** -
  - Improved the tool set for maintaining requirements traceability - Confirm the requirements mapping at each process phase
- **Unplanned growth of functionality during Requirements Analysis**
  - - Improved the monitoring of the evolving specifications against the customer baseline - Continually map the requirements to the functional proposal baseline to identify changes in addition to the passive monitoring of code growth - Improved requirements, design, cost, and schedule tradeoffs to reduce impacts

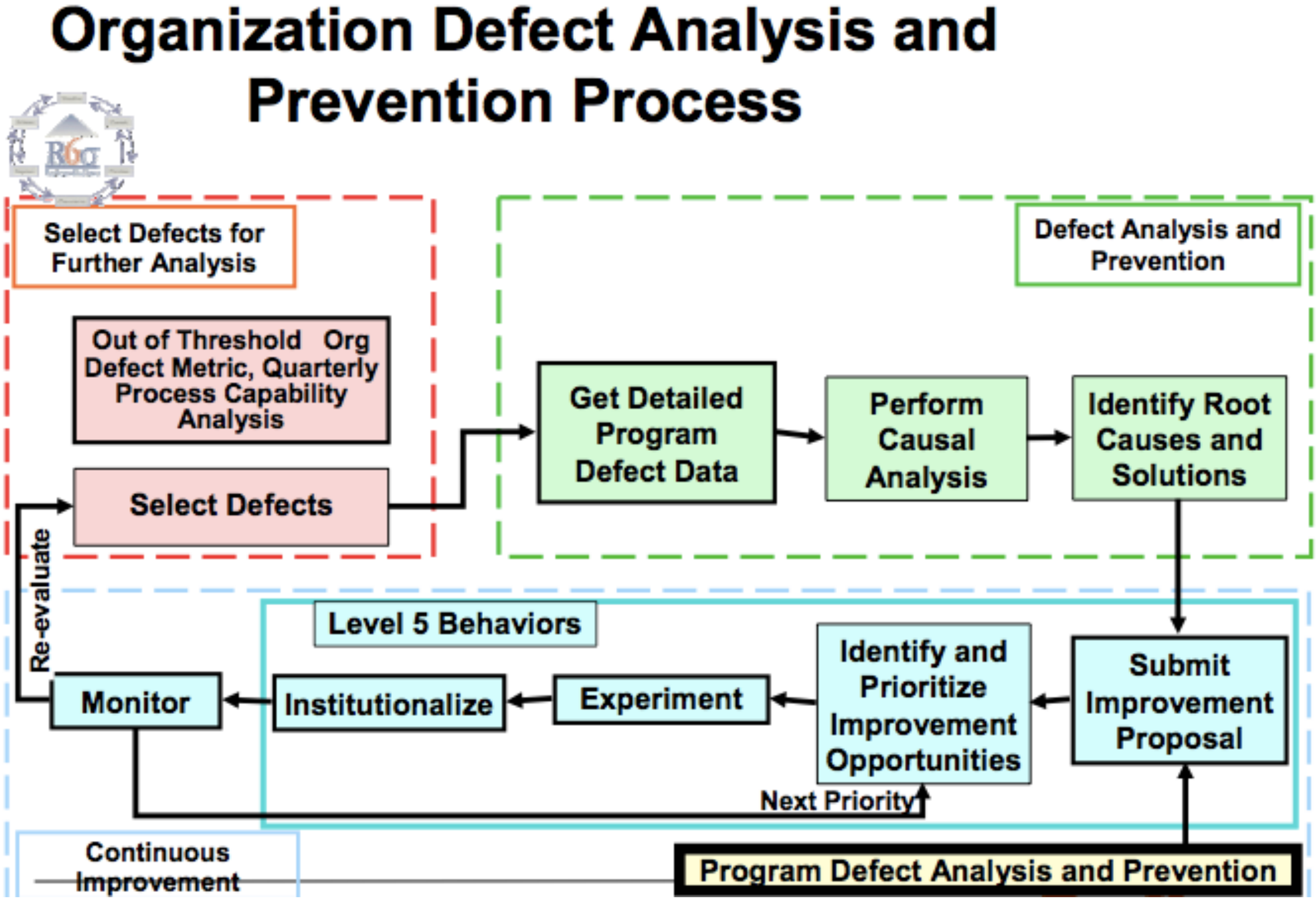
Overall Product Quality: Raytheon 95
(Bug density going down by 3:1)
Defect Density Versus Time



Return On Investment

- $7.70 per $1 invested at Raytheon
- Sell your improvement program to top management on this basis
- Set a concrete target for it
  - PLAN [Our Division, 2 years hence] 8 to 1

The DPP Process



What's Going on Here?

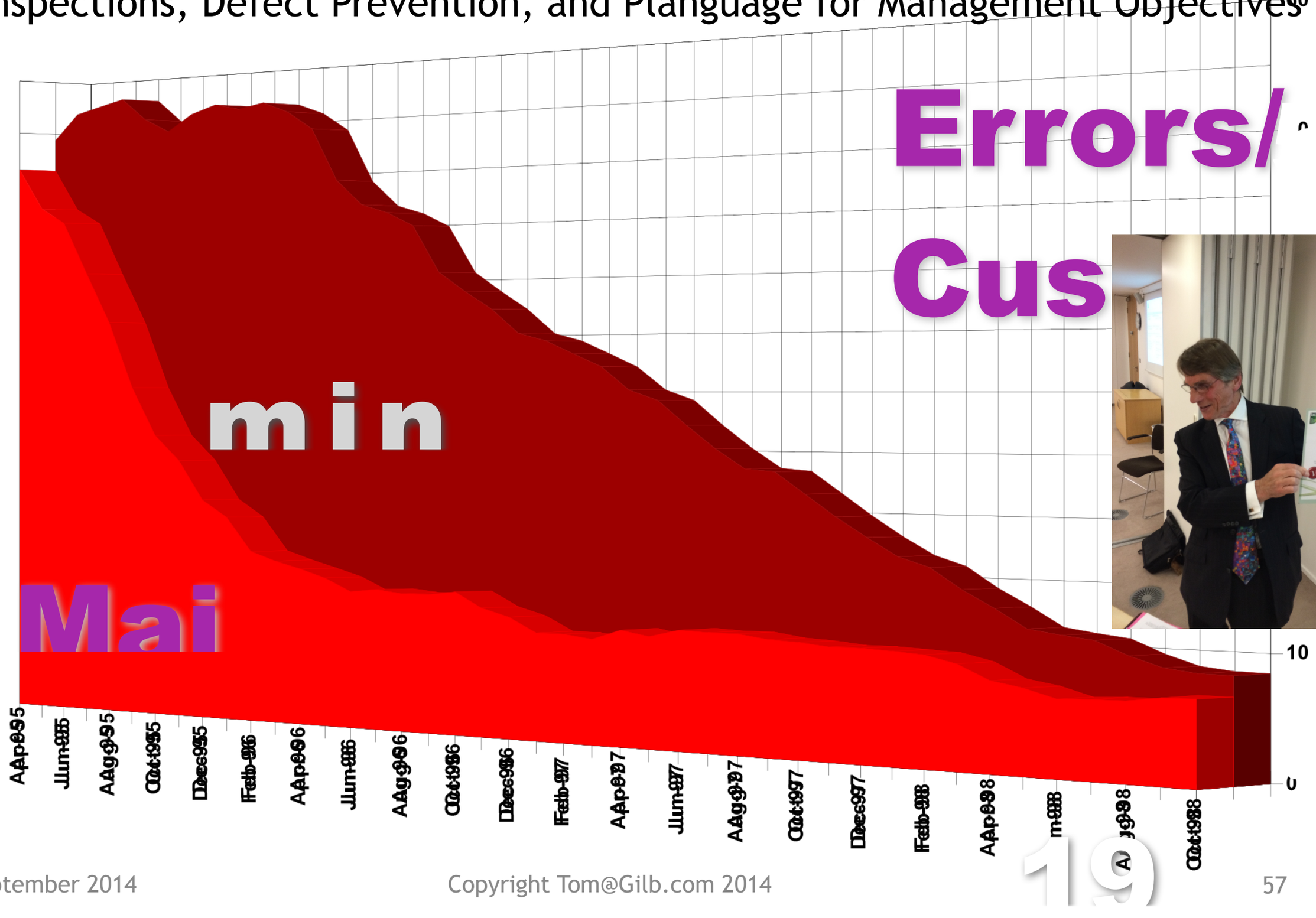
- 1,000 programmers
  - Later joined by 1,000 merged new programmers
  - Are
    - Analyzing their **own** bugs and spec defects
    - Suggesting their **own** work environment changes
    - And reducing their 43% rework by 10 X
- Power has been delegated to the programmers

Improving the *Reliability* Attribute

Primark, London (Gilb Client)

