Practical Management Tools for the Top 5 Issues in IT Management

tom@Gilb.com
gilb.com
Honorary Fellow of BCS

13th Nov 2015, 08:45 - 13th Nov 2015, 17:00

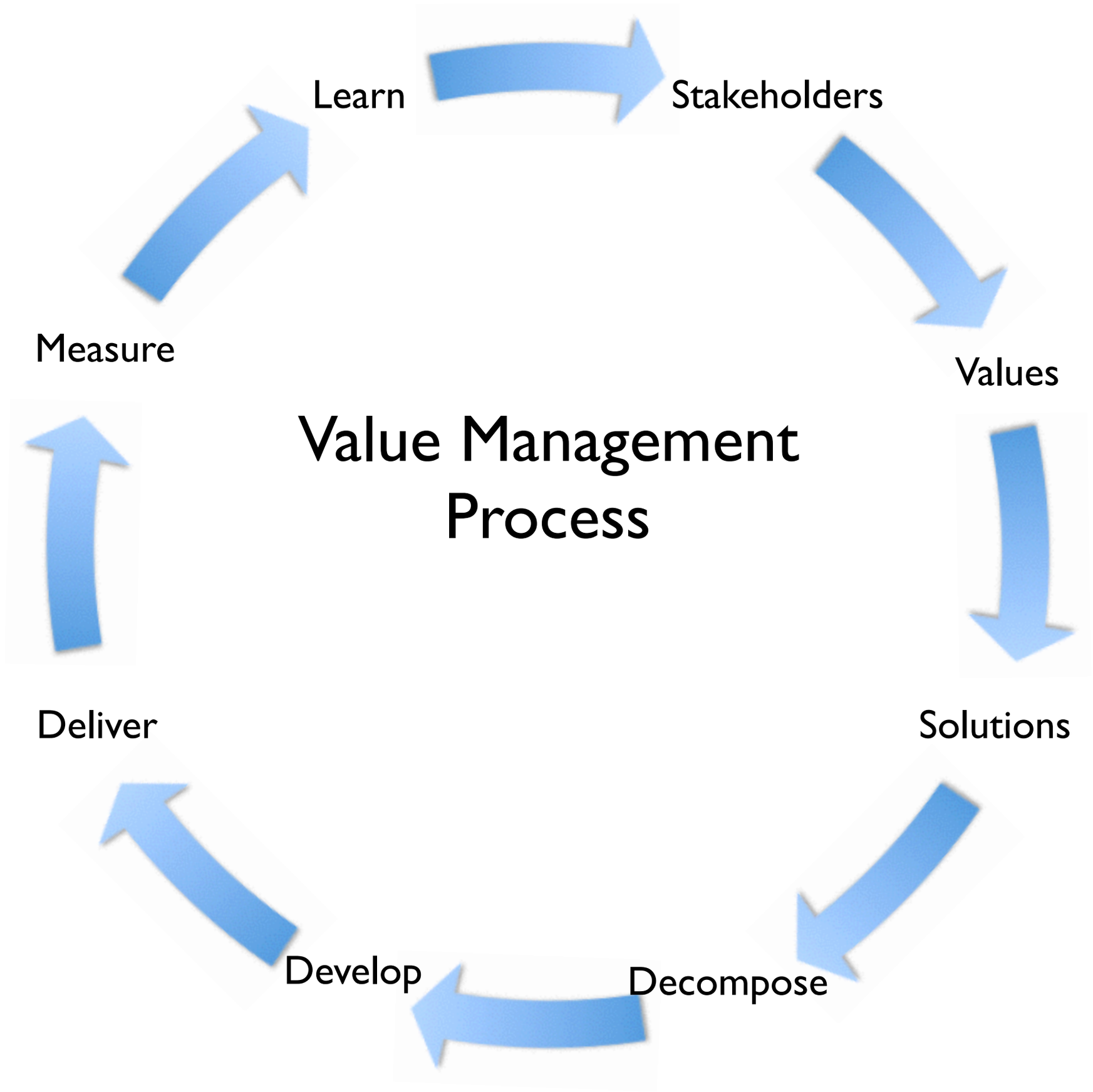
1. Aligning IT with business.

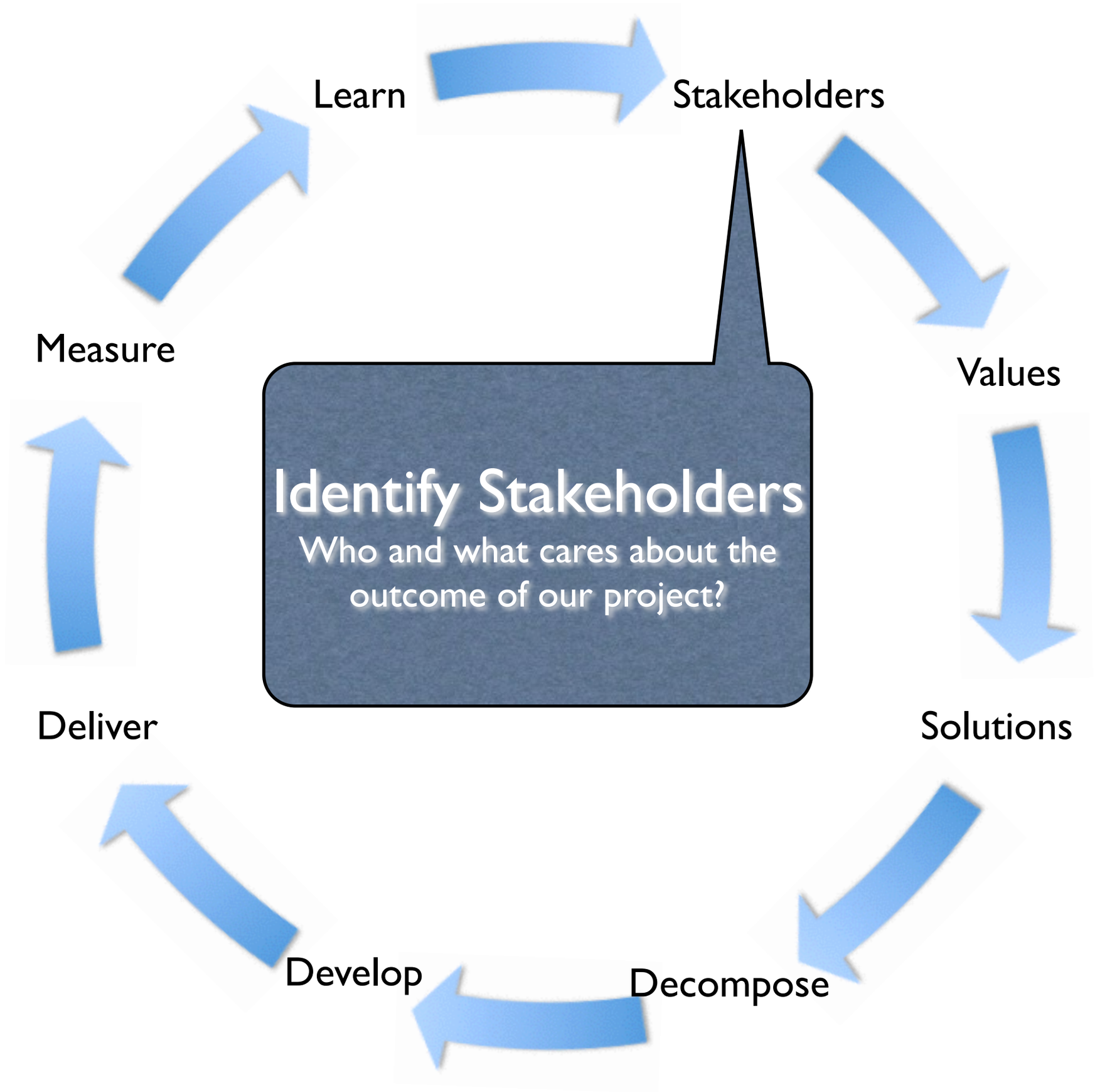
The use of Value Decision Tables
for numerically aligning IT with the business

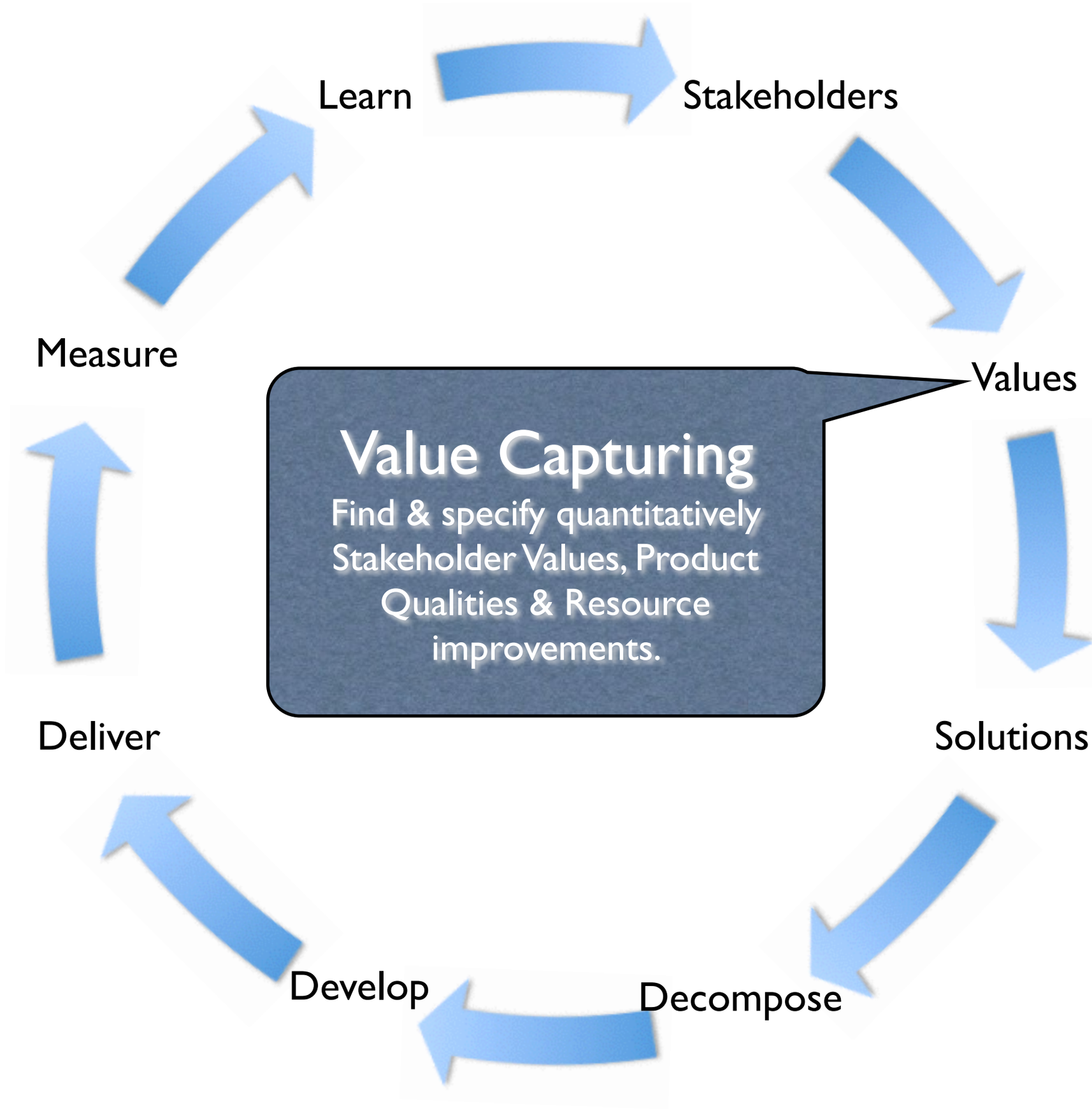
The alignment with multiple stakeholders in and related to the business

Main Point
You can connect any related levels of business and technology.
Numerically,
with multiple critical objectives and
their supporting strategies or ‘means objectives’