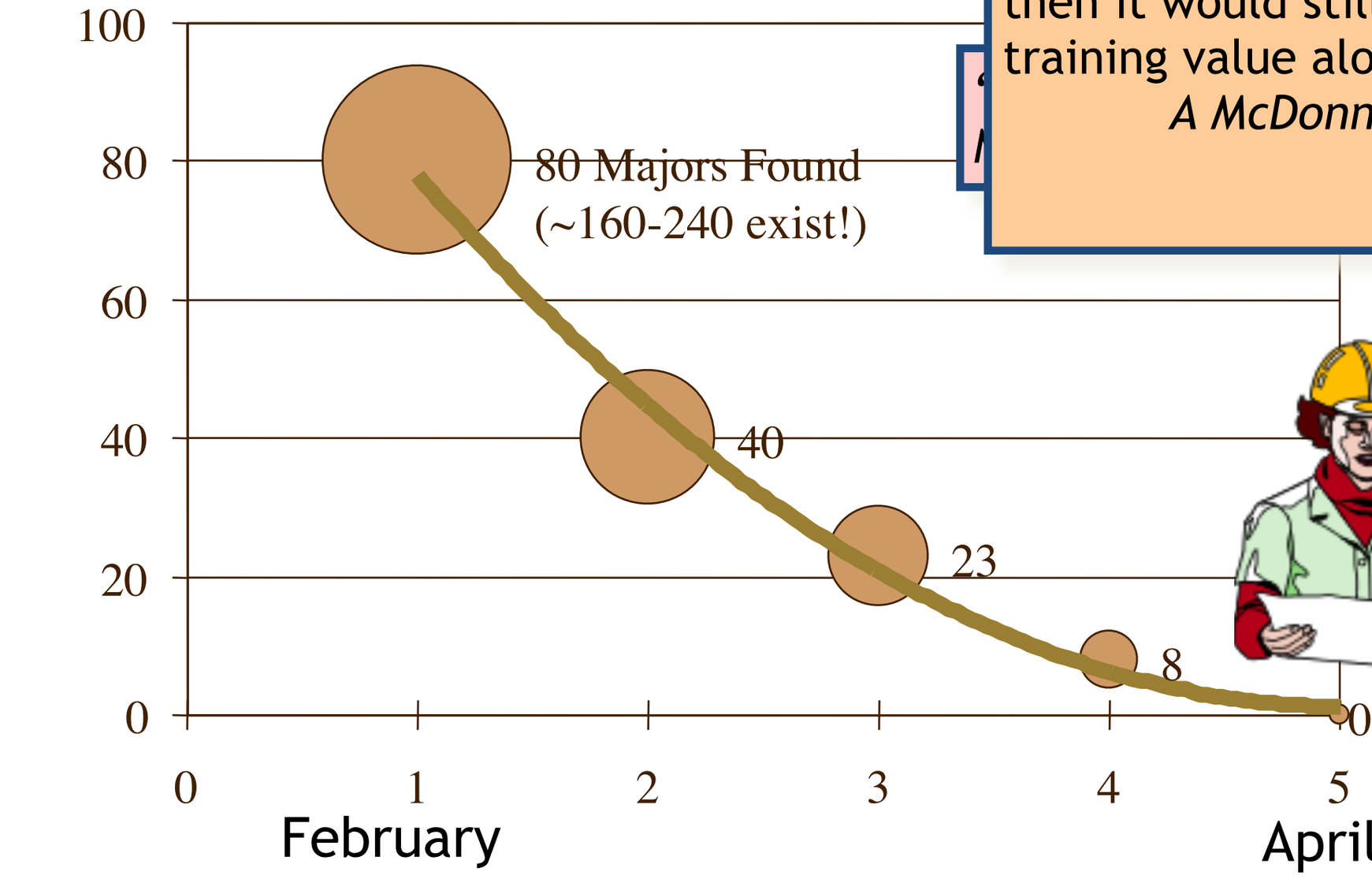
see case study Dick Holland, "Agent of Change" from Gilb.com

Using, Inspections, Defect Prevention, and Planguage for Management Objectives



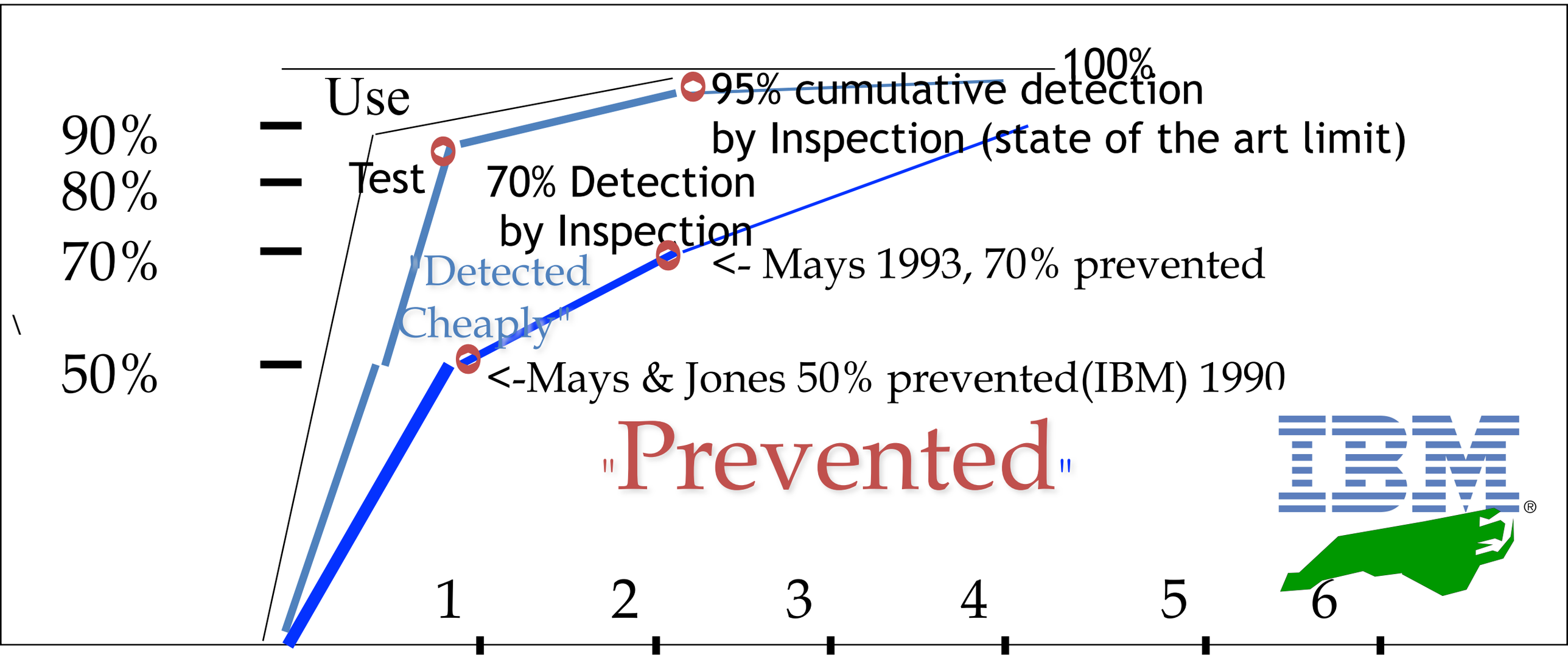
Positive Motivation
Personal Improvement

> "We find an hour of doing Inspection is worth ten hours of company classroom training."
> *A McDonnell-Douglas line manager*
> "Even if Inspection did not have all the other measurable quality and cost benefits which we are finding, then it would still pay off for the training value alone."
> *A McDonnellDouglas Director*



Inspections of Gary's Designs

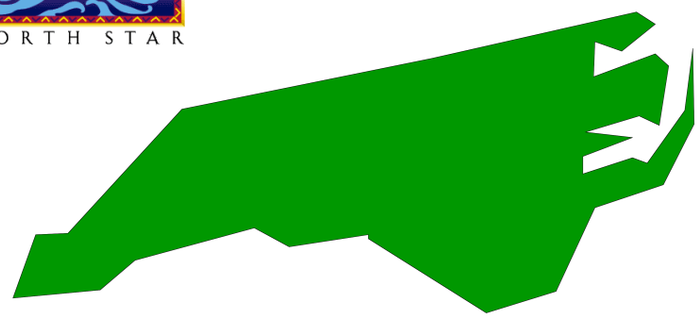
Prevention + Pre-test Detection is the most effective and efficient



- Prevention data based on state of the art prevention experiences (IBM RTP), Others (Space Shuttle IBM SJ 1-95) 95%+ (99.99% in Fixes)
- Cumulative Inspection detection data based on state of the art Inspection (in an environment where prevention is also being used, IBM MN, Sema UK, IBM UK)

IBM MN & NC DP Experience

- **2162 DPP Actions implemented**
  - between Dec. 91 and May 1993 (30 months)<-Kan
- RTP about 182 per year for 200 people.<-Mays 1995
  - 1822 suggested ten years (85-94)
  - 175 test related
- RTP 227 person org<- Mays slides
  - 130 actions (@ 0.5 work-years
  - 34 causal analysis meetings @ 0.2 work-years
  - 19 action team meetings @ 0.1work-years
  - Kickoff meeting @ 0.1 work-years
  - TOTAL costs 1% of org. resources



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

The ICL Bill of Rights for Company Communication (by TsG)

1. You have a right to know precisely what is expected of you.

2. You have a right to clarify things with colleagues, anywhere in the organization.

3. You have a right to initiate clearer definitions of objectives and strategies.

4. You have a right to get objectives presented in measurable, quantified formats.

5. You have a right to change your objectives and strategies, for better performance.

6. You have the right to try out new ideas for improving communication.

007. You have the right to fail when trying, but also to kill failures quickly.

8. You have a right to constructively challenge higher-level objectives and strategies.

9. You have a right to be judged objectively on your performance against measurable objectives.

10. You have a right to offer constructive help to colleagues to improve communication.

Summary DPP
Managers: 0 Devs : 1

- Developers are *better* at managing their own work environment, than their managers are

- ‘Directors’ should NOT design the work environment

- Developers should ‘evolve the environment’
  - through practical deep personal insights,
  - and take responsibility for their own work situation

Case: Delegating Software product design to the Developers

"In the interest of overcoming my reluctance to delegate, starting Monday I want you to do all of my worrying for me."

We gave them a 1 day briefing on our Evo method and Planguage

That's all they needed to succeed!

They were Real engineers

Customer Successes in Corporate Sector

Real Example of 1 of the 25 Quality Requirements

Usability.Productivity:

Scale for quantification: Time in minutes to set up a typical specified Market Research-report

Past Level [Release 8.0]: 65 mins.,

Tolerable Limit [Release 8.5]: 35 mins.,

Goal [Release 8.5]: 25 mins.

Trond Johansen

Shift: from Function to Quality

- **Our new focus is on the daily operations of our Market Research users,**
  - **<u>not</u> a list of features. that they might or might not like. 50% never used!**
  -
  - **We KNOW that increased efficiency, which leads to more profit, will please them.**
  - **The ‘45 minutes actually saved x thousands of customer reports’**
    - **= big $$$ saved**
- **After one week we had defined more or less all the requirements for the next version (8.5) of Confirmit.**

Quantified Value Delivery Project Management in a Nutshell

Quantified Value Requirements, Design, Design Value/cost estimation, Measurement of Value Delivery, Incremental Project Progress to Date

		Current Status	Improvements		Goals			Step9 Recoding			
								Estimated impact		Actual impact	
		Units	Units	%	Past	Tolerable	Goal	Units	%	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)						
15		20,00	45,0	112,5	65	35	25	20,00	50,00	38,00	95,00
					Development resources						
1			101,0	91,8	0		110	4,00	3,64	4,00	3,64

Prior

ity

Next

Cumul

ative

weekly

Constra

Terra

Estim

Week

Every user, every day, was using an average of 65 minutes to set up a report

Usability.Productivity

Scale for quantification: Time in minutes to set up a typical specified Market Research-report

Past Level [Release 8.0]: 65 mins.,

Tolerable Limit [Release 8.5]: 35 mins.,

Goal [Release 8.5]: 25 mins.

							G	BX	BY	BZ	CA
1											
2								Step9			
3								Recoding			
4								Estimated impact		Actual impact	
5							al	Units	%	Units	%
6											
7							0				
8											
9							5				
10							5				
11		0,00	0,0	0,0		0	30	10				
12					Usability.In eness (%)							
13		0,00	0,0	0,0		60	80				
14					Usability.Pro uctivity (minutes)						
15		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

The worst acceptable case requirement, for the next quarterly world release, is 35 minutes, or better; less is 'intolerable'

Usability.Productivity
Scale for quantification: Time in minutes to set up a typical specified Market Research-report

Past Level [Release 8.0]: 65 mins.