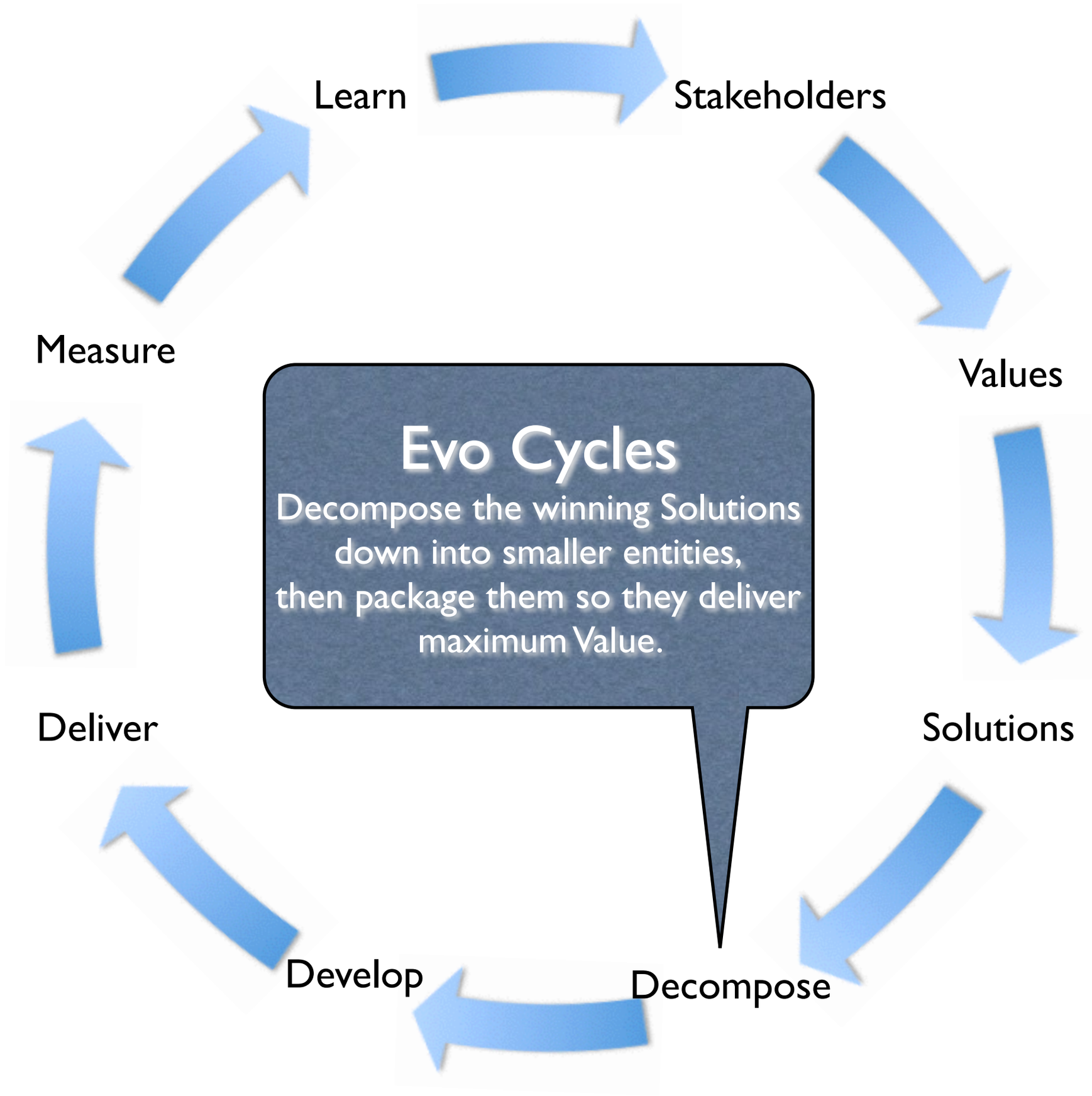
Value Management Process

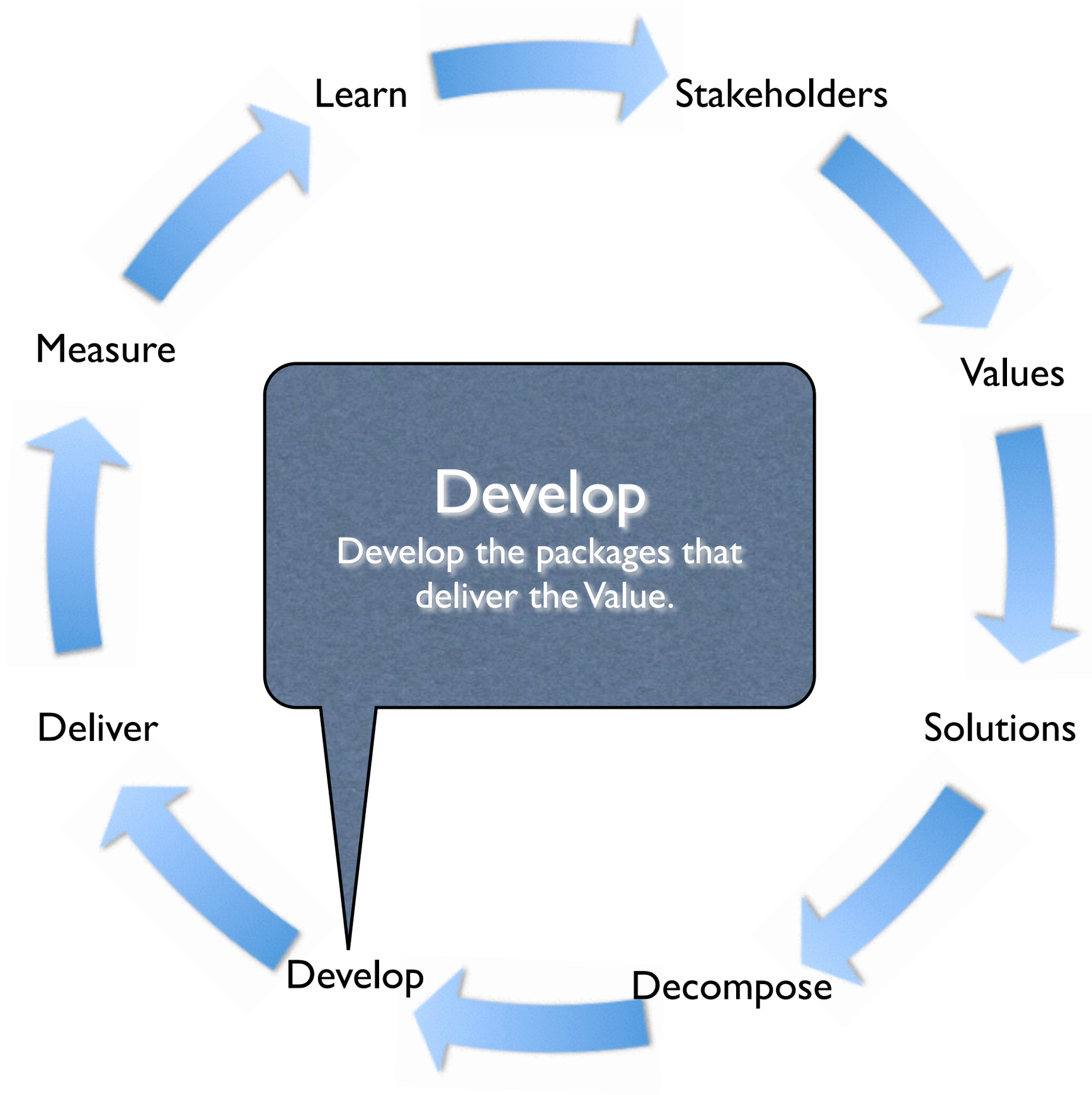



Learn
Stakeholders
Values
Solutions
Decompose
Develop
Deliver
Measure
Identify Stakeholders
Who and what cares about the outcome of our project?

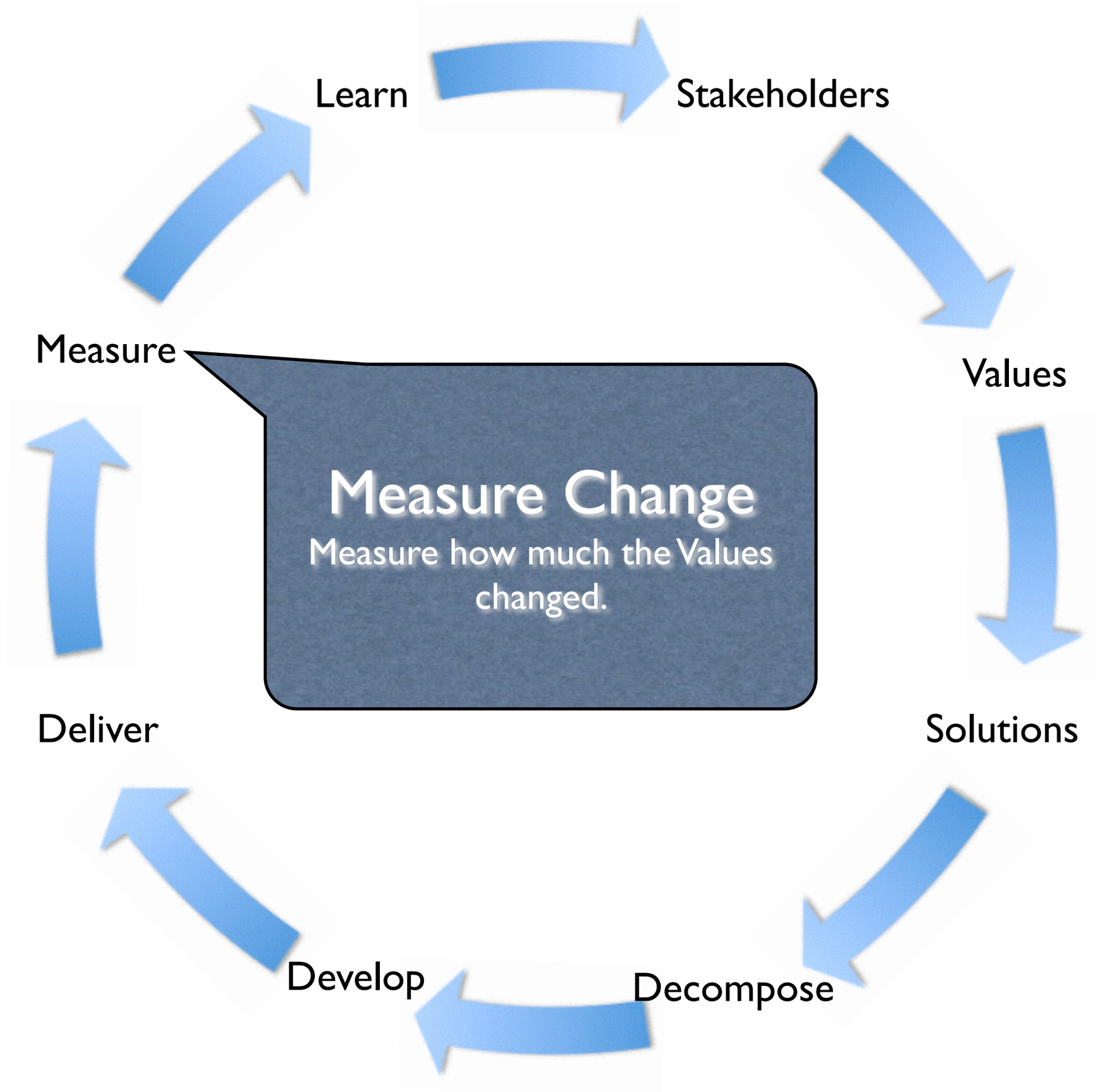

Learn
Stakeholders
Values
Solutions
Decompose
Develop
Deliver
Measure
Value Capturing
Find & specify quantitatively Stakeholder Values, Product Qualities & Resource improvements.

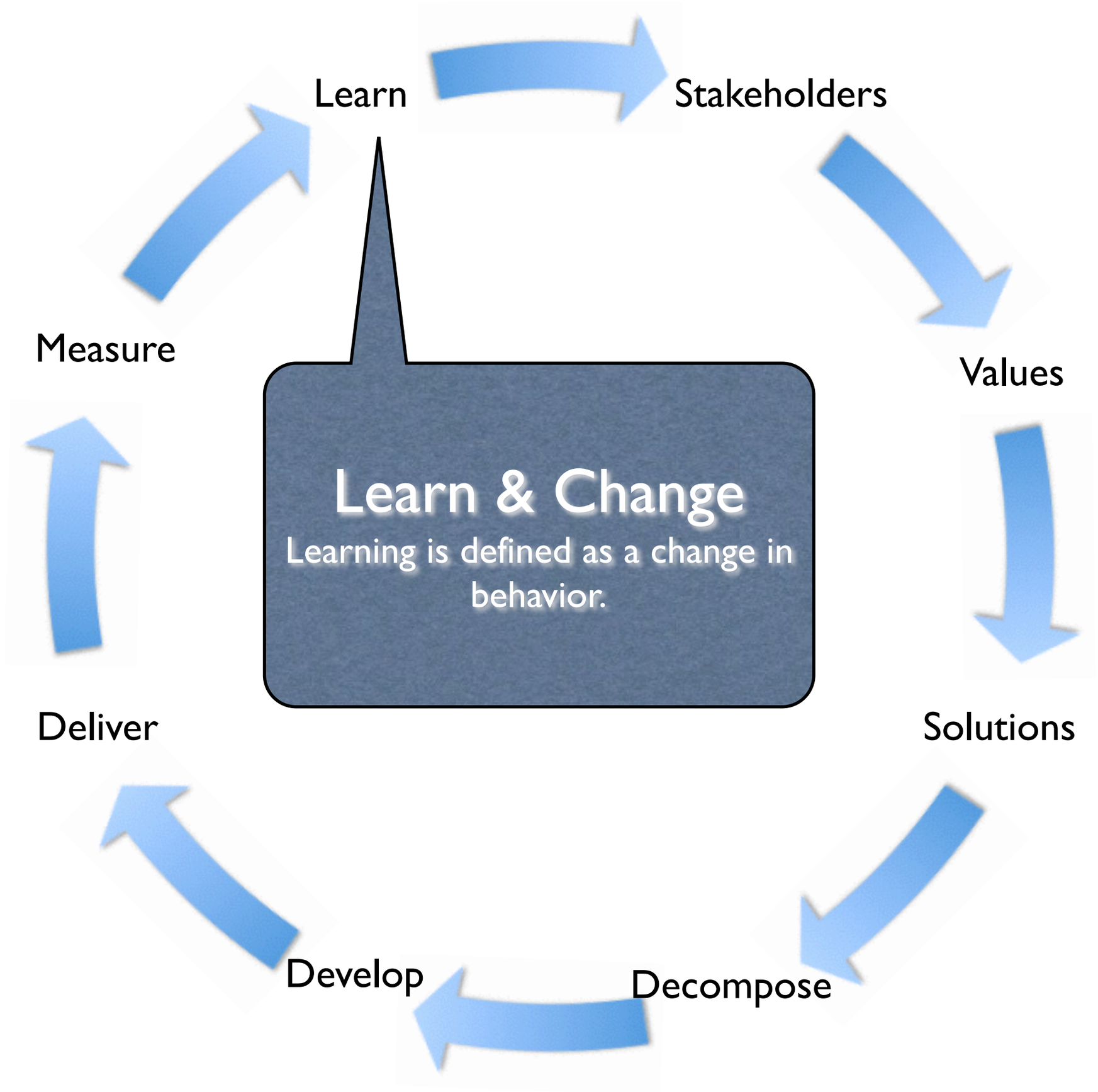

Learn
Stakeholders
Values
Solutions
Decompose
Develop
Deliver
Measure
Solution Prioritization
Find, Evaluate & Prioritize Solutions to satisfy Requirements.


Learn
Stakeholders
Values
Solutions
Decompose
Develop
Deliver
Measure
Evo Cycles
Decompose the winning Solutions down into smaller entities, then package them so they deliver maximum Value.

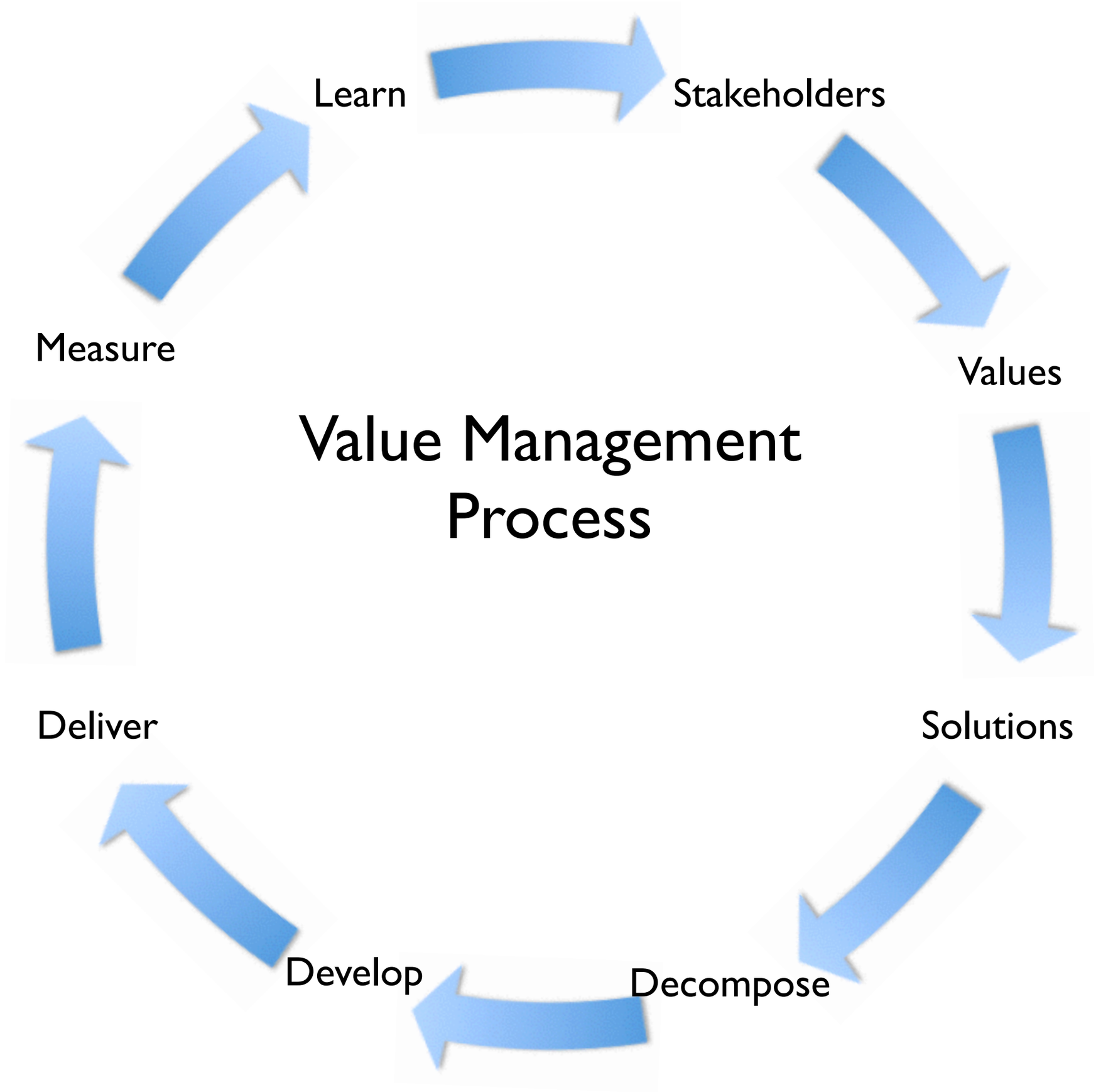

Learn
Stakeholders
Values
Solutions
Decompose
Develop
Deliver
Measure
Develop
Develop the packages that deliver the Value.


Learn
Stakeholders
Values
Solutions
Decompose
Develop
Deliver
Measure
Deliver
Deliver to Stakeholders
improved Value.
(not always a thing or code)


Learn
Stakeholders
Values
Solutions
Decompose
Develop
Deliver
Measure
Measure Change
Measure how much the Values changed.


Learn
Stakeholders
Values
Solutions
Decompose
Develop
Deliver
Measure
Learn & Change
Learning is defined as a change in behavior.