Tolerable Limit [Release 8.5]: 35 mins.,

Goal [Release 8.5]: 25 mins.

							BX	BY	BZ	CA		
1												
2							Step9					
3							Recoding					
4							Estimated impact		Actual impact			
5							Units	%	Units	%		
6												
7						0						
8												
9						5						
10						5						
11						10						
12				Usability.Intuitiveness (%)								
13		0,00	0,0	0,0	0	80						
14				Usability.Productivity (minutes)								
15		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	

The committed target level requirement, the 'Goal', is to get the user task down to 25 minutes or better.

Usability.Productivity
Scale for quantification: Time in minutes to set up a typical specified Market Research-report

Past Level [Release 8.0]: 65 mins.,

Tolerable Limit [Release 8.5]: 35 mins.,

Goal [Release 8.5]: 25 mins.

	A	B	C							Z	CA
1											
2		Current Status	Imp								
3											
4										ctual impact	
5		Units	Units							ts	%
6											
7		1,00									
8											
9		5,00									
10		10,00	1								
11		0,00									
12											
13		0,00									
14											
15		20,00	45,0	112,5	65	35	25	20,00	30,00	38,00	95,00
20					Development resources						
21			101,0	91,8	0		110	4,00	3,64	4,00	3,64

The weekly ‘value delivery cycle’ resource is 110 work-hours (4 days, effective time for the team of 3 to 4 people)

Work Hours available this weekly delivery cycle. For 4 people. 110 effective hours

		Current Status	Improvem					Step9 Recoding			
								Estimated impact		Actual impact	
		Units	Units					Units	%	Units	%
		1,00	1,0								
		5,00	5,0								
		10,00	10,0								
		0,00	0,0	0,0	0	30	10				
					Usability.Intuitiveness (%)						
		0,00	0,0	0,0	0	60					
					Usability.Productivity (minutes)						
		20,00	45,0	112,5	65	35		20,00	50,00	38,00	95,00
					Development resources						
			101,0	91,8	0		110	4,00	3,64	4,00	3,64

The developer team can choose the requirement they want to prioritize, and work on, this week. They chose the 0.0 (no improvement yet, in last 8 weeks) of the 'Productivity requirement

	A	B	C	D	E	F	G	BX	BY	BZ	CA
1											
2		Current Status	Improven	nts							
3											
4											l impact
5		Units	Units	%	Past						%
6					Usabili						
7		1,00	1,0	50,0							
8					Usabili						
9		5,00	5,0	100,0							
10		10,00	10,0	200,0							
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)						
15		20,00	45,0	0.0	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

The team chooses to work on a weak point.
This is 'dynamic prioritization' – Decisions based on the weekly 'state of play'

Every user, every day, was using an average of 65 minutes to set up a report. We want a 40 minute improvement to that, to 25 minutes

Usability.Productivity

Scale for quantification: Time in minutes to set up a typical specified Market Research-report

Past Level [Release 8.0]: 65 mins.,

Tolerable Limit [Release 8.5]: 35 mins.,

Goal [Release 8.5]: 25 mins.

							BX	BY	BZ	CA
1										
2							Step9			
3							Recoding			
4							Estimated impact		Actual impact	
5							Units	%	Units	%
6										
7										
8										
10										
11										
12										
13										
14					Usability.Productivity (minutes)					
15		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

The team has a 30 minute ‘design’ meeting, to suggest designs which might help move from 65 minutes for the task, towards the 25 minute Goal level

	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	Units	%	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)						
15		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

‘Recoding’ is the name of 1 of 12 suggested, brainstormed, designs for saving user effort, by any member of the developer team

	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	Units	%	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)						
15		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

‘Recoding’ was estimated, by the suggester, to save 20 minutes time for the users

	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	Units	%	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)						
15		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

‘Recoding’ was also estimated to take the entire 4 day delivery cycle available. No time left to add more solutions, in order to try to get closer to the target, on this delivery cycle.

	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	Units	%	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)						
15		20,00	45,0	112,5	35	35	25	20,0	50,00	38,00	95,00
20					Development resources						
21			101,0	91,8	0			4,00	3,64	4,00	3,64

And 20 minutes saving, was the best 'impact' estimated from the 12 total suggestions made by the team members. So 'Recoding' (of marketing codes) was chosen as the best thing to do that week.

	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	Units	%	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)						
15		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

And 20 minutes saving, is equivalent to 50% of the way betweem Past and Goal (65 – 25 = 40, 20/40 = 50%).

This is another way of expressing the expected impact of Recoding

		Current Status	Improvements		Goals			Step9			
								Recoding			
								Estimated impact		Actual impact	
		Units	Units	%	Past	Tolerable	Goal	Units	%	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)						
15		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

The team commits to the 'Recoding' solution. They code, test and handover to Microsoft usability Labs in Washington State, who volunteered to independently measure all the Usability designs.

	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	Units	%	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)						
15		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

The result was a saving, or improvement of 38 minutes, or 95% of the way to the target requirement of 25 minutes

	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	Units	%	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)						
15		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

This was not good enough for Trond Johansen.

And he did not want to use 1 of the 3 remaining weeks to release (10, 11, 12th weeks) in order to get to 100% of the target.

So, he asked one team member to spend the weekend tuning the 'Recoding' solution.

And he managed to get the timing down to 20 minutes.

12.5% more than the 25 minutes targeted.

Thus total impact is 112.5%

	A	B	C	D	E	F	G	BX	BY	BZ	CA
1											
2		Current Status	Improvement		Goals			Step9			
3								Recoding			
4								Estimated impact		Actual impact	
5		Units	Units	%	Past	Tolerable	Goal	Units	%	Units	%
6					Usability.Replacability (feature count)						
7		1,00	1,0		2	1	0				
8					Usability.Speed.NewFeaturesImpact (%)						
9		5,00	5,0		0	15	5				
10		10,00	10,0		0	15	5				
11		0,00	0,0		0	30	10				
12					Usability.Intuitiveness (%)						
13		0,00	0,0		0	60	80				
14					Usability.Productivity (minutes)						
15		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

And the priority flag turns Green (no priority, Goal reached)

	A	B	C	D	E	F	G	BX	BY	BZ	CA
1											
2		Current Status	Improvement		Goals			Step9			
3								Recoding			
4								Estimated impact		Actual impact	
5		Units	Units	%	Past	Tolerable	Goal	Units	%	Units	%
6					Usability.Replacability (feature count)						
7		1,00	1,0		2	1	0				
8					Usability.Speed.NewFeaturesImpact (%)						
9		5,00	5,0		0	15	5				
10		10,00	10,0		0	15	5				
11		0,00	0,0		0	30	10				
12					Usability.Intuitiveness (%)						
13		0,00	0,0		0	60	80				
14					Usability.Productivity (minutes)						
15		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

EVO Plan Confirmit 8.5 in **Evo Step Impact *Measurement***

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

9

8

3

3

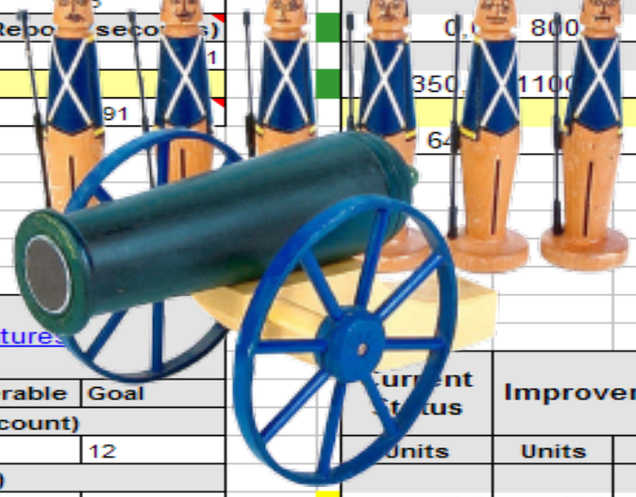
Impact Estimation Table: Reportal codename "Hyggen"

Current Status	Improvements		Reportal - E-SAT features		
Units	Units	%	Past	Tolerable	Goal
			Usability.Intuitivness (%)		
75,0	25,0	62,5	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.ExportRep (min es		
1,0	12,0	150,0	13	13	5
			Usability.ResponseTime.ViewRepo seco s)		
1,0	14,0	100,0	15		1
			Development resources		
203,0			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		
0,0	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



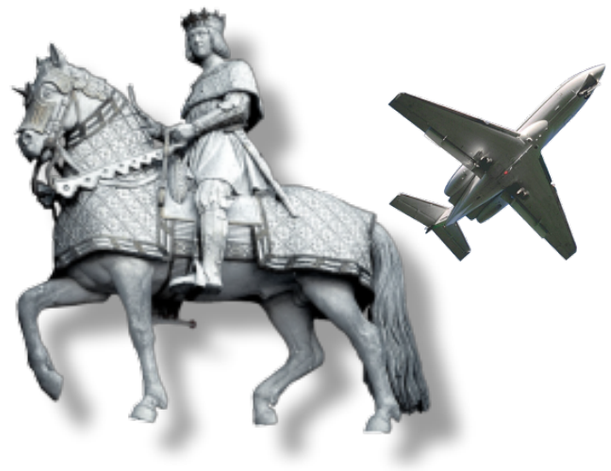
Confirmit Evo Weekly Value Delivery Cycle

	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 1st Qtr

- Only 5 highlights of the 25 impacts are listed here

Description of requirement/work task



confirmit®

Release 8.5

Developers love 'Empowered Creativity'

- **EVO has resulted in**
  - **increased motivation and**
  - **enthusiasm amongst developers,**
  - **it opens up for empowered creativity**

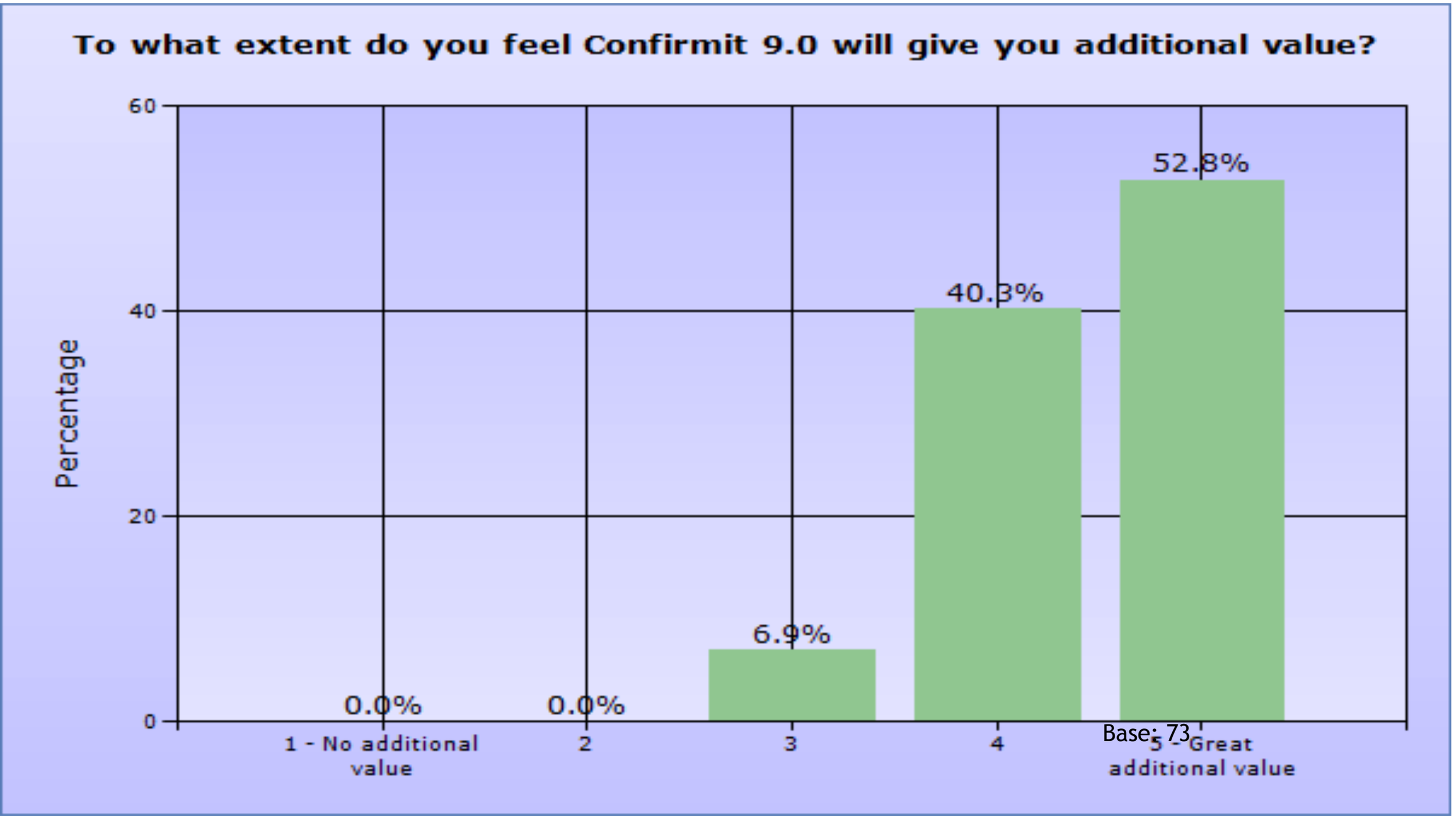
- **Developers**
  - **embraced the method and**
  - **saw the value of using it,**
  - **even though they found parts of Evo difficult to understand and execute (without training)**

Trond Johansen

Initial Customer Feedback on the new Confirmit 9.0

November 24th, 2004

Initial perceived value of the new release (Base 73 people)



Evo's impact on Confirmit 9.0 product qualities Results from the second quarter of using Evo. 1/2

Product quality	Description	Customer value
Intuitiveness	**Probability that an inexperienced user can intuitively figure out how to set up a defined Simple Survey correctly.**	**Probability increased by 175%**
Productivity	**Time in minutes for a defined advanced user, with full knowledge of 9.0 functionality, to set up a defined advanced survey correctly.**	**Time reduced by 38%**

Product quality	Description	Customer value
Productivity	**Time (in minutes) to test a defined survey and identify 4 inserted script errors, starting from when the questionnaire is finished to the time testing is complete and is ready for production. (Defined Survey: Complex survey, 60 questions, comprehensive JScripting.)**	**Time reduced by 83% and error tracking increased by 25%**

Evo's impact on Confirmit 9.0 product qualities Results from the second quarter of using Evo. 2/2

Product quality	Description	Customer value
Performance	Max number of panelists that the system can support without exceeding a defined time for the defined task, with all components of the panel system performing acceptable.	Number of panelists increased by **1500%**
Scalability	Ability to accomplish a bulk-update of X panelists within a timeframe of Z second	Number of panelists increased by **700%**
Performance	Number of responses a database can contain if the generation of a defined table should be run in 5 seconds.	Number of responses increased by **1400%**

Case:
Delegating
Developer Environment
to Developers
using **Multimensional Engineering**

Technical debt

consequences of poor software architecture and software development within a codebase.

-

Causes of technical debt

1. **Business pressures**
2. **Lack of process or understanding**
3. **Lack of building loosely coupled components**,
4. **Lack of test suite**,
5. **Lack of documentation**,
6. **Lack of collaboration**
7. **Parallel**
8. **Delayed Refactoring**

There is a smarter way

- But it means we have to become **real** software *engineers*,

- Not just- - - *softcrafters**

- * coders, developers, programmers.
  - Term coined in
  - “Principles of Software Engineering Management”, 1988, Gilb

Code quality – ”green” week
Empowered Creativity: for Maintainability

- **Instead of Refactoring 1 day a week (failed)**
- **Let the Dev Teams <u>engineer</u> using ‘agile’ (Evo): Design Dev Quality in to their own process**
- **To meeting their own internal stakeholder Quality Objectives**
- **1 week a month**

Current Status		Improvement	Goals			Step 6 (week 14)		Step 7 (wee	
	Units		Past	Tolerable	Goal	Estimated Impact	Actual Impact	Estimated Impact	A
	100,0	100,0	0	80	100			100	
Speed									
	100,0	100,0	0	80	100	100	100		
Maintainability.Doc.Code									
	100,0	100,0	0	80	100	100	100		
InterviewerConsole									
NUnitTests									
	0,0	0,0	0	90	100				
PeerTests									
	100,0	100,0	0	90	100			100	
FxCop									
	0,0	10,0	10	0	0				
TestDirectorTests									
	100,0	100,0	0	90	100			100	
Robustness.Correctness									
	2,0	2,0	0	1	2	2	2		
Robustness.BoundaryConditions									
	0,0	0,0	0	8					
Speed									
	0,0	0,0	0	8					
ResourceUsage.CPU									
	100,0	0,0	100	8					
Maintainability.Doc.Code									
	100,0	100,0	0	8					
SynchronizationStatus									
NUnitTests									

Speed

Maintainability

Nunit Tests

PeerTests

TestDirectorTests

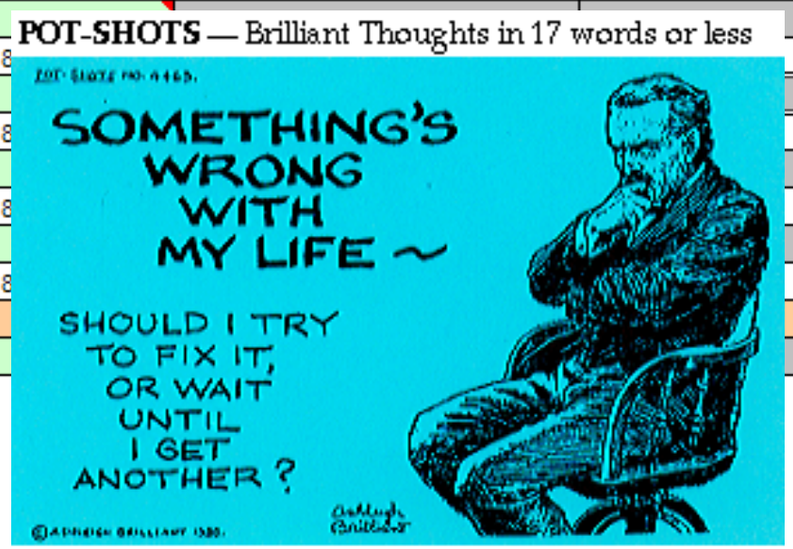
Robustness.Correctness

Robustness.Boundary Conditions

ResourceUsage.CPU

Maintainability.DocCode

SynchronizationStatus



Same Process as for their External (User, Customer) stakeholders

- 1. **define** better quality dev and testing **environment** QUANTITATIVELY
  - Scale of measure and Goal level
- 2. **Figure** out, brainstorm ANY systems engineering design or **architecture** to get to their self determined improvement goals
  - Not just code refactoring, but any tools, processes, motivations, hardware etc that WORK
- 3. **Implement, measure**
  - Keep the stuff that works
  - Dump the stuff that does not MEASURABLY work
- 4. Keep on trucking' (monthly, forever, or …)
  - DONE is when **devs** have no further improvement needs

The Monthly ‘Green Week’

User Week 1

- Select a Goal
- Brainstorm Designs
- Estimate Design Impact/Cost
- Pick best design
- Implement design
- Test design
- Update Progress to Goa

User Week 2

- Select a Goal
- Brainstorm Designs
- Estimate Design Impact/Cost
- Pick best design
- Implement design
- Test design
- Update Progress to Goa

User Week 3

- Select a Goal
- Brainstorm Designs
- Estimate Design Impact/Cost
- Pick best design
- Implement design
- Test design
- Update Progress to Goa

Developer Week 4

- **Select a Goal**
- **Brainstorm Designs**
- **Estimate Design Impact/ Cost**
- **Pick best design**
- **Implement design**
- **Test design**
- **Update Progress to Goal**

Conclusion: Technical Debt

- **Developers**

  Acting like real software engineers

  Can engineer technical debt reduction

It is NOT about refactoring, and patterns

though if they work measurably best, we can use them.

But, did you ever see measurement or re they just belief systems?

It is about mature teams, with common goals, and practical experience, taking charge of their own fate

If management resists, I suggest going on strike!

Why should we suffer agonizing technical debt, wasting 50% or more of our work hours,

Surely we have better things to do!