Value Management Process



Value Management Process

Learn

Stakeholders

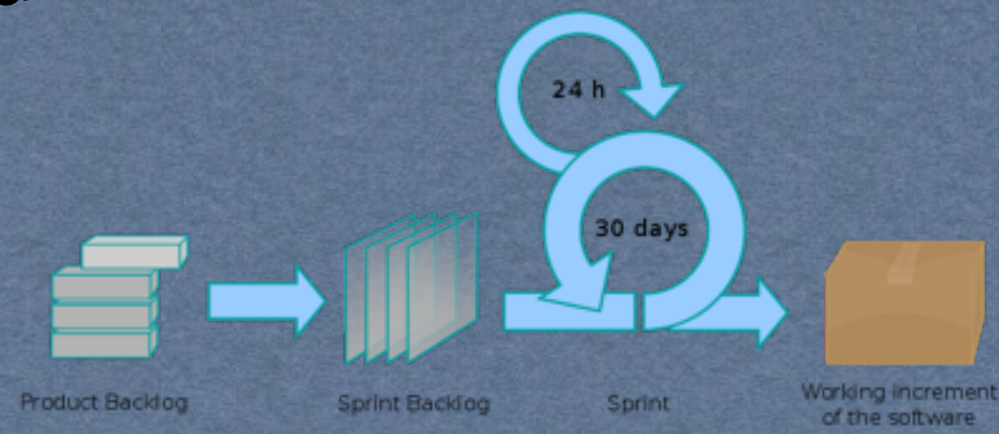
Values

Solutions

Deliver

Measure

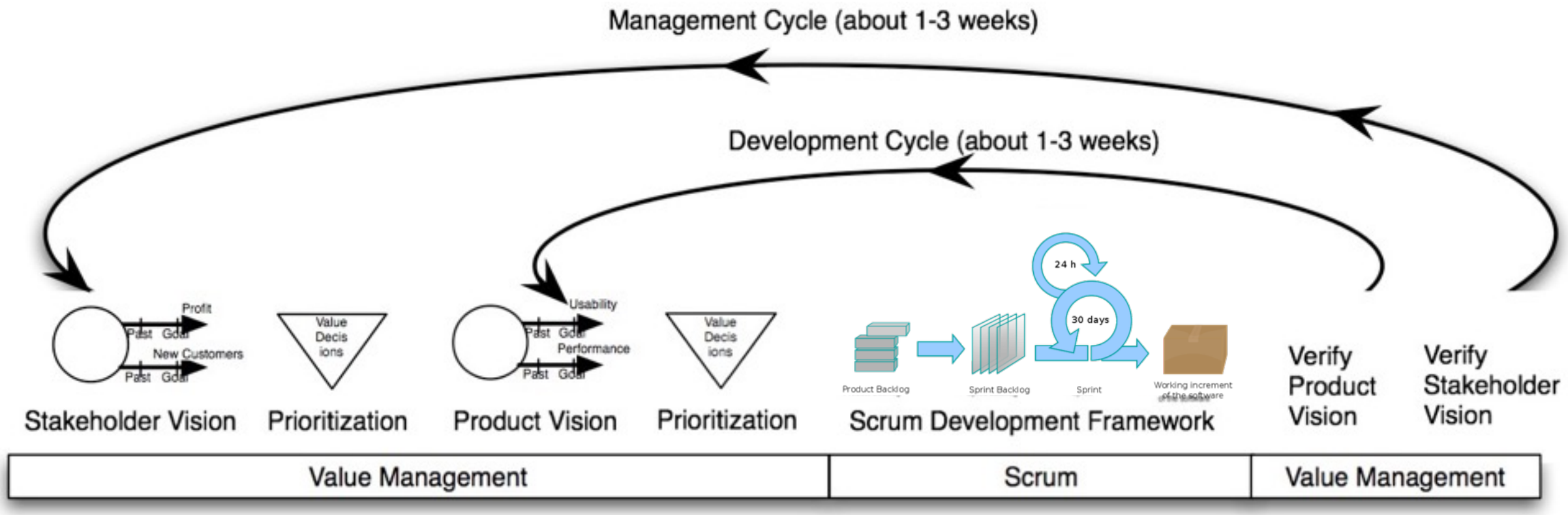
Scrum



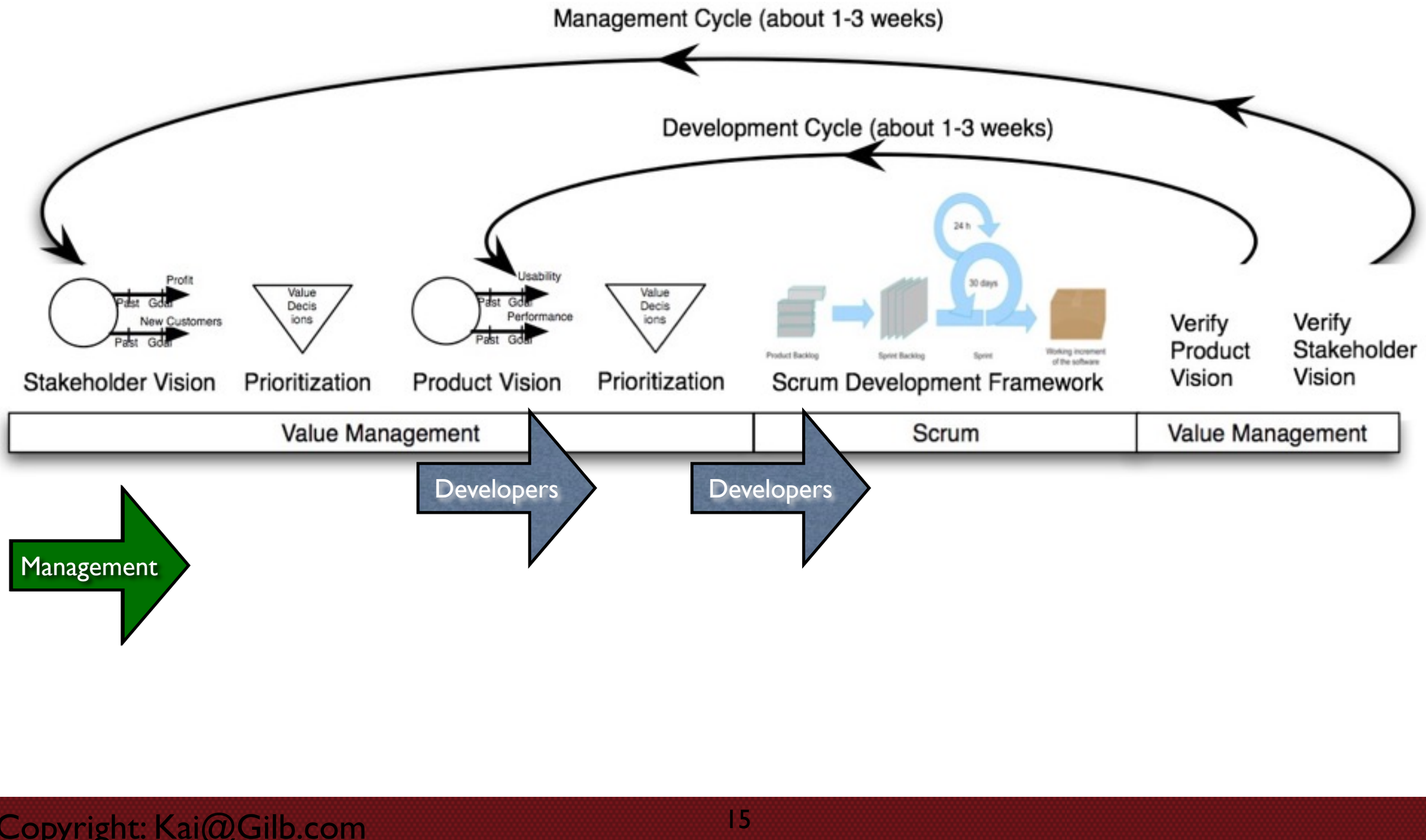
Develop

Decompose

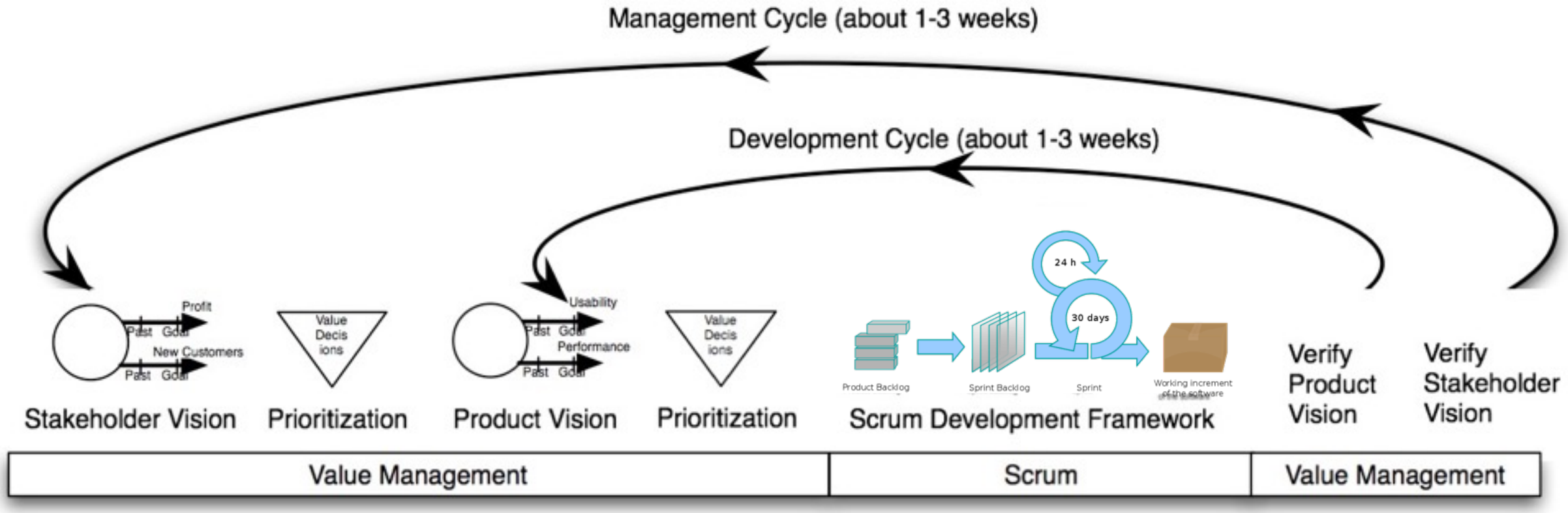
Value Management



Value Management



Value Management



Value Decision Tables

Based on a real ‘project saving’ case by Kai Gilb at ‘Bring’ (Package Transportation)

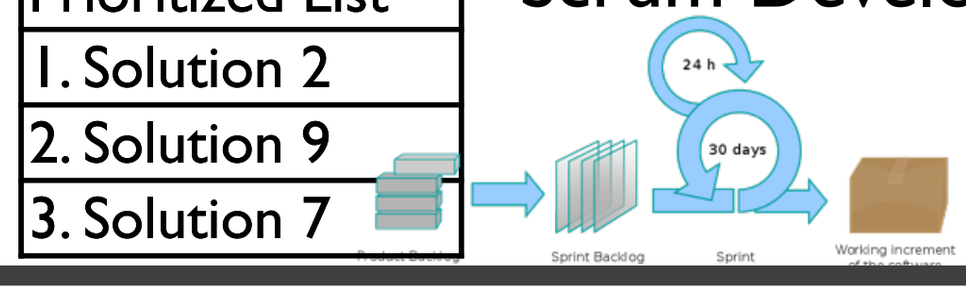
Business Goals	Stakeholder Value 1	Stakeholder Value 2
Business Value 1	-10%	40%
Business Value 2	50%	10%
Resources	20%	10%

Stakeholder Val.	Product Value 1	Product Value 2
Stakeholder Value 1	-10%	50 %
Stakeholder Value 2	10 %	10%
Resources	2 %	5 %

Product Values	Solution 1	Solution 2
Product Value 1	-10%	40%
Product Value 2	50%	80 %
Resources	1 %	2 %

Prioritized List
1. Solution 2
2. Solution 9
3. Solution 7

Scrum Develops



We measure improvements
Learn and Repeat

Value Decision Tables

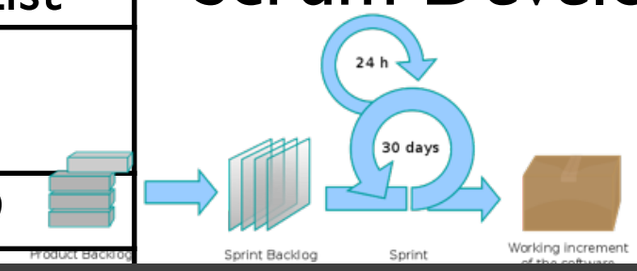
Business Goals	Training Costs	User Productivity
Profit	-10%	40%
Market Share	50%	10%
Resources	20%	10%

Stakeholder Val.	Intuitiveness	Performance
Training Costs	-10%	50 %
User Productivity	10 %	10%
Resources	2 %	5 %

Product Values	GUI Style Rex	Code Optimize
Intuitiveness	-10%	40%
Performance	50%	80 %
Resources	1 %	2 %

Prioritized List
1. Code Optimize
2. Solution 9
3. Solution 7

Scrum Develops



We measure improvements
Learn and Repeat

Value Decision Tables

Business Goals	Training Costs	User Productivity
Profit	-10 %	40 %
Market Share	50 %	10 %
Resources	20 %	10 %

Stakeholder Val.	Intuitiveness	Performance
	-10 %	50 %
	10 %	10 %
Resources	2 %	5 %

Product Values	GUI Style Rex	Code Optimize
Intuitiveness	-10 %	40 %
Performance	50 %	80 %
Resources	1 %	2 %

Prioritized List
1.
2. Solution 9
3. Solution 7

Value Decision Tables

Business Goals	Training Costs	**User Productivity**
Profit	-10%	40%
Market Share	50%	10%
Resources	20%	10%

U P gives me 40% progress towards my 'Profit' Goal

Stakeholder Val.	Intuitiveness	Performance
	-10 %	50 %
	10 %	10 %
Resources	2 %	5 %

Product Values	GUI Style Rex	Code Optimize
Intuitiveness	-10 %	40 %
Performance	50 %	80 %
Resources	1 %	2 %

Prioritized List
1.
2. Solution 9
3. Solution 7

Value Decision Tables

Business Goals	Training Costs	User Productivity
Profit	-10%	40%
Market Share	50%	10%
Resources	20%	10%

Stakeholder Val.	Intuitiveness	Performance
Training Costs	-10 %	50 %
User Productivity	10 %	10 %
Resources	2 %	5 %

Product Values	GUI Style Rex	Code Optimize
Intuitiveness	-10 %	40 %
Performance	50 %	80 %
Resources	1 %	2 %

Prioritized List
1.
2. Solution 9
3. Solution 7

Value Decision Tables

Business Goals	Training Costs	User Productivity
Profit	-10%	40%
Market Share	50%	10%
Resources	20%	10%

Stakeholder Val.	Intuitiveness	Performance
Training Costs	-10 %	50 %
User Productivity	10 %	10 %
Resources	2 %	5 %

Product Values	GUI Style Rex	Code Optimize
Intuitiveness	-10 %	40 %
Performance	50 %	80 %
Resources	1 %	2 %

Prioritized List
1.
2. Solution 9
3. Solution 7

Value Decision Tables

Business Goals	Training Costs	User Productivity
Profit	-10%	40%
Market Share	50%	10%
Resources	20%	10%

Stakeholder Val.	Intuitiveness	Performance
Training Costs	-10 %	50 %
User Productivity	10 %	10 %
Resources	2 %	5 %

Product Values	GUI Style Rex	**Code Optimize**
Intuitiveness	-10 %	40 %
Performance	50 %	80 %
Resources	1 %	2 %

Prioritized List
1.
2. Solution 9
3. Solution 7

Value Decision Tables

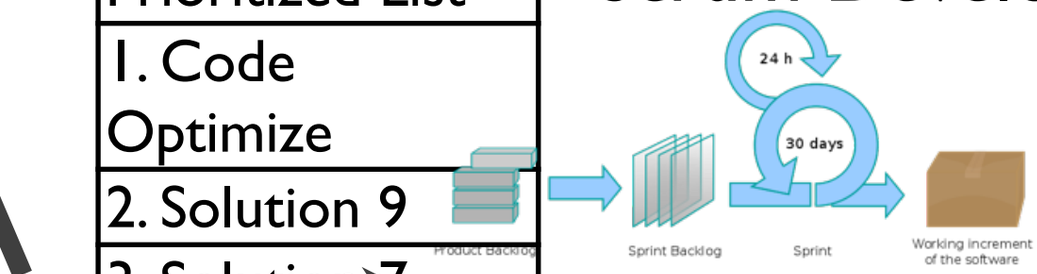
Business Goals	Training Costs	User Productivity
Profit	-10%	40%
Market Share	50%	10%
Resources	20%	10%

Stakeholder Val.	Intuitiveness	Performance
Training Costs	-10%	50 %
User Productivity	10 %	10%
Resources	2 %	5 %

Product Values	GUI Style Rex	Code Optimize
Intuitiveness	-10%	40%
Performance	50%	80 %
Resources	1 %	2 %

Prioritized List
1. Code Optimize
2. Solution 9
3. Solution 7

Scrum Develops



We measure improvements
Learn and Repeat

Business Owners

Steering Committee

Push Technical Solutions

Wants to make decisions about Technical Solutions

Project Management

Thinks and understands Technical Solutions

Developers

Business Owners

What are your real needs?