Cleanroom

In the Cleanroom Method, developed by IBM's Harlan Mil
1970-1980 they reported:
IBM SJ 4/80

- *"Software Engineering began to emerge in FSD" (IBM Federal Systems Division, from 1996 a part of Lockheed Martin Marietta) "some ten years ago [Ed. about 1970] in a continuing evolution that is still underway:*
- *Ten years ago general management expected the worst from software projects – cost overruns, late deliveries, unreliable and incomplete software*
- *Today [Ed. 1980!], management has learned to expect on-time, within budget, deliveries of high-quality software. A Navy helicopter ship system, called LAMPS, provides a recent example. LAMPS software was a four-year project of over 200 person-years of effort, developing over three million, and integrating over seven million words of program and data for eight different processors distributed between a helicopter and* **a ship in 45 incremental deliveries** *[Ed. Note 2%!]s***. Every one of those deliveries was on time and under budget**
- *A more extended example can be found in the NASA space program,*
- *- Where in the past ten years, FSD has managed some 7,000 person-years of software development, developing and integrating over a hundred million bytes of program and data for ground and space processors in over a dozen projects.*
- **- There were few late or overrun deliveries in that decade, and none at all in the past four years."**

In the Cleanroom Method, developed by IBM's Harlan Mills (1980) they report

PERFECT SOFTWARE PROJECTS: by Feedback

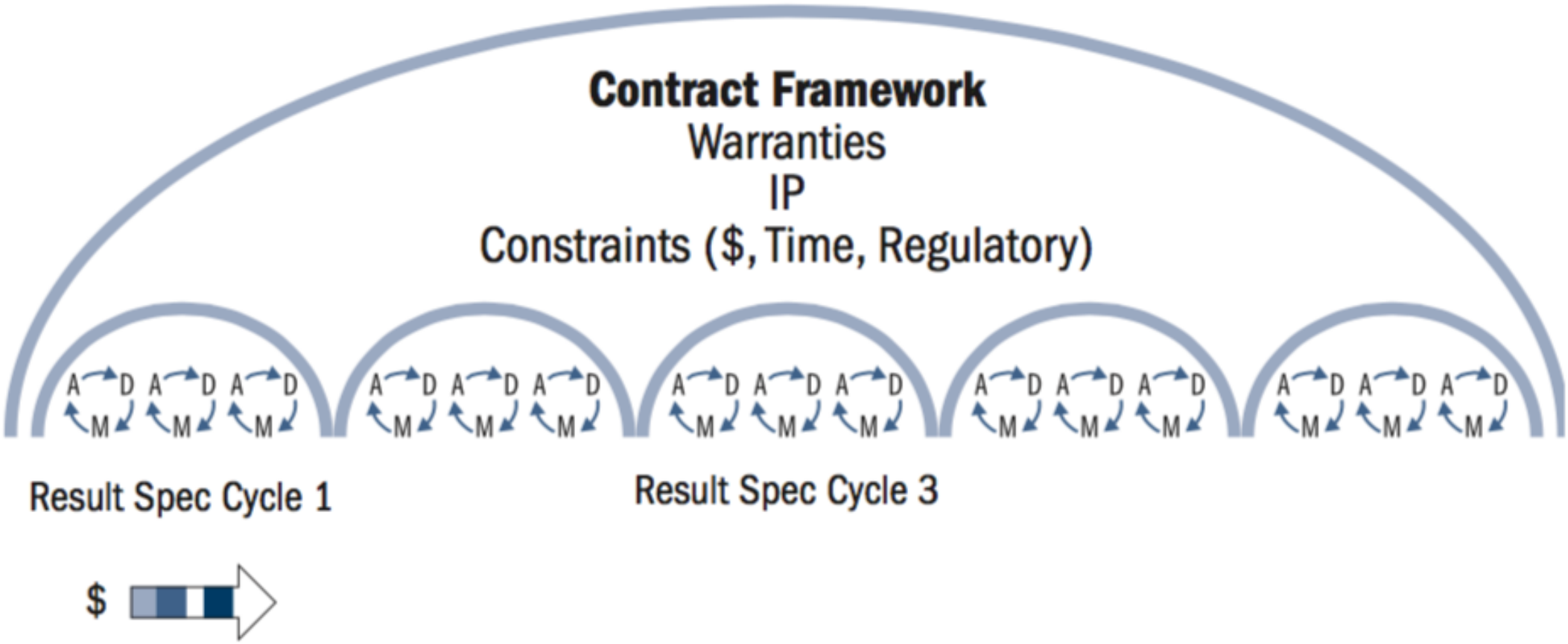
- *"Software Engineering began to emerge in FSD" (IBM Federal Systems Division, f[illegible] out*
- *[illegible] cts – cost overruns, late deliveries, unreliable and incomplete software*
- *Today [Ed. 1980!], management has learned to expect on-time, within budget, deliveries of high-quality software. A Navy helicopter ship system, called LAMPS, provides a recent example. LAMPS software was a four-year project of over 200 person-years of effort, developing over three million, and integrating over seven million words of program and data for eight different processors distrib[illegible]veries [Ed. Note 2[illegible]dget*
- *A mon[illegible],*
- *- Whe[illegible]-years of softwa[illegible]million bytes of pro[illegible]en projects.*
- *- Ther[illegible]ne at all in the pa[illegible]*

in 45 incremental deliveries

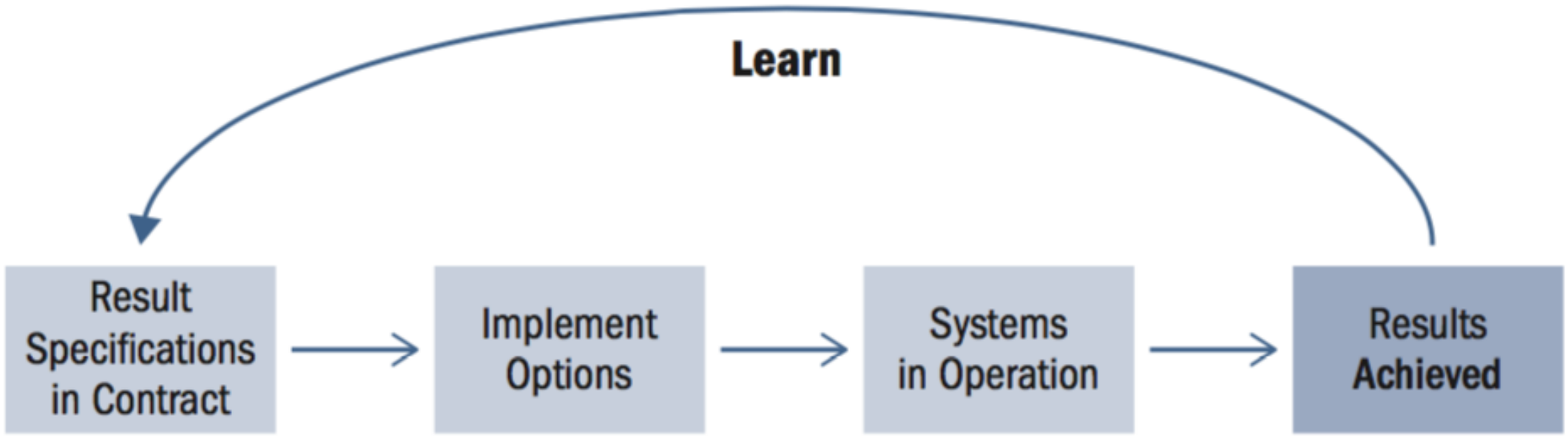
were few late or overrun deliveries in that decade, and none at all in the past four years

6 Agile Contracting: decisions and commitments in smaller increments

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?

WHAT IS A FLEXIBLE CO
A 'flexible contract' is an a **t**
achieves this in several w

The contract focuses on c
features). By focusing on **s to**
align their interests and m
The supplier is given the f **s of**
the contract and stays wit

The fees (or at least part c **to**
achieve the target outcom

The contract is structured **itions,**
under which short-term st **of**
Work, but instead of 'work **o**
acquired knowledge and c

In respect of each SOTO t **n learn**
rapidly what works and w

The contract adopts lightw **O at a**
time, so the financial expo
understand and requires l **n the**
activities of the supplier c

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

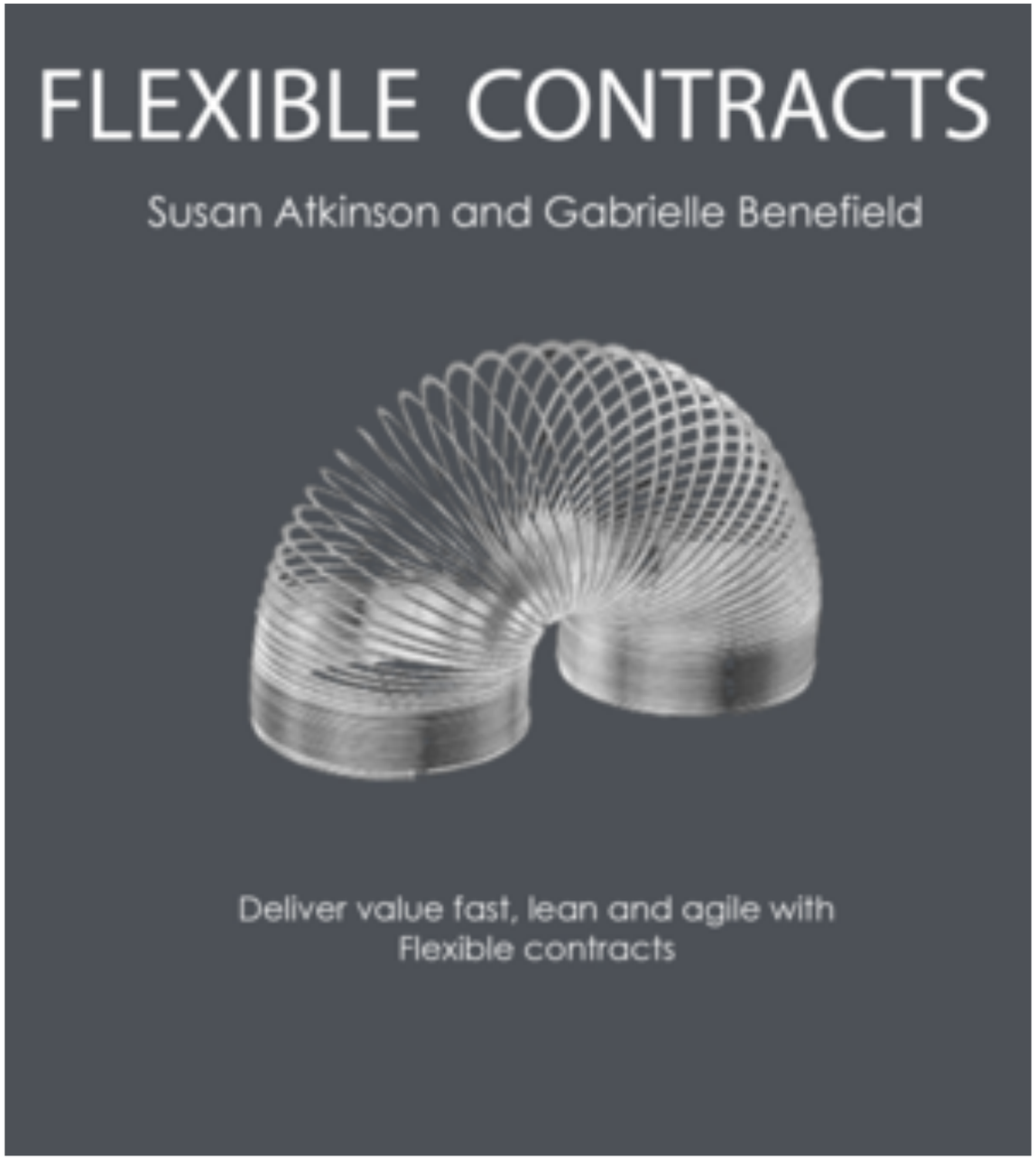
SOTO Completion Date	*NOTE: Please state not applicable if this is not being used.*
The problem or opportunity to be addressed	
The Business Objectives	
The Target Outcomes	*NOTE: These should be in line with the Business Objectives. They should be bullet points only and listed in order of priority.*
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	

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: • Unit of measure:	i.e. the metric used to measure e.g. time, percentage or number
• party responsible for conducting measurement:	i.e. a named person or group responsible for conducting the measurement e.g. the Customer
• Method for measurement:	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
• Frequency of measurement:	i.e. The period of time when measurements will be taken e.g. every ***[2 weeks]*** with their end-users
• Baseline (starting point):	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
- www.mobiusmodel.org

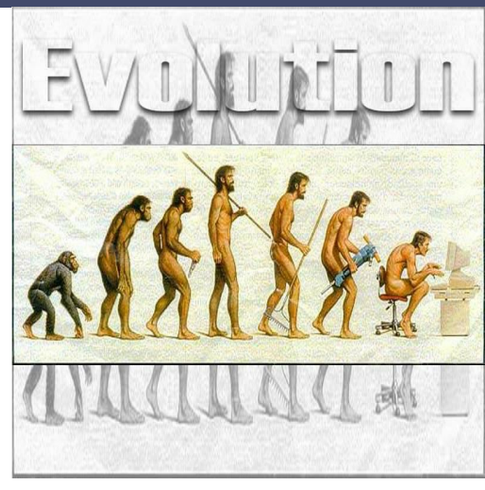
- I have been working together with Susan Atkinson and Gabrielle Benefield for several years regarding these ideas.
- So it is no surprise that they are very complimentary to the Evo and Planguage methods in my writings, such as
- Competitive Engineering (2005), and Value Planning (2014, manus)

References
www.flexiblecontracts.com

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

7 Evo:
a project planning framework
for decision making

’Evo’ defined



A project management process delivering evolutionary results
‘high-value-first’ progress towards the desired goals, and seeking to obtain, and use, realistic, early feedback.

”Complete focus on early rapid delivery of stakeholder value”

Evo characteristics

- *frequent* delivery of system changes (steps)
- steps delivered to stakeholders for *real* use
- feedback obtained from *stakeholders* to determine *next* step(s)
- the *existing* system is used as the initial system base
- *small* steps (ideally between 2%-5% of total project financial cost and time)
- steps with *highest value* and benefit-to-cost ratios given highest *priority* for delivery
- feedback used 'immediately' to modify long term plans and requirements and, also
- to decide on the *next* step total systems approach ('change *anything* that helps') -
- *results*-orientation ('delivering the results' is prime concern)

How does EVO differ from waterfall/prototyping?

In a nutshell

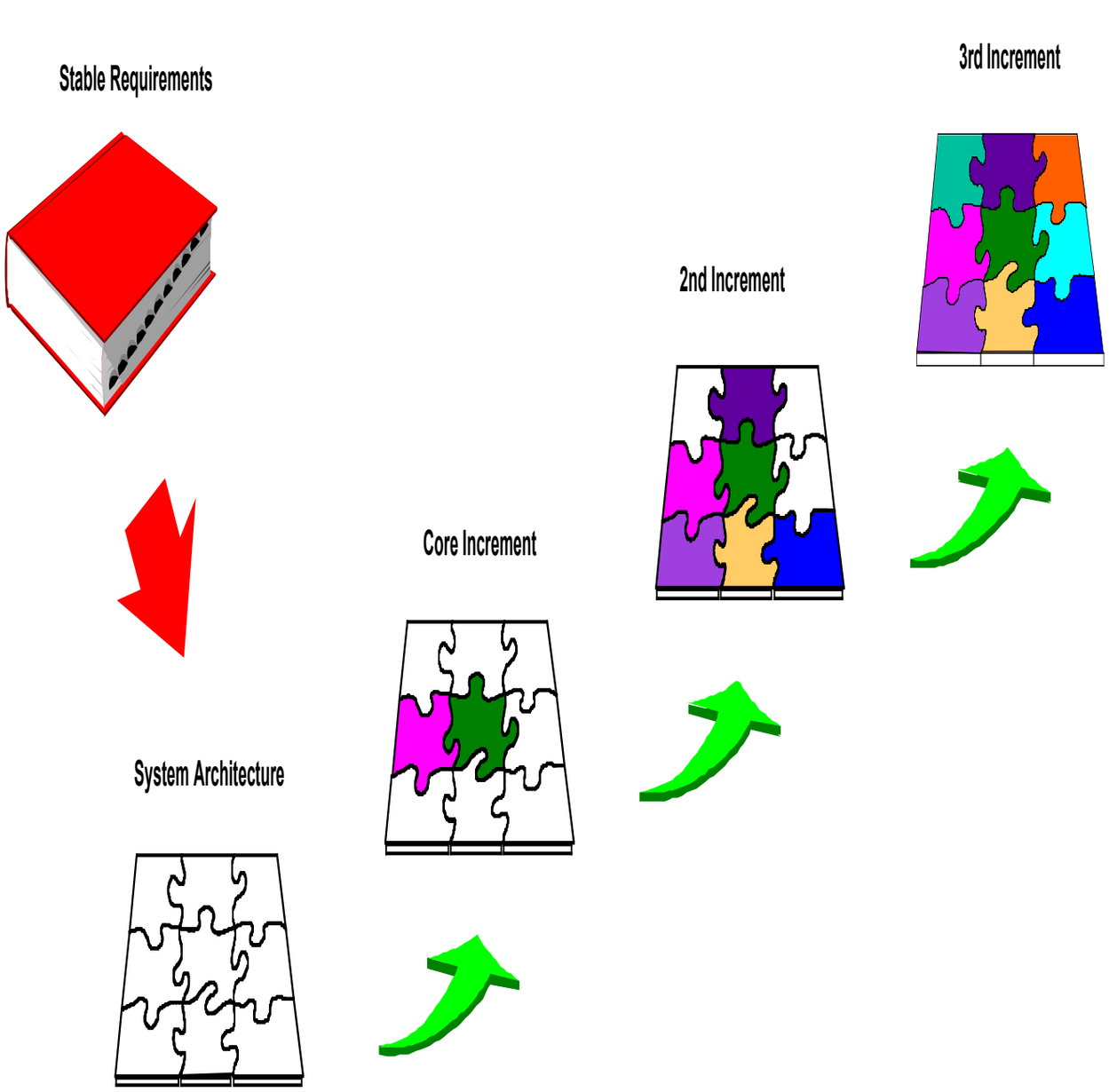
Early visible results in the business.

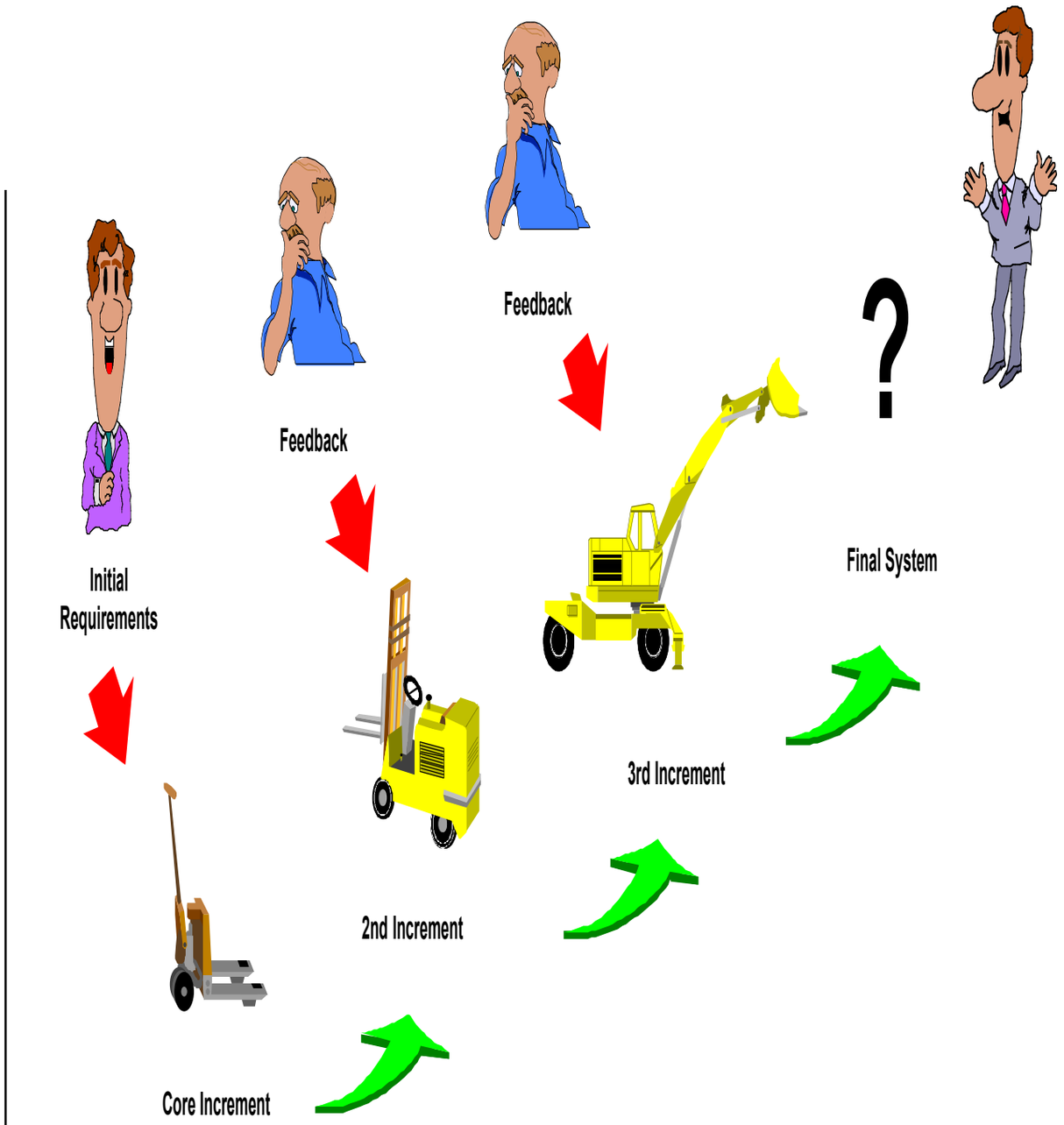
In more detail:

1. ***Weekly*** **result delivery focus: *real* action**
2. **Results at *beginning* of project**
3. ***Total systems*** **thinking - not 'IT'**
4. **More intimate concern for business needs**
5. **Proof of ability to deliver value**
6. **Staff priority deployment flexibility**
7. **Value/cost ratio much more visible**

How does Evo differ from Incremental?