Steering Committee

Sign off on Value Improvements

Project Management

What technical solution will give maximum Product Value improvements?

Developers

Down's Syndrome Case Objectives, Functions: Brodie PhD Case 2014

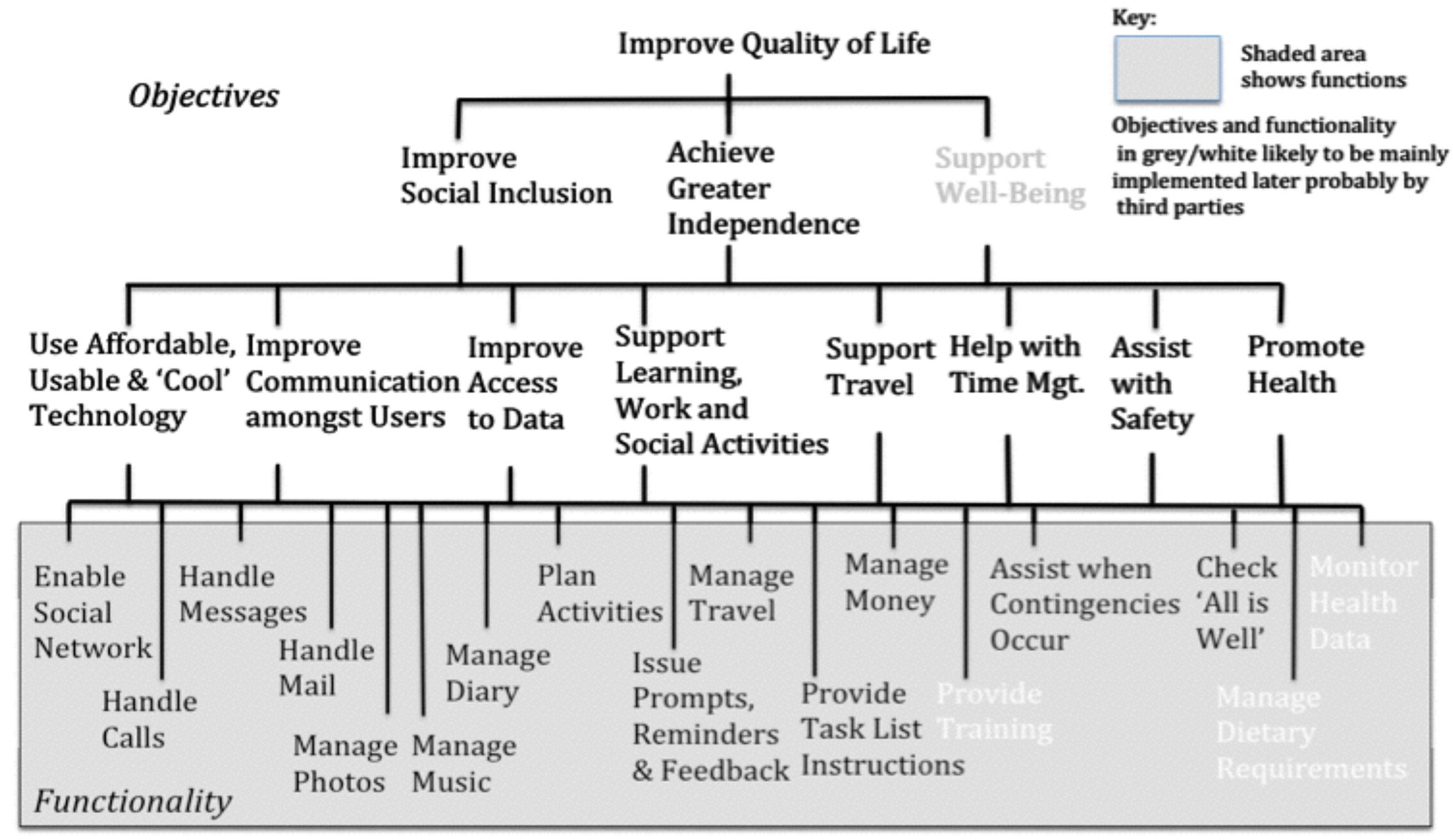


Figure 5.X: Primary user objectives and functionality

2. Benefit/value ROI from IT / data-information -

maximising / demonstrating /realisation (related to IT governance)

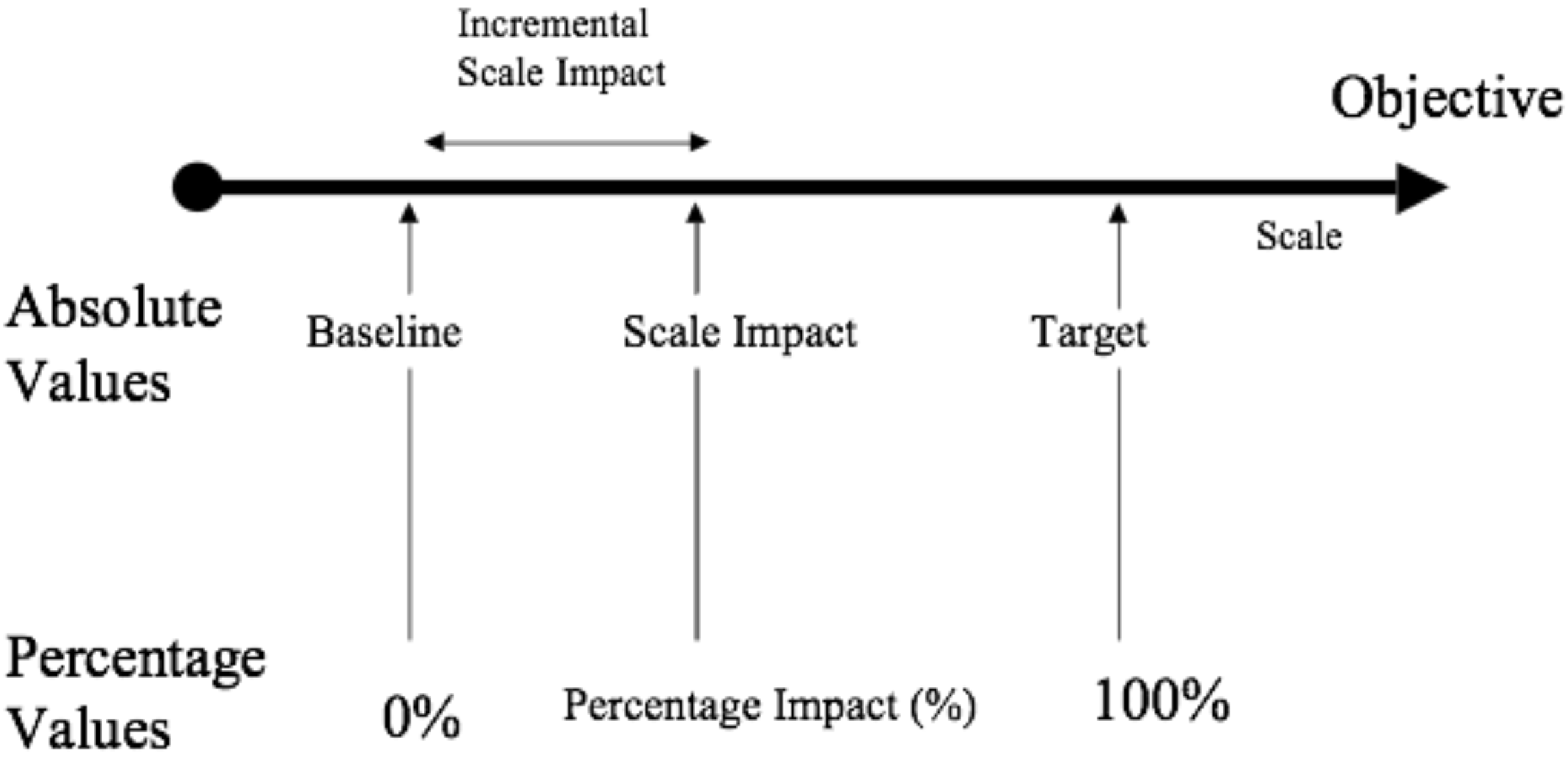
Using the Value Decision Tables to bring out the multiple values of strategies with respect to their multiple costs

Using Value Decision Tables to track value delivery numerically in project management

Main Points.

Absolutely all business values
can be expressed numerically
and can be measured continuously and incrementally
and
can be related to any interesting cost aspects
(CapEx, OpEx, Time, People)
to determine
Values for resources

Impact Estimation Basic Concepts



Source: Lindsey Brodie PhD (2015), Editor of Competitive Engineering May 2000

Wine Year Impact Table

	02	01	00	99	98	97	96	95	94	93
Champagne	♎	♎	♎	3	2	2	3	★	2	3
Red Bordeaux	♎	3	★	3	3	3	3	☆	1	2
Red Burgundy	☆	3	3	3	3	3	★	☆	✝	3
White Burgundy	☆	3	2	2	2	3	★	☆	✝	2
Beaujolais	2	1	3	3	3	✝	✝	✝	✝	✝
Cotes du Rhone	2	3	☆	☆	★	3	2	3	2	1
Alsace	♎	☆	3	3	3	3	★	3	3	2
Rioja	♎	♎	2	2	2	★	1	2	3	1
Australia	3	3	☆	2	★	2	3	☆	3	2
New Zealand	3	☆	☆	3	★	2	2	2	3	3
California	♎	♎	3	☆	3	★	3	3	☆	3

★ Don's All-Time Greats

☆ Exceptional

3 Very Good

2 Good

1 Average

✝ Dead & Buried - no longer worth worrying about their rating

♎ Still to be declared

Niels Malotaux

Impact Estimation principle

How much % of what we want to achieve do we achieve by this solution

At what cost ?

Possible solutions to achieve it

Could we get all, within the budgets of time and cost ?

		Design Idea #1	Design Idea #2	Design Idea #3	Total Impact
What to achieve	Objectives	Impact on Objective	Impact on Objective	Impact on Objective	Sum of Impacts on Objectives
Cost to achieve it	Resources Time Money	Impact on Resources	Impact on Resources	Impact on Resources	Sum of Impact on Resources
Return on Investment	Benefits to Cost Ratio	Benefits / Cost	Benefits / Cost	Benefits / Cost	

PLANGUAGE SAMPLE

EFFORT

Motivation

Design Skill

Control

Scale & Meter

Expectations [The desired rewards

PERFORMANCE

Target & Benchmark

Requirements	Resources	Goal Clarity	Time
Reduce time on placing stock away	[2013-2014] Custom Monthly Report + Observation	Target: 5 minutes [Q3 – 2013]: Constraint: 30minutes	[2012]: 120 minutes ←Observation measures & report
Decrease time taken to process order request	[2013] Audit Paper Analysis & Custom Monthly Report	Target: 5 minutes [2013]: Constraint: 15 minutes per day	[2013]: 30 minutes per day ←Physical audit analysis
Decrease time taken to picking order request	[2013] Custom Monthly Report + Observation	Target: 5 minutes [2013]: Constraint: 15 minutes per day	[2012]: 120 Minutes ←Report in August & September
Reduce manual requirement for process	[2014] Observation	Target: 40% Constraint: 85%	[2013]: 100% ←Training Log Report
Increase volume of transactions per day	[2013] Custom Report	Target: 50 items Constraint: 70 items	[2012]: 387 ←Based on Observation &
Reduce time required to validate items picked	[2013] Audit paper analysis	Target: 250 per year thereafter [2013]: Constraint: 1000	[2012]: 2960 per year + ←Report in August & September
Decrease Time to Learn Process	[2013] Procedure file log	Target: 60 minutes Constraint: 120 minutes	[2012]: 180 minutes ←Training Log Report
Reduce the volume of loss productivity	[2012] Custom report	Target: 40 days Constraint: 80 days	[2012]: 162 days ←Based on absence report

Man-Chie Tse1,2 & Ravinder Singh Kahlon 1,2
{Man-Chie, Ravi}@dkode.co

Healthcare Impact Estimation

Man-Chie Tse1,2 & Ravinder Singh Kahlon 1,2
{Man-Chie, Ravi}@dkode.co

HEALTHCARE SYSTEM IMPACT ESTIMATION

	Automate Rules	Web Self Service	Decision Support	Total Impacts
Increase Transmission of Requests *(30 minutes → 10 minutes)*	10 minutes 100%	3 minutes 100%	-	200%
Decrease Number of Errors Occurring *(353 per week → 30 per week)*	100 errors 80%	< 50 90%	-	170%
Decrease Time for Processing of Requests *(70 minutes → 10 minutes)*	35 minutes 70%	-	< 10 minutes 90%	160%
Decrease Time to Learn process *(1 day → 1 hour)*	-	1 hour 100%	10 minutes 103%	203%
TOTAL DESIGN REQUIREMENT IMPACT	250%	290%	193%	

HEALTHCARE SYSTEM IMPACT ESTIMATION

	Automate Rules	Web Self Service	Decision Support	Total Impacts
Increase Transmission of Requests *(30 minutes → 10 minutes)*	10 minutes 100%	3 minutes 100%		200%
Decrease Number of Errors Occurring *(353 per week → 30 per week)*	100 errors 80%	< 50 90%		170%
Decrease Time for Processing of Requests *(70 minutes → 10 minutes)*	minutes 70%		< 10 minutes 90%	160%
Decrease Time to Learn process *(1 day → 1 hour)*	-	1 hour 100%	10 minutes 103%	203%
TOTAL DESIGN REQUIREMENT IMPACT	250%	290%	193%	
Budget (£40,000)	50% +30%	25% +10%	25% +10%	100% +50%
Time (6 months)	50% +20%	20% +10%	30% +15%	100% +40%
TOTAL BUDGET IMPACT	100% +50%	45% +20%	55% +20%	
BENEFIT TO COST RATIO	250/100 = 2.5	290/45 = 6.44	193/55 = 3.51	

Design

Objectives

Costs

Value for Money

estimated impact of a design on a critical objective

3. Strategy

- integration of IT strategy with business strategy

How to estimate the effectiveness
of any class of strategy or IT Architecture
with regard to multiple objectives
of any level of responsibility

How to understand the riskiness and credibility
of any estimates of strategy effectiveness and costs

Main Points.
All ‘strategies’/architectures/means
can have their effectiveness
estimated and measured against
any set of critical objectives.

The risk of any such ‘impact estimation’
can be determined, and quantified.

- Figure 1: Real (NON-CONFIDENTIAL version) example of an initial draft of setting the objectives that engineering processes must meet.