(see next slide for text summary)





Source: A Strategy for Acquiring Large and Complex Systems. Dr. Helmut Hummel, Bonn September 23 2002, see note for paper, Email: hummel@iabg.de

How does Evo differ from Incremental?

Evo

- Focus on *business value*
- Ability to *learn* rapidly
- *Quantified* value tracking
- *Cooperation* with users *continuous*

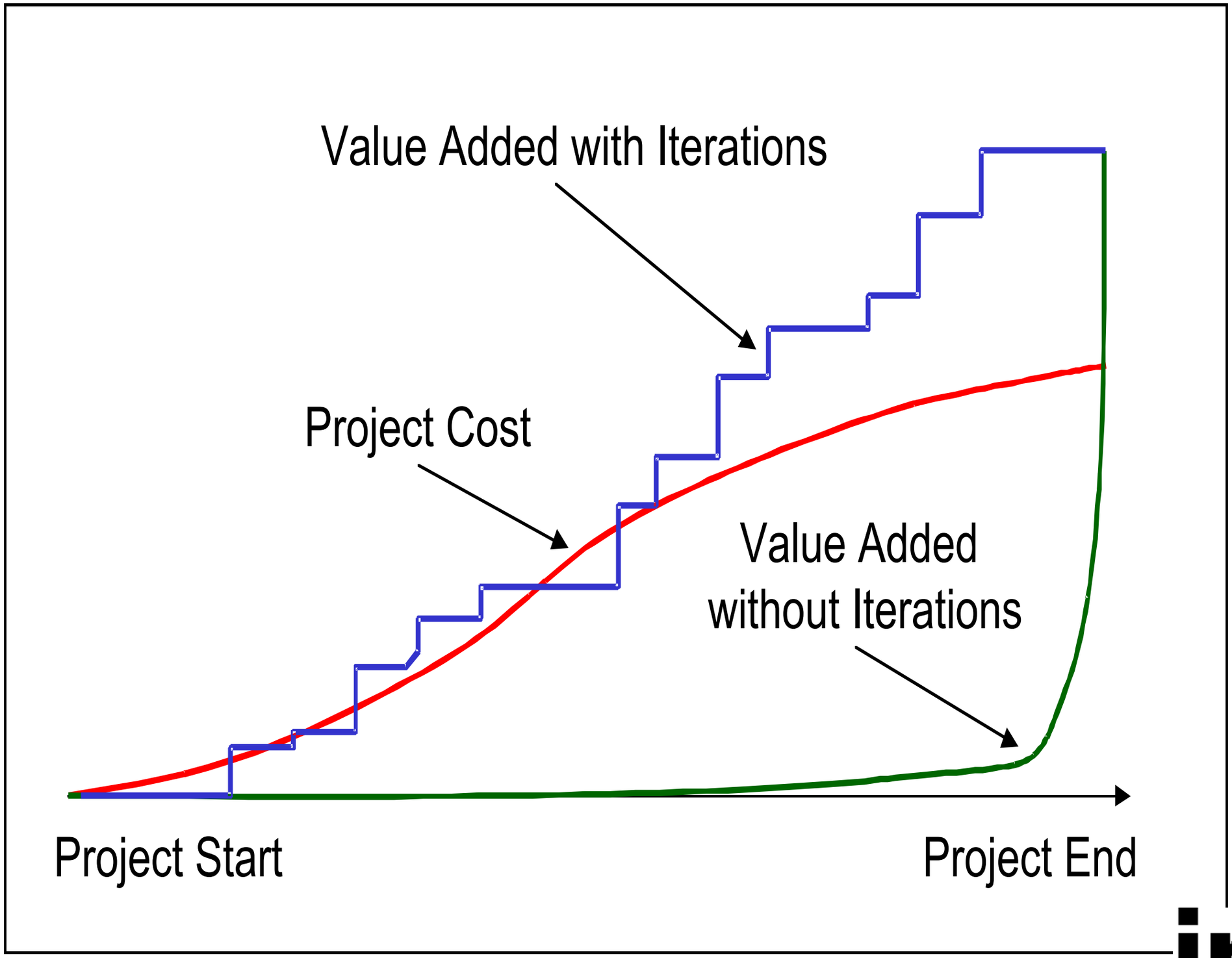
Incremental

- Focus on *construction*
- No intent to learn or change plans
- No value tracking
- No plan to cooperate with users

What are the major benefits of Evo?

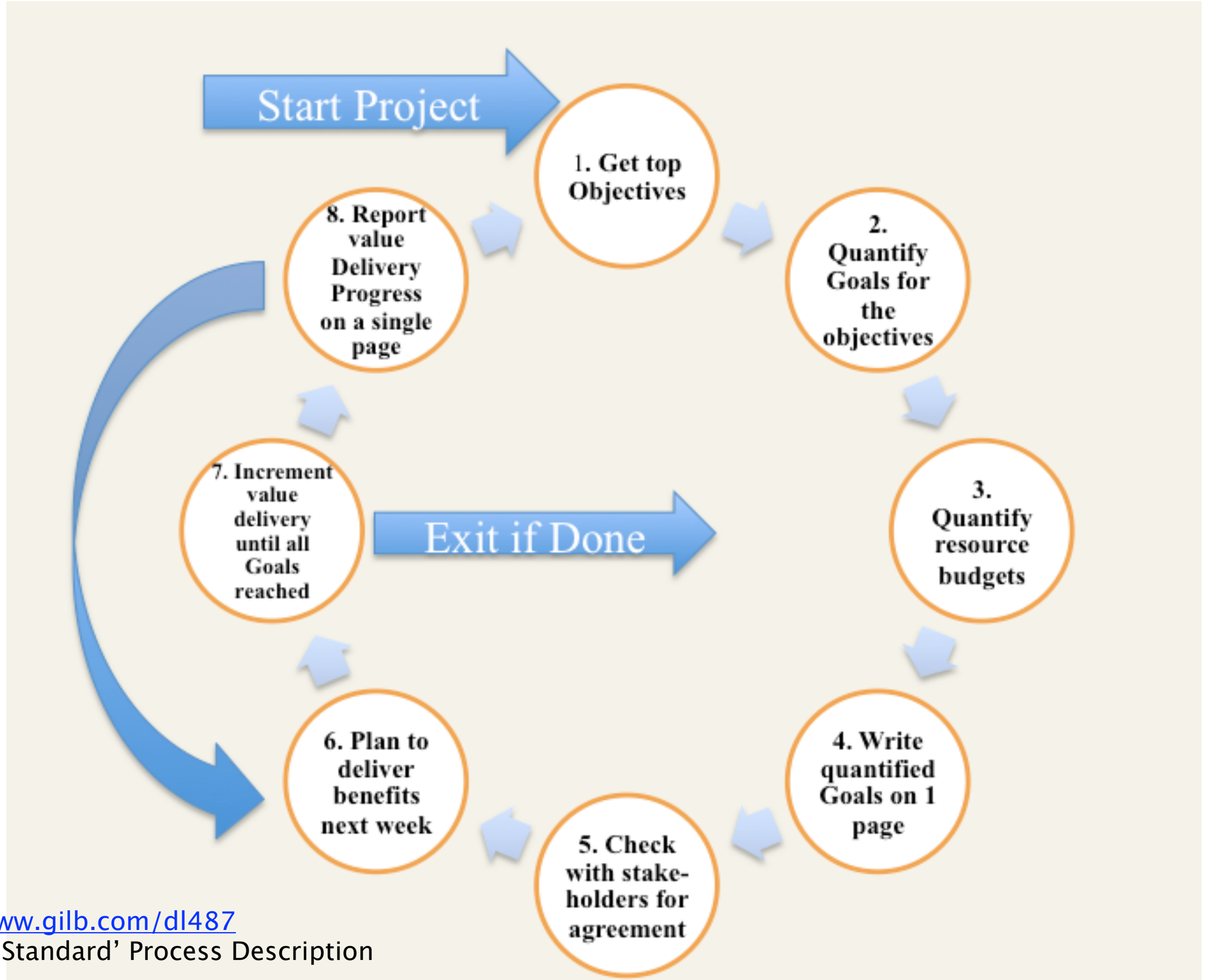
- Management control of value
- Management control of costs
- Enforcing business thinking
  - Instead of IT thinking
- Flexibility for management to re-prioritize projects and spend
- Improves system maintenance culture
  - Because you ‘maintain’ at each step
  - Very low risk to do it and see if it works