Business objective	Measure	Goal (200X)	Stretch goal ('0X)	Volume	Value	Profit	Cash
Time to market	Normal project time from GT to GT5	<9 mo.	<6 mo.	X		X	X
Mid-range	Min BoM for The Corp phone	<$90	[illegible]	X		X	X
Platformisation Technology	# of Technology 66 Lic. shipping > 3M/yr	4	[illegible]	X		X	X
Interface	Interface units	>11M	>13M	X		X	X
Operator preference	Top-3 operators issue RFQ spec The Corp	1	2	X		X	X
Productivity							
Get Torden	Lyn goes for Technology 66 in Sep-04	Yes		X		X	X
Fragmentation	Share of components modified	<10%	<5%		X	X	X
Commoditisation	Switching cost for a UI to another System	>1yr	[illegible]		X	X	X
Duplication	The Corp share of 'in scope' code in best-selling device	>90%	>95%		X	X	X
Competitiveness	Major feature comparison with MX	Same	Better	X		X	X
User experience	Key use cases superior vs. competition	5	10	X	X	X	X
Downstream cost saving	Project ROI for Licensees	>33%	>66%	X	X	X	X
Platformisation IFace	Number of shipping Lic.	33	55	X		X	X
Japan	Share of of XXXX sales	>50%	>60%	X		X	X
	Numbers are intentionally changed from real ones						

Business Objectives Quantified

Strategy Impact Estimation:
for a $100,000,000 Organizational Improvement Investment

Technical Strategies

Objectives

Business Objective	hardware adaptation	Telephony	Reference designs	iFace	ModUarity	Defend vs Technology 66	Tools	User Experice	GUI & Graphics	Security	Defend vs OCD	Enterprise
Time to market	20%	10%	30%	5%	15%	5%	15%	0%	0%	0%	5%	5%
Mid-range	15%	0%	15%	0%	30%	15%	5%	10%	5%	5%	0%	0%
Platformisation Technology	25%	10%	5%	0%	0%	0%	0%	5%	0%	10%	0%	5%
Interface	5%	15%	15%	0%	5%	0%	5%	0%	0%	10%	0%	10%
Operator preference	0%	10%	0%	15%	5%	20%	5%	10%	10%	20%	5%	10%
Get Torden	25%	10%	10%	0%	0%	25%	0%	10%	-20%	10%	10%	5%
Commoditisation	20%	10%	20%	10%	-20%	25%	15%	0%	0%	5%	10%	5%
Duplication	15%	10%	10%	0%	0%	40%	0%	0%	0%	5%	20%	5%
Competitiveness	10%	15%	20%	0%	10%	20%	10%	10%	20%	10%	10%	10%
User experience	5%	0%	0%	0%	0%	0%	0%	30%	10%	0%	0%	0%
Downstream cost saving	15%	5%	20%	0%	10%	20%	0%	10%	0%	0%	10%	5%
Platformisation IFace	10%	10%	20%	40%	0%	20%	5%	0%	0%	0%	0%	5%
Japan	10%	5%	20%	0%	10%	0%	0%	10%	5%	0%	0%	0%
Contribution to overall result	15%	9%	17%	4%	7%						6%	5%
Cost (£M)	£ 2.85	£ 0.49	£ 3.21	£ 2.54	£ 1.92	£ 2.31	£ 0.81	£ 1.21	£ 2.68	£ 0.79	£ 0.62	£ 0.60
ROI Index (100=average)	106	358	109	33	78	137	148	127	10	152	202	174

"Benefits"

Impacts on Objectives

Cost

Benefit/Cost ratio

358 !

Using Impact Estimation to get a quick initial picture of how the 7 Strategies ‹#› are expected to impact the 11 Objectives and 1 cost factor.

		.....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

Top-level goal	Measure	Goal (2007)	Stretch goal ('07)	Weight
Volume	Addressable market share	>40%	>50%	50 %
Value	Avg. gross margin / uni	≥$3.75	≥$4.25	30 %
Profit	Breakeven point	Q4/05	n/a	10 %
Cash	Min. cash balance w/o funding	-£50m	n/a	10 %

Business objective	Measure	Goal (2005)	Stretch goal ('05)	Volume	Value	Profit	Cash
Time to market	Normal project time from TGc to TG5	<9 mo.	<6 mo	X		X	X
Mid-range	Min BoM for Symbian phone	<$80	<$6	X		X	X
Platformisation S60	# of S60 Lic. shipping > 3M/y	2		X		X	X
CDMA	CDMA units	>1M	>3M			X	X
Operator preference	Top-3 operators issue RFQ spec Symbian	1				X	X
Productivity						X	X
Get Thunder	Thunder goes for S60 in Sep-04					X	X
Fragmentation	Share of components modified				X	X	X
Commoditisation	Switching cost for a UI to another OS				X	X	X
Duplication	Symbian share of 'in scope' code in best-selling phone				X	X	X
Competitiveness	Major feature comparison with M$					X	X
User experience	Key use cases superior vs. competition	5	1		X	X	X
Downstream cost saving	Project ROI for Licensees	>20%	>30%		X	X	X
Platformisation UIQ	Number of shipping Lic.	3				X	X
Japan	Share of of FOMA sales	>50%	>60%	X		X	X

		Valhalla Deliverables											
Business Objective	Valhalla Weight	hardware adaptation	Telephony	Reference designs	UIQ	Modularity	Defend vs S60	Tools	User Exper'ce	GUI & Graphics	Security	Defend vs OCD	Enterprise
Time to market	15 %	20 %	10 %	30 %	5 %	10 %	5 %	15 %					5 %
Mid-range	15 %	15 %	0 %	15 %	0 %	30 %	15 %	5 %					0 %
Platformisation S60	10 %	25 %	10 %	30 %	0 %	0 %	10 %	0 %					5 %
CDMA	5 %	5 %	15 %	15 %	0 %	5 %	0 %	5 %					10 %
Operator preference	5 %	0 %	10 %	0 %	15 %	5 %	20 %	5 %					10 %
Get Thunder	3 %	25 %	10 %	10 %	-10 %	0 %	20 %	0 %					5 %
Commoditisation	5 %	20 %	10 %	20 %	10 %	-20 %	25 %	15 %					5 %
Duplication	10 %	15 %	10 %	10 %	0 %	0 %	40 %	0 %					5 %
Competitiveness	5 %	10 %	15 %	20 %	0 %	10 %	20 %	10 %					10 %
User experience	10 %	5 %	0 %	0 %	0 %	20 %	0 %	0 %					0 %
Downstream cost saving	8 %	15 %	5 %	20 %	0 %	10 %	20 %	0 %					5 %
Platformisation UIQ	3 %	10 %	10 %	20 %	40 %	0 %	20 %	5 %					5 %
Japan	6 %	10 %	5 %	20 %	0 %	10 %	0 %	0 %					0 %
Contribution to overall result		14 %	7 %	17 %	3 %	9 %	14 %	5 %	8 %	3 %	5 %	5 %	4 %
Cost (£M)		£ 2,85	£ 0,49	£ 3,21	£ 2,54	£ 1,92	£ 2,31	£ 0,81	£ 1,21	£ 2,68	£ 0,79	£ 0,62	£ 0,60
ROI Index (100=average)		106	312	113	24	103	128	128	134	23	129	174	148

Corporate

Marketing

Impacts

Up

Technical

Symbian company level (operational board perspective)

			Medium-term (2005)			Long-term (2007)		
Fundamental objectives	Measure	Current level (H104)	Pain level (2005)	Goal level (2005)	Stretch level ('05)	Pain level (2007)	Goal level (2007)	Stretch level ('07)
Momentum	YoY growth of units sold	135 %	75 %	100 %	150 %	75 %	100 %	150 %
Winning	Relative market share to second largest open OS	15x	5x	10x	15x%	1x	2x	3x
Value	Avg. gross margin / unit	$4.72	<$3.75	≥$4.37	≥$4.72	$2.50	≥$3.75	≥$4.25
Profit	Breakeven point	Q3/05		Q4/05	n/a	2007	Q4/05	n/a
Cash	Min. cash balance w/o funding	-£21m	<-£21			<-£50m	-£50m	n/a

			Medium-term (2005)			Long-term (2007)			Impacts top-level objectives directly				
Strategic objectives	Measure	Current level (H104)	Pain level (2005)	Goal level (2005)	Stretch level ('05)	Pain level (2007)	Goal level (2007)	Stretch level ('07)	Momentum	Winning	Value	Profit	Cash
Device time to market	'Normal project' time from first call to mass production	12-15 mo.	>12 mo.	<9 mo.	<6 mo.	>9 mo.	<6 mo.	<3 mo.	X	X		X	X
Mid-range	Lowest BoM shipping Symbian phone	$138	>$100	<$80	<$60	>$80	<$60	<$50	X	X		X	X
Platformisation	# of Lic. shipping > 2M/yr	1	1	3	5	3	5	8	X	X		X	X
CDMA2000	CDMA2000 units	0	<1M	>1M	>3M	<3M	>10M	>15M	X	X		X	X
Operator preference	Operators platformising on Symbian OS	1	0	1	2	1	3	5	X	X		X	X
Productivity	Avg. cost per PREQ implemented	£60-80k	>£70k	<£50k	<£40k	>£70k	<£50k	<£40k				X	X
Enterprise	# of devices used to access enterprise apps remotely	100k	<1M	>3M	>5M	<3M	>8M	>20M	X	X	X		
Thunder	Thunder volumes as % of Thunder forecast	n/a	<50%	100 %	>100%	n/a	n/a	n/a	X	X		X	X
Fragmentation	Overhead incurred by partners to port between UI1 & UI2	15 %	>20%	<10%	<5%	>20%	<10%	<5%			X		
Commoditisation	Switching cost for a UI to another OS	?	< 6 mo.	>1 yr.	>2 yrs.	< 6 mo.	>1 yr.	>2 yrs.		X	X		
Competitiveness of OS	Comparative reviews of flagship Symbian device vs others	Bet/Wor	Wor/Wor	Bet/Sam	Bet/Bet	Wor/Wor	Bet/Bet	n/a		X			
Agility	Average age of new PREQs in last release (PREQ-MS4c)	24 mo.	24 mo.	18 mo.	12 mo.	18 mo.	12 mo.	9 mo.	X				
Integration cost	Number of engineers on a normal/lead project	50-200	>50	30	15	>50	30	15	X	X	X		
3G	Share of W-CDMA devices shipping ww	25 %	<25%	>35%	>45%	<25%	>35%	>45%	X	X			
Japan	Share of of FOMA sales	29 %	<30%	>40%	>50%	<30%	>40%	>50%	X	X		X	X

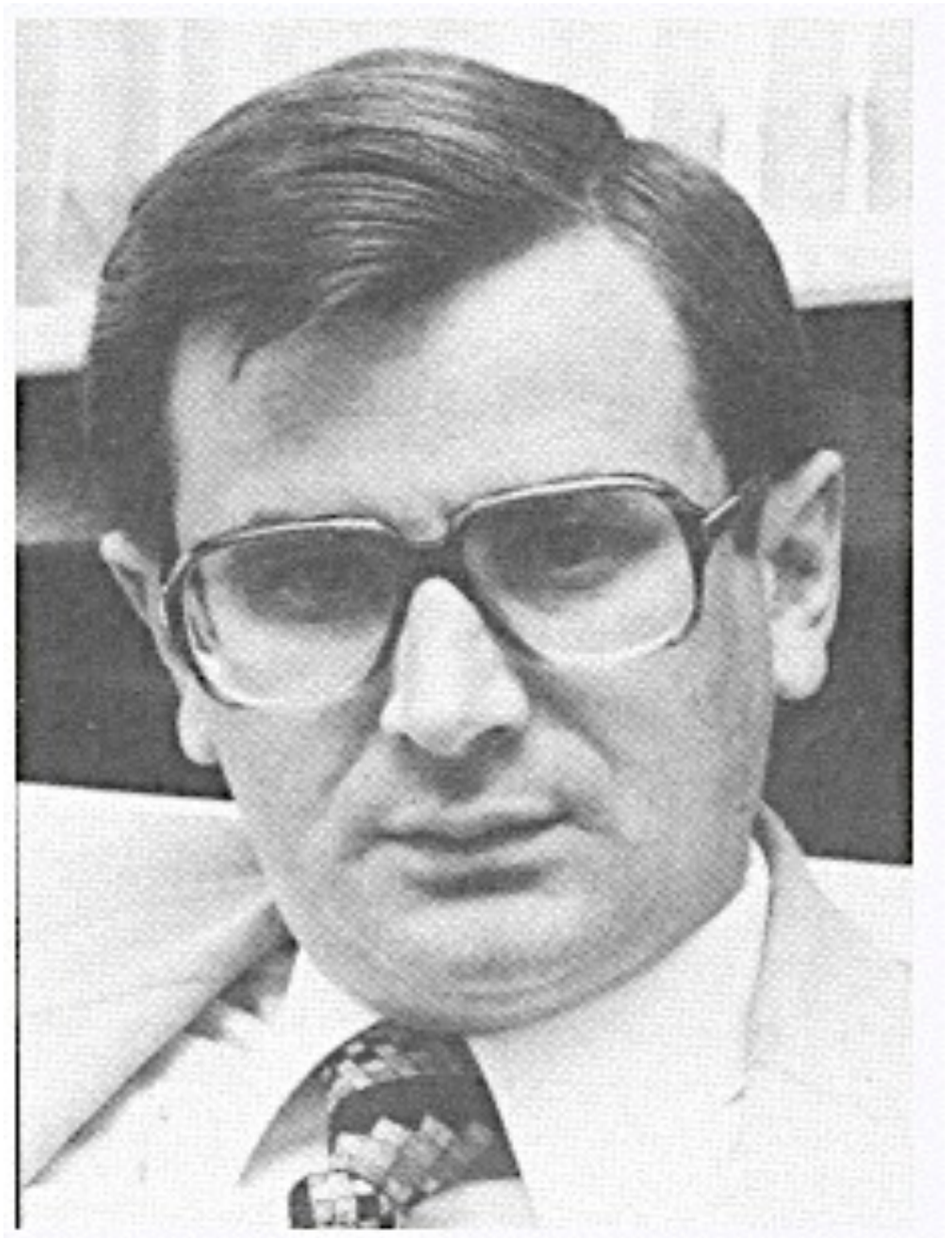
Valhalla level

Yellow % indicated extent to which Valhalla deliverable will contribute to bridging the gap between actual and 'goal' level performance on selected business objectives in access of 'going concern'		Ability to achieve goal w/o Valhalla	Valhalla Deliverables												Valhalla contribution to gap	Expected delivery on goal with Valhalla
Strategic objectives	Measure		hardware adaptation	Telephony	Reference designs	UIQ	Modularity	Defend vs S60	Tools	User Exper'ce	GUI & Graphics	Security	Defend vs OCD	Enterprise		
Device time to market	'Normal project' time from first call to mass production	33 %	20 %	5 %	20 %	5 %	10 %	5 %	20 %	0 %	0 %	0 %	5 %	5 %	95 %	97 %
Mid-range	Lowest BoM shipping Symbian phone	75 %	20 %	0 %	15 %	0 %	15 %	0 %	5 %	10 %	5 %	5 %	0 %	0 %	75 %	94 %
Platformisation	# of Lic. shipping > 2M/yr	50 %	15 %	5 %	30 %	10 %	5 %	10 %	5 %	5 %	5 %	10 %	0 %	5 %	105 %	103 %
CDMA2000	CDMA2000 units	75 %	5 %	10 %	15 %	0 %	5 %	0 %	5 %	0 %	5 %	10 %	0 %	10 %	65 %	91 %
Operator preference	Operators platformising on Symbian OS	75 %	0 %	5 %	5 %	25 %	0 %	10 %	5 %	10 %	15 %	20 %	5 %	10 %	110 %	103 %
Productivity	Avg. cost per PREQ implemented	75 %													0 %	75 %
Enterprise	# of devices used to access enterprise apps remotely	50 %	0 %	5 %	0 %	5 %	10 %	0 %	10 %	5 %	0 %	0 %	0 %	20 %	55 %	78 %
Thunder	Thunder volumes as % of Thunder forecast	100 %													0 %	100 %
Fragmentation	Overhead incurred by partners to port between UI1 & UI2	50 %	10 %	5 %	5 %	10 %	0 %	30 %	10 %	0 %	-10 %	10 %	15 %	5 %	90 %	95 %
Commoditisation	Switching cost for a UI to another OS	50 %	20 %	10 %	15 %	10 %	-20 %	25 %	15 %	0 %	0 %	5 %	10 %	0 %	90 %	95 %
Competitiveness of OS	Comparative reviews of flagship Symbian device vs others	50 %	5 %	0 %	0 %	0 %	10 %	0 %	0 %	20 %	30 %	5 %	0 %	0 %	70 %	85 %
Agility	Average age of new PREQs in last release (PREQ-MS4c)	75 %	5 %	5 %	0 %	0 %	5 %	10 %	5 %	0 %	5 %	5 %	10 %	5 %	55 %	89 %
Integration cost	Number of engineers on a normal/lead project	50 %	15 %	5 %	20 %	5 %	10 %	5 %	10 %	0 %	0 %	0 %	10 %	5 %	85 %	93 %
3G	Share of W-CDMA devices shipping ww	66 %	10 %	15 %	20 %	5 %	5 %	0 %	0 %	10 %	15 %	5 %	15 %	5 %	105 %	102 %
Japan	Share of of FOMA sales	66 %	10 %	5 %	20 %	0 %	10 %	0 %	0 %	10 %	5 %	0 %	0 %	0 %	60 %	86 %
Contribution to overall resul	(unweighted)		135 %	75 %	165 %	75 %	65 %	95 %	90 %	70 %	75 %	75 %	70 %	70 %	1060 %	92 %
Annual cost (£M)			£ 2,34	£ 1,47	£ 2,63	£ 2,08	£ 1,58	£ 4,04	£ 1,20	£ 0,99	£ 2,20	£ 0,65	£ 0,51	£ 0,49	£ 20,18	
Performance to cost ratio (indexed to average=100)			110	97	119	69	78	45	143	134	65	221	262	271	0,53	

Remaining Issues:	
Not all business objectives are equally important, do we weigh them?	Try to give achiving the goal a monetary value, also consider level of risk and authority of stakeholders
Not all measures easy to track (eg. number of engineers on projects)	Define meters, can be dynamic
How to link business objectives to top-level goals (eg no £ value of business objectives)	See 1

ICL Case Study

BCS June 12 Lecture 2015
Slides are at
http://www.gilb.com/dl846



Robb Wilmot. ICL's energetic new MD

ICL Objectives

Competitiveness	
Growth	
Profitability	
Market Share	
Brand Admiration	
Viability	***Share Price up 100% within 3 years***

Impact Estimation: ICL Objectives versus Robb's Strategies (very rough approximation to show principles)

	One Per Desk	Mid Range	Large Scale Fujitsu	∑ impact
Competitive-ness	***10%***	***20%***	***15%***	***45%***
Growth	***30%***	***35%***	***5%***	***70%***
Profitability	***45%***	***30%***	***50%***	***125%***
Market Share	***20%***	***40%***	***5%***	***65%***
Brand Admiration	***30%***	***30%***	***25%***	***85%***
Viability	***5%***	***40%***	***30%***	***75%***
Sum ∑	***150%***	***195%***	***130%***	

US DoD. Persinscom **Impact EstimationTable**:

Designs

Requirements

	Technology Investment	*Business Practices*	*People*	*Empowerment*	*Principles of IMA Management*	*Business Process Re-engineering*	*Sum Requirements*
Customer Service ? <->0 Violation of agreement	50%	1	5%	5%	5%	60%	185%
Availability 90% <-> 99.5% Up time	50%		5–10%	0%	0%	200%	265%
Usability 200 <-> 60 Requests by Users			5–10%	50%	0%	10%	130%
Responsiveness 70% <-> ECP's on time	50%	10%	90%	25%	5%	50%	180%
Productivity	45%						303%
3:1 Return on Investment	50%						251%
Morale 72 <-> 60 per month on Sick Leave							
Data Integrity 88% <-> 97% Data Error %	42%						177%
Technology Adaptability 75% Adapt Technology	5%						160%
Requirement Adaptability ? <-> 2.6% Adapt to Change	80%						260%
Resource Adaptability 2.1M <-> ? Resource Change	10%	80%	5%	50%	50%	75%	270%
Cost Reduction FADS <-> 30% Total Funding	50%	40%	10%	40%	50%	50%	240%
Sum of Performance	*482%*	*280%*	*305%*	*390%*	*315%*	*649%*	
Money % of total budget	15%	4%	3%	4%	6%	4%	36%
Time % total work months/year	15%	15%	20%	10%	20%	18%	98%
Sum of Costs	*30*	*19*	*23*	*14*	*26*	*22*	
Performance to Cost Ratio	*16:1*	*14:7*	*13:3*	*27:9*	*12:1*	29.5 :1	

Estimated Impact of Design -> Requirements

US Army Example: PERSINSCOM: Personnel System

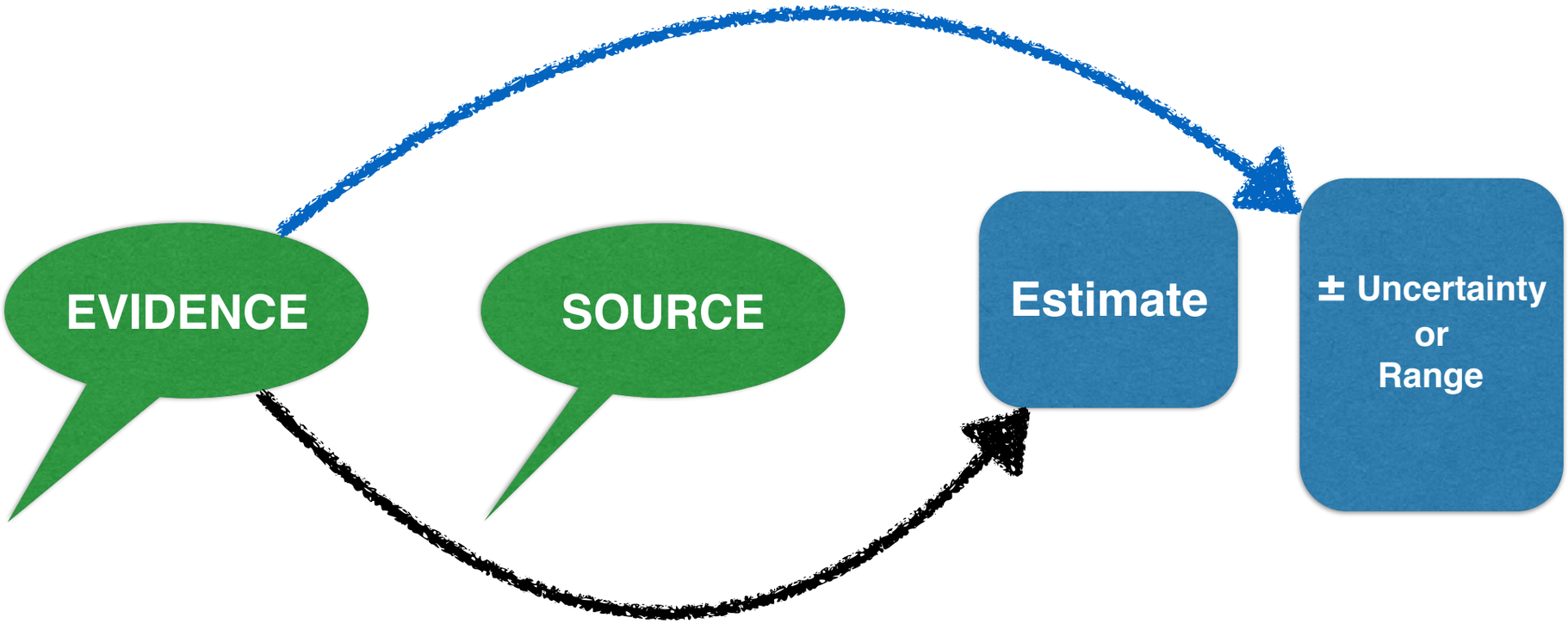
STRATEGIES ➔ **OBJECTIVES**	Technology Investment	Business Practices	People	Empow-erment	*Principles of IMA Management*	Business Process Re-engineering	SUM
Customer Service ?➔0 Violation of agreement	50%	10%	5%	5%	5%	60%	185%
Availability 90% ➔ 99.5% Up time	50%	5%	5-10%	0	0	200%	265%
Usability 200 ➔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