Value Added Paradigm



Courtesy: Erik Simmons, Intel Oregon

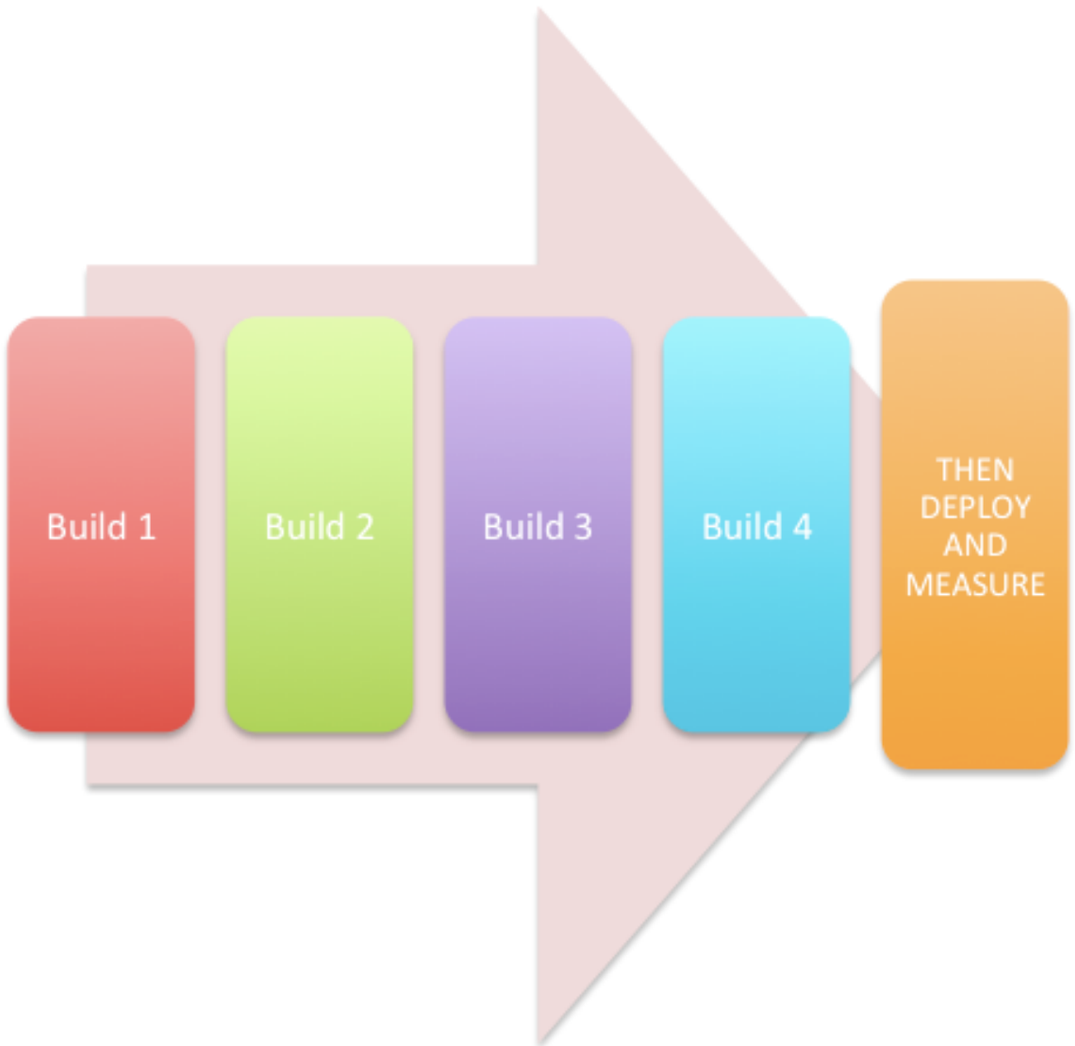
A View of the ‘Evo’ Agile for values Project Management Process



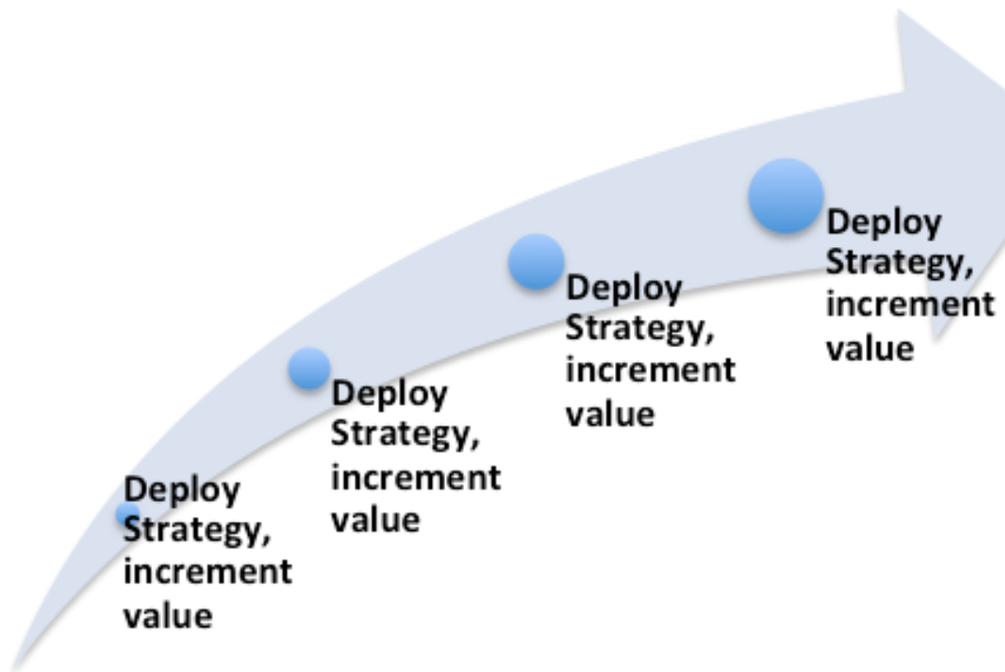
http://www.gilb.com/dl487
The Evo ‘Standard’ Process Description

Value Decomposition

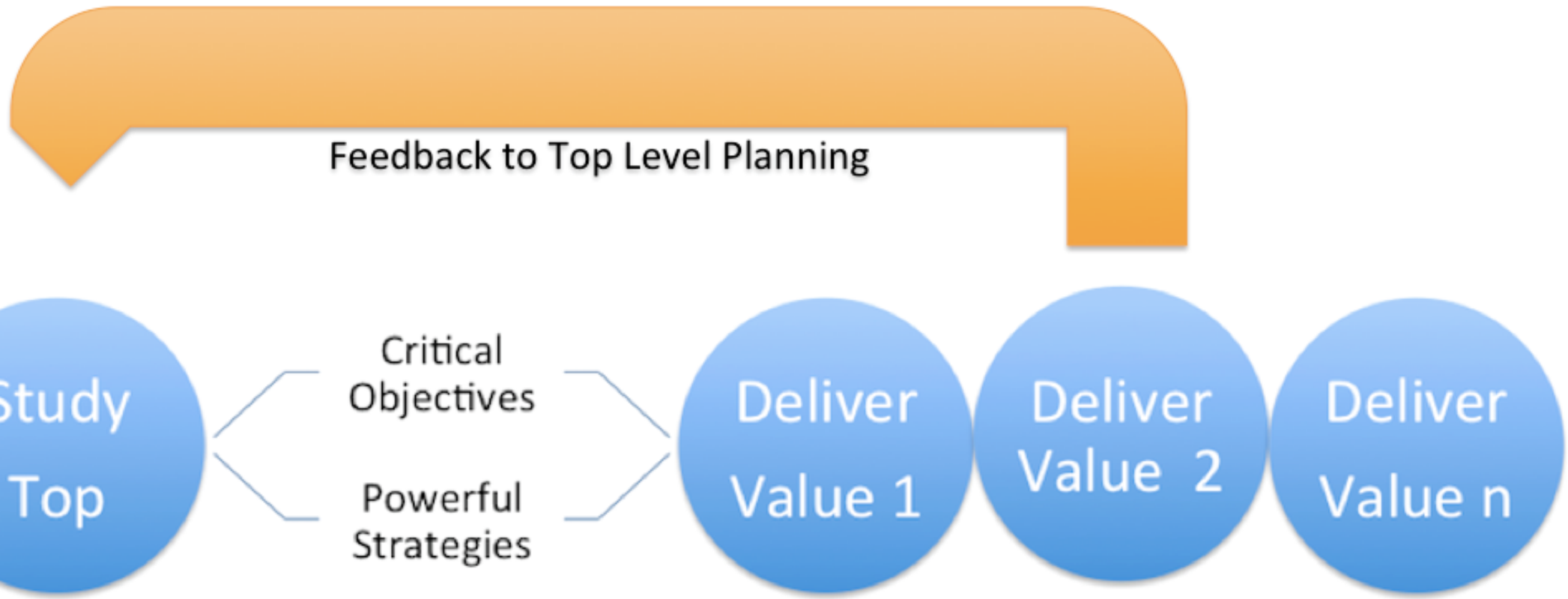
Not decomposition for this



More like this



Value Delivery Cycle Decomposition



What are the major technology process changes?

You need clear, quantified *requirements* to 'evolve' towards - 'stakeholders view' requirements

Test process: changes - rapid, early

User involvement continuous

Teamwork towards one user result

Open Ended Architecture to Evo in

Backroom and Frontroom management

How do you best manage it?

Motivate development team by results

Empower stakeholders to think value

Train development in Evo

Equip with Evo ‘tools’ (templates etc)

Support and advise (new) teams

Feed budget to teams with best value

What are the pitfalls?

Failing to focus on real value

Failing to use value/cost priority

Failure to train and support after training

Giving up too early and falling back on old habits

Lack of management commitment

Lack of management support

Defeatism: giving up rather than cracking problems.

What are the pre-requisites?
(eg componentised architecture)

Clear management policy

Evo tools (standards)

Trained Project Management

Reward structure

Long term quantified objectives

Evo plan for Evo method

Enthusiastic volunteer projects

Open architecture is useful but not a start condition!

Are there types of apps/users that EVO might not be appropriate for?

In principle no, but

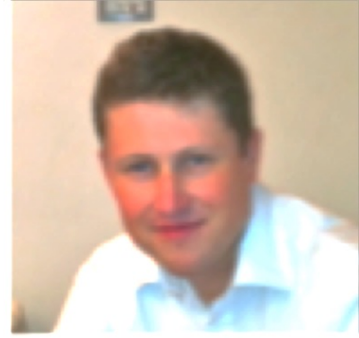
- Some projects will have greater benefits
- Even ‘old’ failing projects can be ‘saved’ by Evo restructuring
- Bigger projects will have more benefit
- There may be some projects with ‘constraints’ (like dates for laws or consortium agreements) so you can’t really deliver much before a distant time.

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

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

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

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



Richard Smith's Planning Tool which we are using on BCS Courses Great for 'First Week' and all later weeks followup

needsandmeans.mod.bz

Tom Gilb & K...ents-Material | appleinsider.com | Google Docs | TOM'S NET Services | Resources » | NORSKE STEDER | T r a v e l 4 TOM | Social Sites | NEWS | ALLE AND

Startup Planning Course Doc - Dropbox | Needs & Means - Demo

needs&means | Specifications | Impact Tables | Documents | Glossary | Follow Me | Help

Home / Impact Tables / IET-6PGBWPE

BCS.Managing-Software-Technology

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

https://app.needsandmeans.com

End Game

The Fundamental Principles of Value-Driven IT Systems 'Engineering'.

1. Values are multiple and simultaneous: unavoidable.

2. All technical solutions contain multiple values and costs.

3. All values and costs have unknowns, uncertainties and risks.

4. Value delivery must work incrementally, with feedback and change.

Free Book Manuscript

- Tinyurl.com/ValuePlanning (a live dropbox)
- Manuscript 104 subchapters
- Drafted Summer/Fall 2014
- Major 50% Edit Summer 2015, Ongoing in Fall
- Feedback appreciated
- Aimed at ‘management’
  - (not IT or Engineers)