Risk Analysis

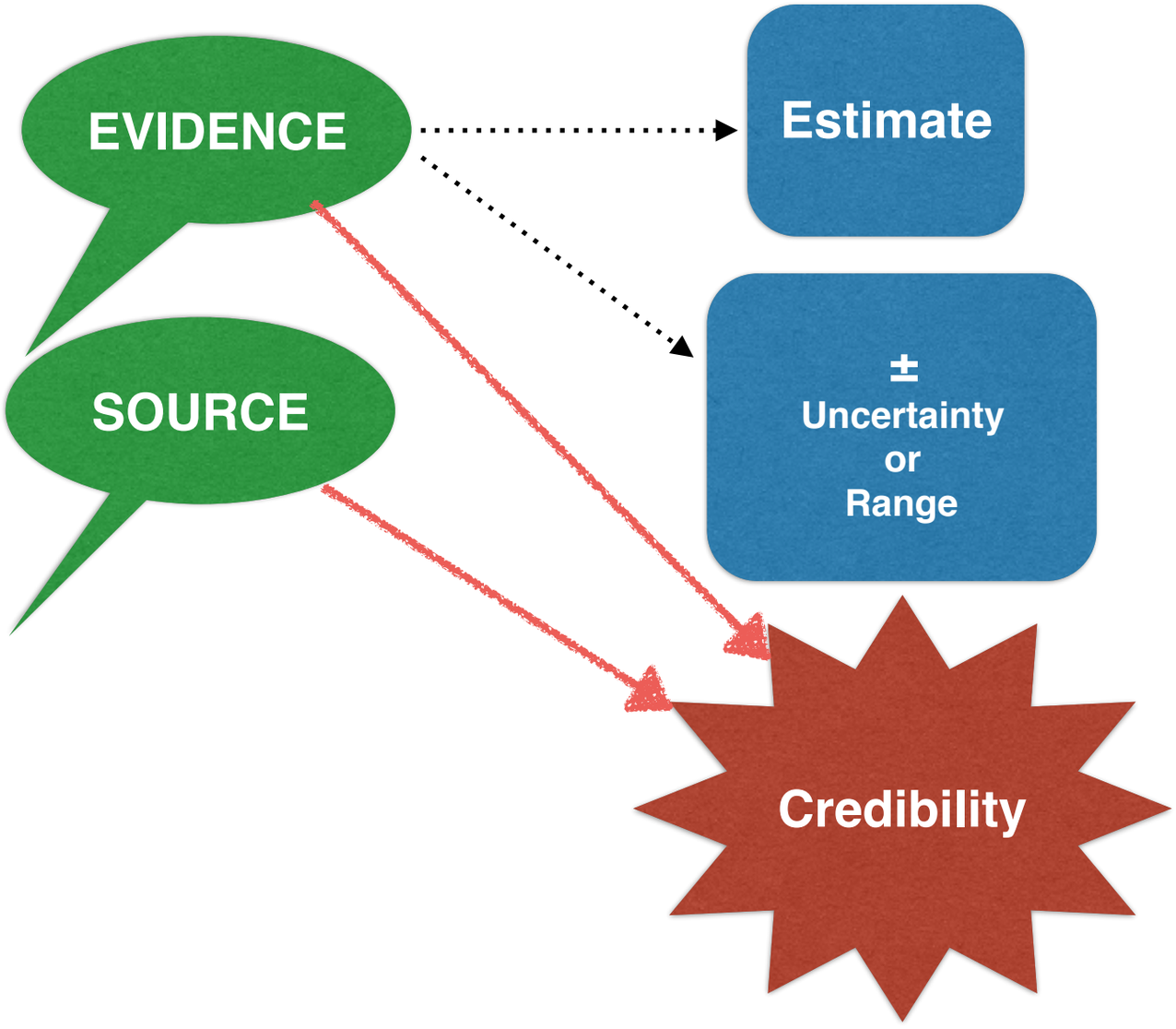
EVIDENCE
+ SOURCE
= CREDIBILITY

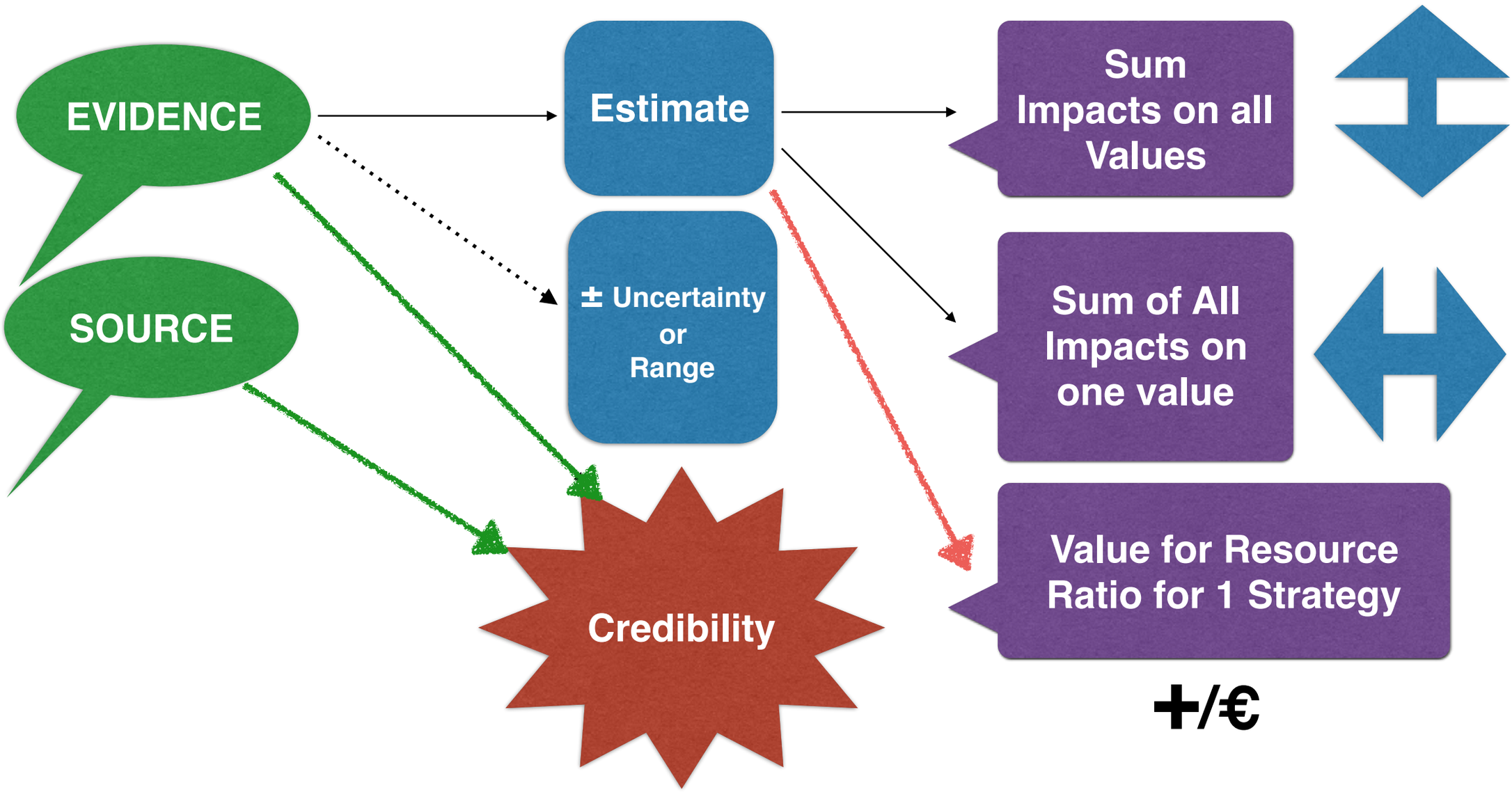
VP Book 9.5 A

Basic Factors

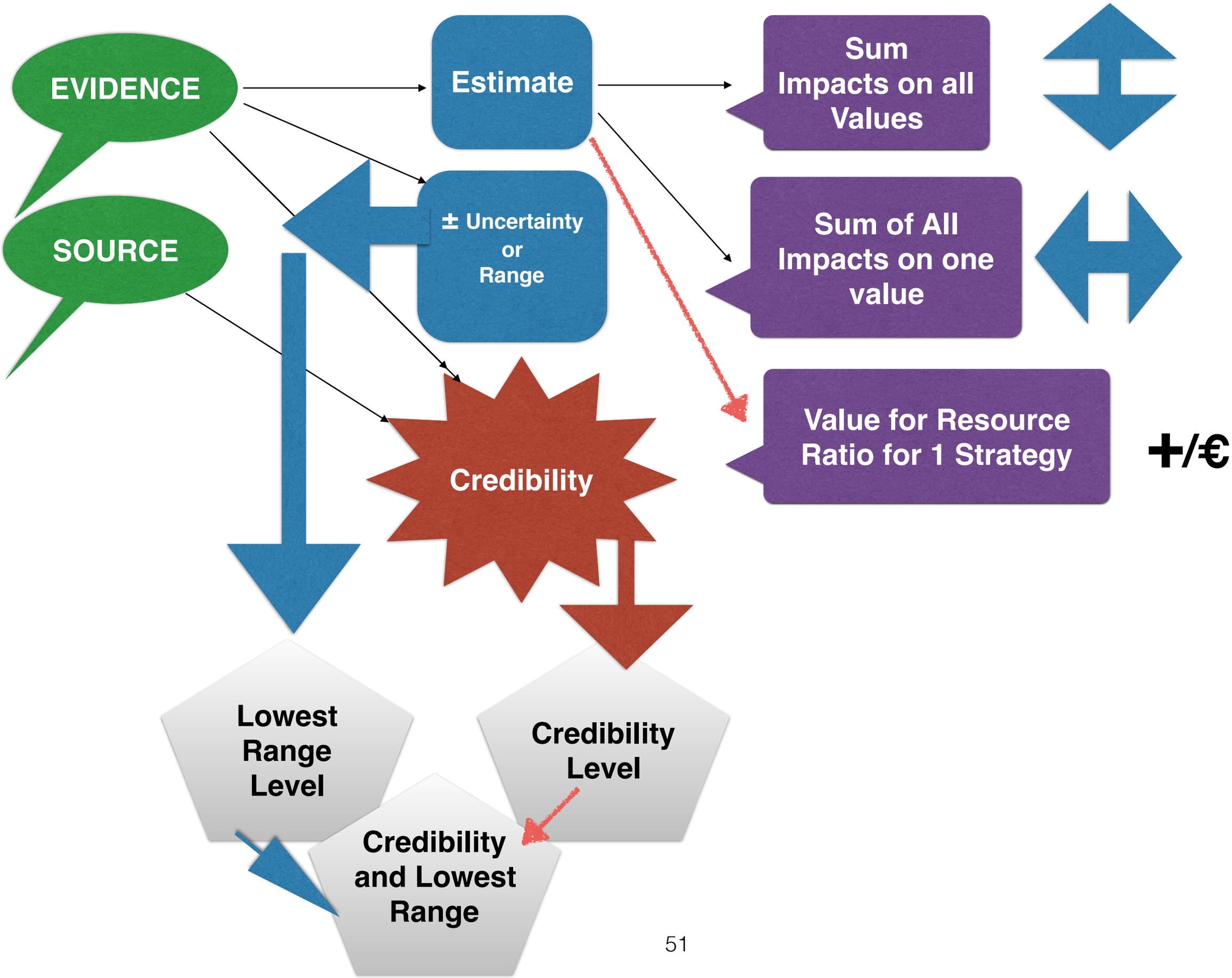


9.5 B Basic Derivatives




EVIDENCE
SOURCE
Estimate
± Uncertainty or Range
Credibility
Sum Impacts on all Values
Sum of All Impacts on one value
Value for Resource Ratio for 1 Strategy
+/€

9.5 D Risk Summaries



± Uncertainty

Range of possible impacts

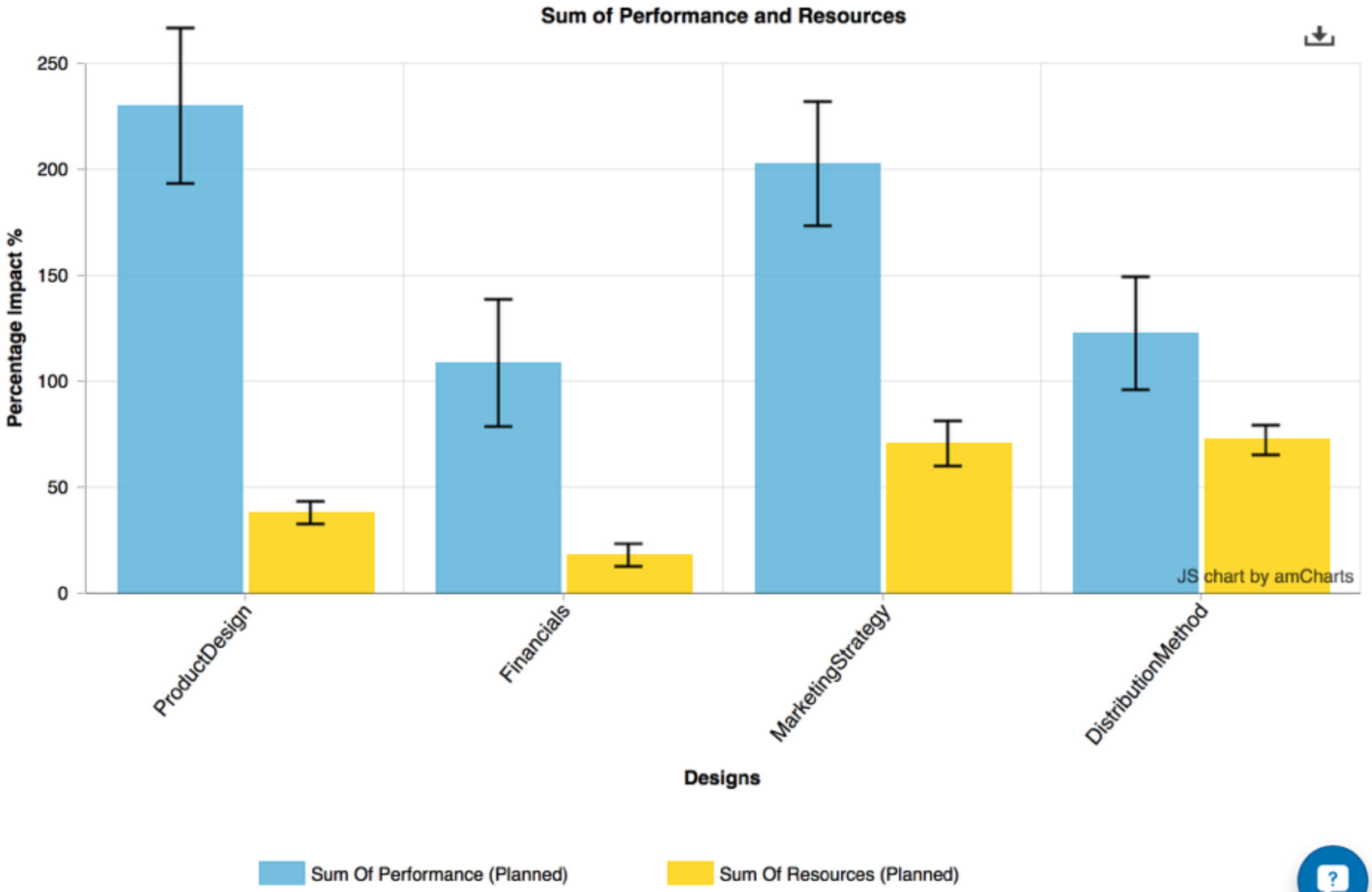
app.needsandmeans.com/iet/IET-AFGGXOT

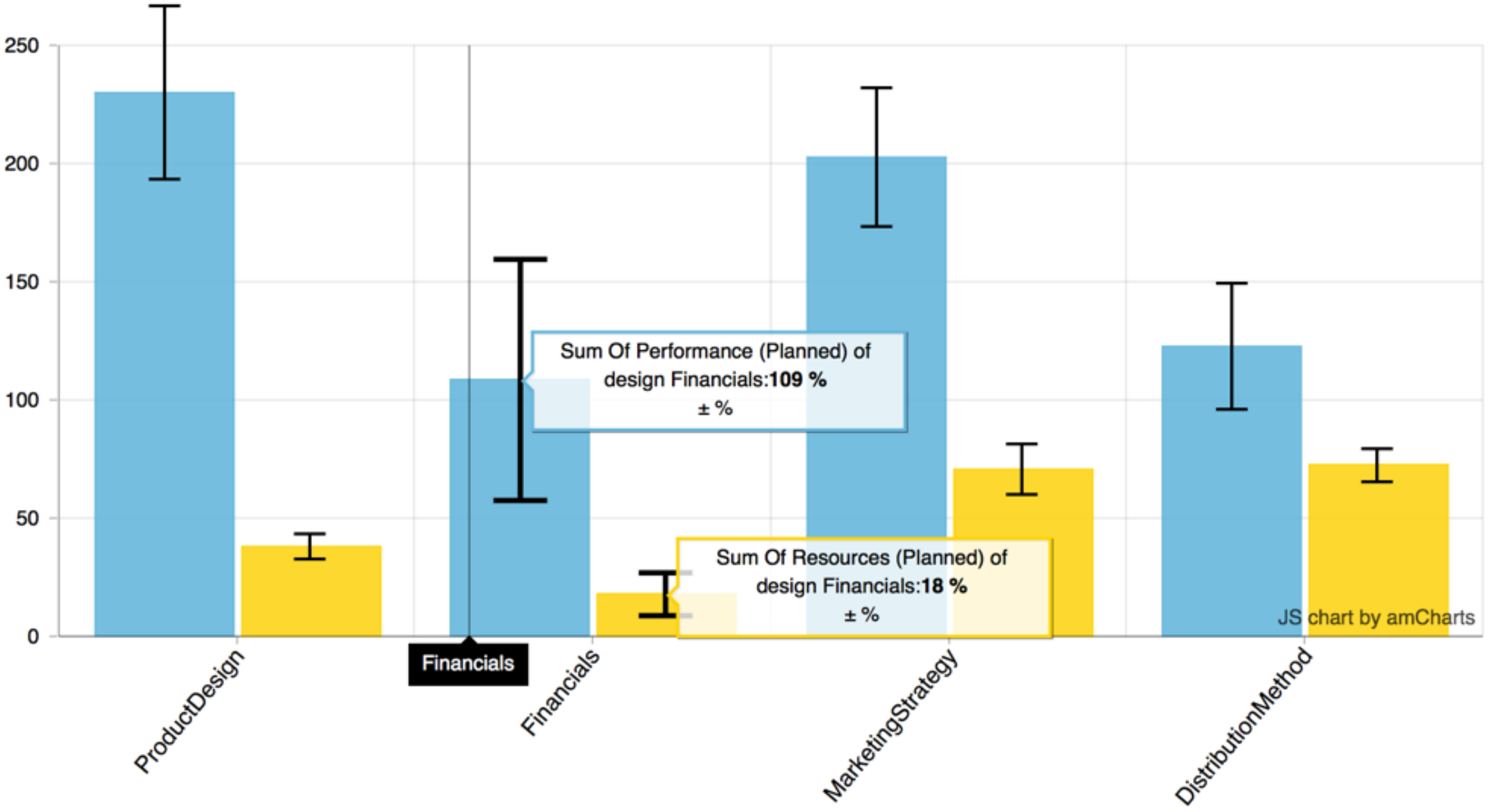
Untitled

Requirements	ProductDesign	Financials	MarketingStrategy	DistributionMethod	Sum
Demographic Past: **0** → Wish: **50** %	20 ± 5 % 0 Δ%: **40** ± 10 % 40	27 ± 5 % 0 Δ%: **54** ± 10 % 94	23 ± 3 % 0 Δ%: **46** ± 6 % 140	10 ± 0 % 0 Δ%: **20** ± 0 % 160	Σ Δ%: **160** ± 26 %
Millionaire Past: **1** → Wish: **1000000** $	450000 ± 15000 $ 0 Δ%: **45** ± 15 % 45	400000 ± 10000 $ 0 Δ%: **40** ± 10 % 85	100000 ± 50000 $ 0 Δ%: **10** ± 5 % 95	200000 ± 10000 $ 0 Δ%: **20** ± 10 % 115	Σ Δ%: **115** ± 40 %
MarketSegment Past: **4** → Wish: **1** Market Rank	1 ± 1 Market... 0 Δ%: **100** ± 33 % 100	4 ± 1 Market... 0 Δ%: **0** ± 33 % 100	2 ± 1 Market... 0 Δ%: **67** ± 33 % 167	3 ± 1 Market... 0 Δ%: **33** ± 33 % 200	Σ Δ%: **200** ± 132 %
Geography Past: **0** → Wish: **100** %	5 ± 5 % 0 Δ%: **5** ± 5 % 5	10 ± 4 % 0 Δ%: **10** ± 4 % 15	40 ± 5 % 0 Δ%: **40** ± 5 % 55	30 ± 5 % 0 Δ%: **30** ± 5 % 85	Σ Δ%: **85** ± 19 %
Market Past: **0** → Wish: **100** %	40 ± 10 % 0 Δ%: **40** ± 10 % 40	5 ± 3 % 0 Δ%: **5** ± 3 % 45	40 ± 10 % 0 Δ%: **40** ± 10 % 85	20 ± 5 % 1 Δ%: **20** ± 5 % 105	Σ Δ%: **105** ± 28 %
Sum Of Performance:	Σ%: **230** ± 73 % 230	Σ%: **109** ± 60 % 339	Σ%: **203** ± 59 % 542	Σ%: **123** ± 53 % 665	
TimeToMarket Past: **1** → Wish: **8** Weeks	2 ± 0.5 Weeks 0 Δ%: **14** ± 7 % 14	2 ± 0.5 Weeks 0 Δ%: **14** ± 7 % 28	3 ± 0.75 Weeks 0 Δ%: **29** ± 11 % 57	4 ± 1 Weeks 0 Δ%: **43** ± 14 % 100	Σ Δ%: **100** ± 39 %
ShowMeTheMoney Past: **0** → Wish: **5005** £	1200 ± 200 £ 0 Δ%: **24** ± 4 % 24	205 ± 200 £ 0 Δ%: **4** ± 4 % 28	2100 ± 500 £ 0 Δ%: **42** ± 10 % 70	1500 ± 0 £ 0 Δ%: **30** ± 0 % 100	Σ Δ%: **100** ± 18 %
Sum Of Resources:	Σ%: **38** ± 11 % 38	Σ%: **18** ± 11 % 56	Σ%: **71** ± 21 % 127	Σ%: **73** ± 14 % 200	
Performance To Cost:	6.05	6.06	2.86	1.68	
Ratio (Worst Case)	3.20	1.69	1.57	0.80	

157/49 = 3.2

Settings... | + Add to table | Sort designs




250
200
150
100
50
0
Sum Of Performance (Planned) of design Financials:109 %
± %
Sum Of Resources (Planned) of design Financials:18 %
± %
JS chart by amCharts
ProductDesign
Financials
Financials
MarketingStrategy
DistributionMethod

Credibility

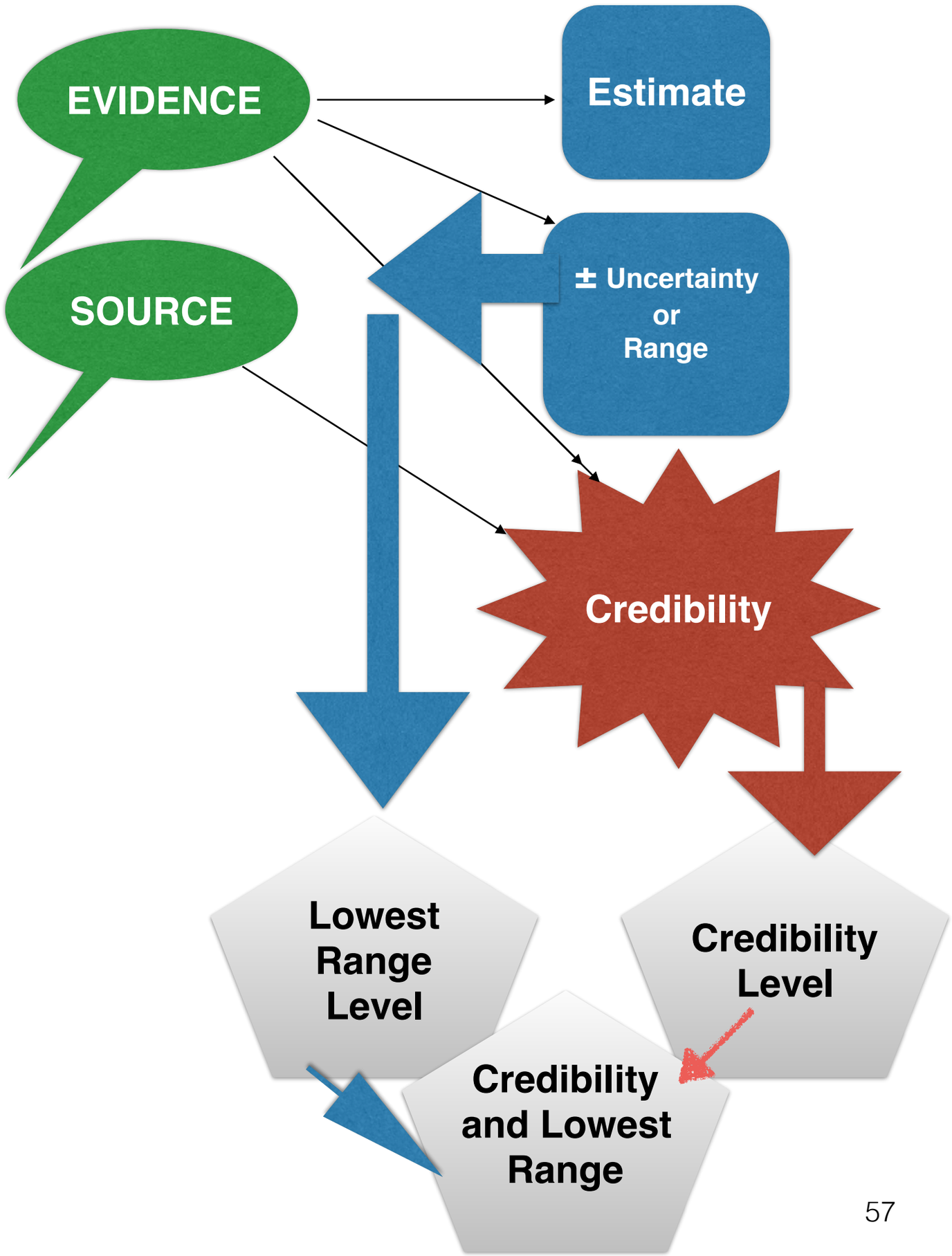

EVIDENCE
SOURCE
Estimate
± Uncertainty or Range
Credibility
Lowest Range Level
Credibility and Lowest Range
Credibility Level

Table 9.3 Example of a Credibility Ratings Table

Credibility Rating	*Meaning*
0.0	Wild guess, no credibility
0.1	We know it has been done somewhere
0.2	We have one measurement somewhere
0.3	There are several measurements in the estimated range
0.4	The several measurements are relevant to our case
0.5	The method used to obtain the several relevant measurements is considered reliable
0.6	We have used the method/design/idea/strategy in-house
0.7	We have reliable measurements for the design idea in-house
0.8	Reliable in-house measurements correlate to independent external measurements
0.9	We have used the idea on this project and measured it (Evo step, pilot and field trial)
1.0	Perfect credibility, we have rock solid, contract-guaranteed, long-term and credible experience with this idea on this project and, the results are unlikely to disappoint us

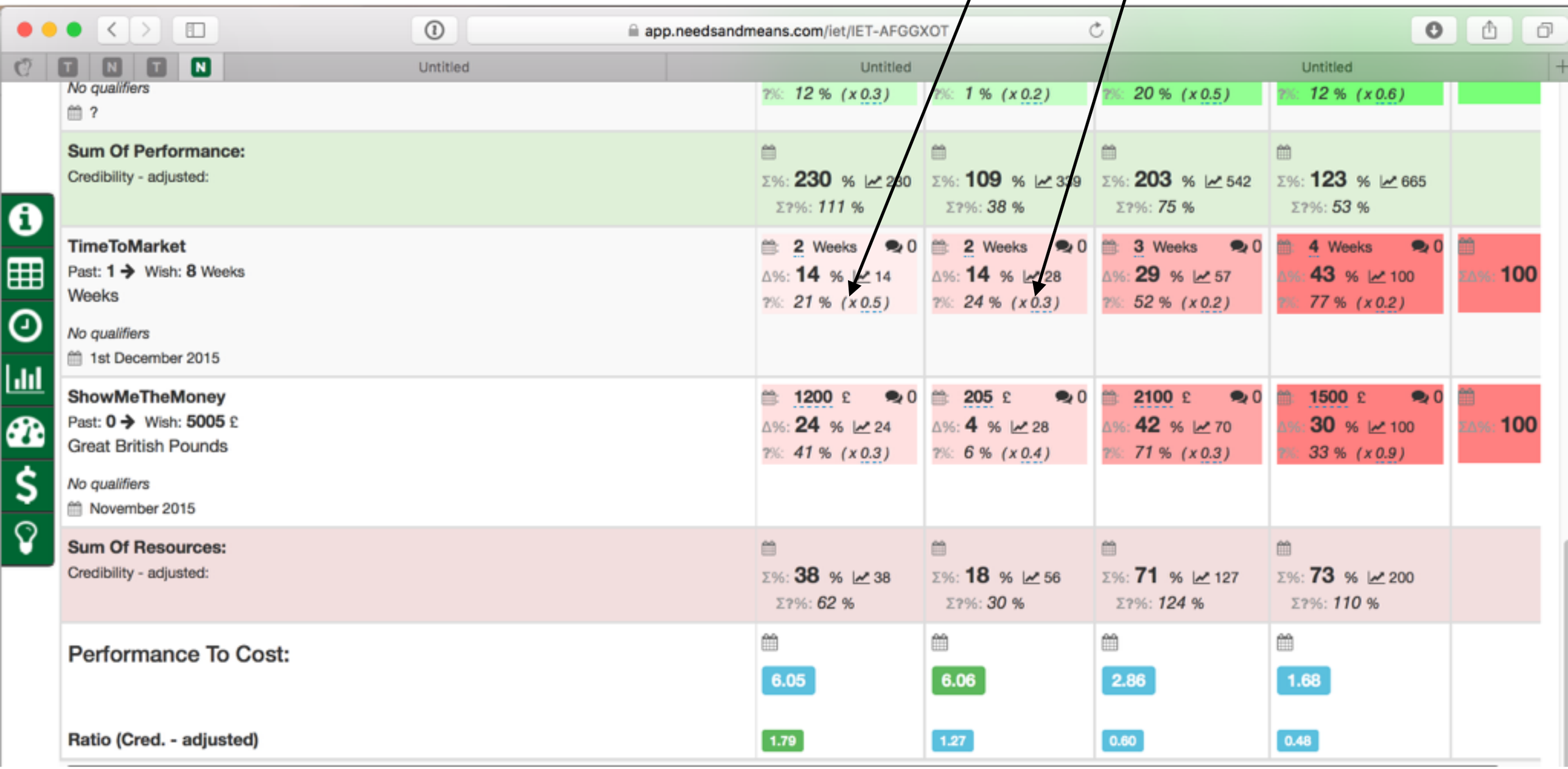
Performance Factors (Top of Table) with CREDIBILITY FACTOR

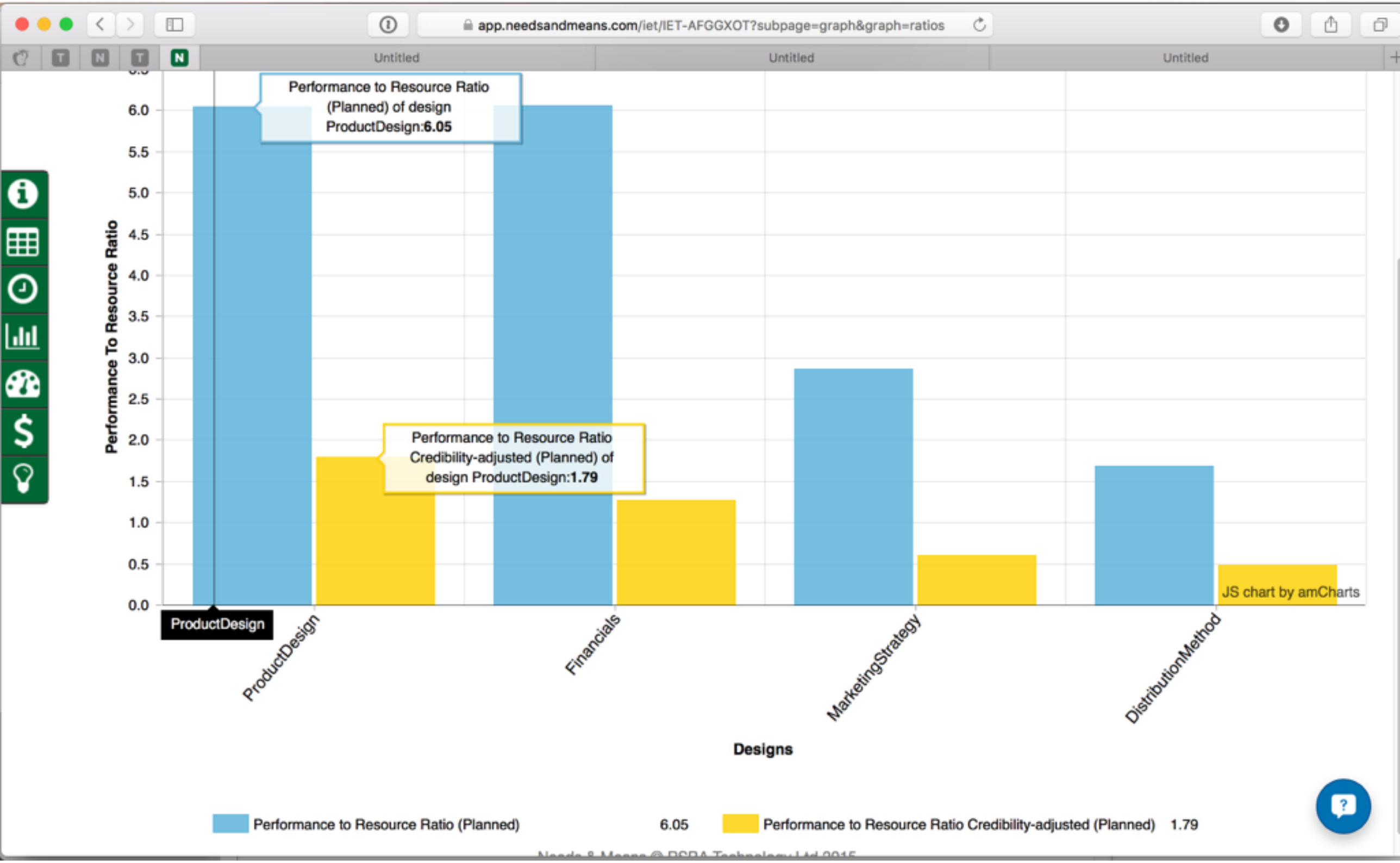
app.needsandmeans.com/iet/IET-AFGGXOT

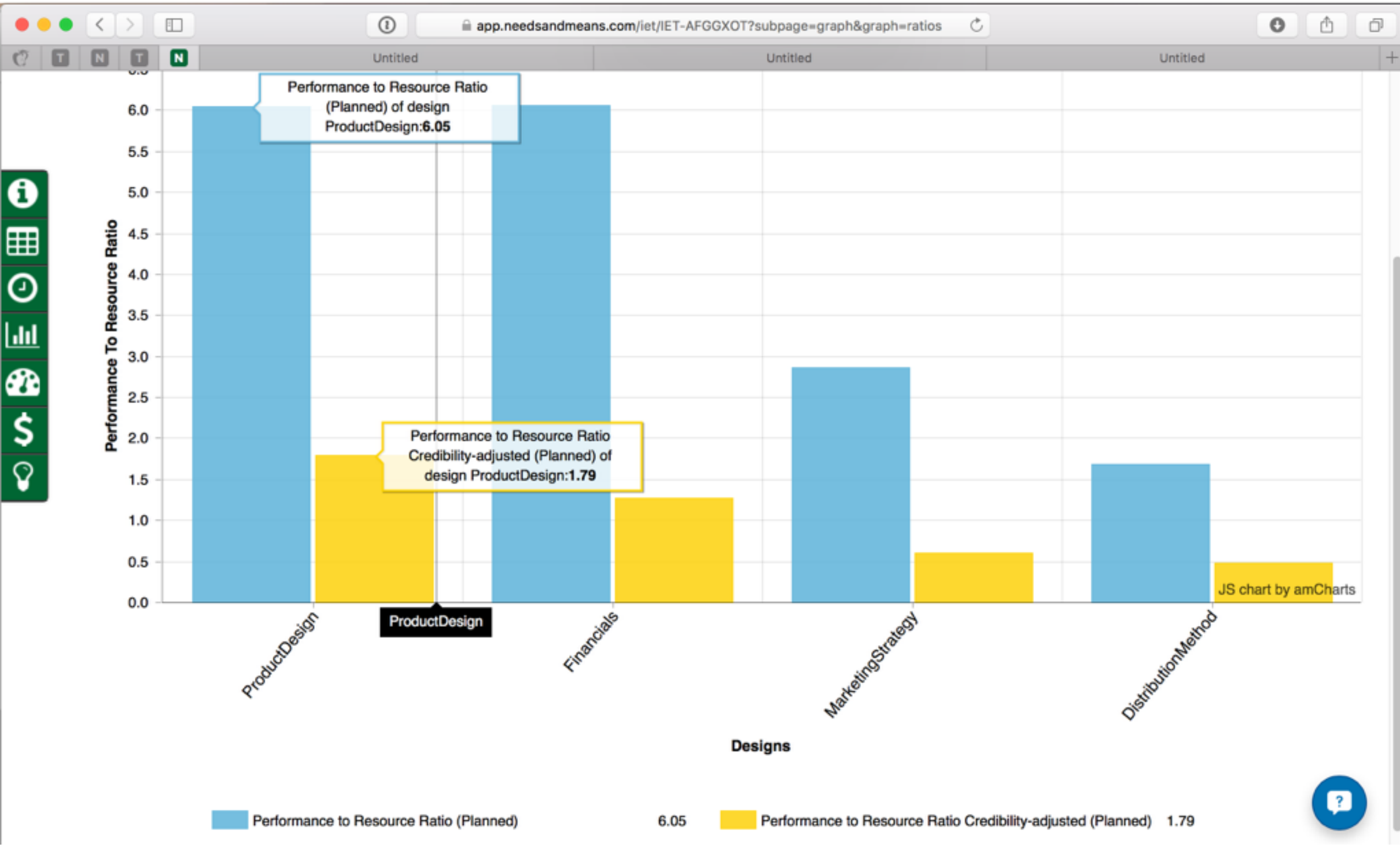
Untitled | Untitled | Untitled

Requirements	ProductDesign	Financials	MarketingStrategy	DistributionMethod	Sum
Demographic Past: **0** → Wish: **50** % %Market Share of this Demographic *No qualifiers* 11th November 2015	20 % 0 Δ%: 40 % 40 ?%: 12 % (x 0.3)	27 % 0 Δ%: 54 % 94 ?%: 5 % (x 0.1)	23 % 0 Δ%: 46 % 140 ?%: 9 % (x 0.2)	10 % 0 Δ%: 20 % 160 ?%: 4 % (x 0.2)	Σ Δ%: 160
Millionaire Past: **1** → Wish: **1000000** $ Dollars and Cents *No qualifiers* December 2016	450000 $ 0 Δ%: 45 % 45 ?%: 5 % (x 0.1)	400000 $ 0 Δ%: 40 % 85 ?%: 28 % (x 0.7)	100000 $ 0 Δ%: 10 % 95 ?%: 4 % (x 0.4)	200000 $ 0 Δ%: 20 % 115 ?%: 8 % (x 0.4)	Σ Δ%: 115
MarketSegment Past: **4** → Wish: **1** Market Rank Ranking, in terms of Sales Volume Quantity for defined [Products] and defined [Markets]. [Products = A, Markets = UK] End 2015	1 Marke... 0 Δ%: 100 % 100 ?%: 80 % (x 0.8)	4 Market... 0 Δ%: 0 % 100 ?%: 0 % (x 0.3)	2 Market... 0 Δ%: 67 % 167 ?%: 13 % (x 0.2)	3 Market... 0 Δ%: 33 % 200 ?%: 17 % (x 0.5)	Σ Δ%: 200
Geography Past: **0** → Wish: **100** % *No qualifiers* ?	5 % 0 Δ%: 5 % 5 ?%: 2 % (x 0.4)	10 % 0 Δ%: 10 % 15 ?%: 4 % (x 0.4)	40 % 0 Δ%: 40 % 55 ?%: 28 % (x 0.7)	30 % 0 Δ%: 30 % 85 ?%: 12 % (x 0.4)	Σ Δ%: 85
Market Past: **0** → Wish: **100** % *No qualifiers* ?	40 % 0 Δ%: 40 % 40 ?%: 12 % (x 0.3)	5 % 0 Δ%: 5 % 45 ?%: 1 % (x 0.2)	40 % 0 Δ%: 40 % 85 ?%: 20 % (x 0.5)	20 % 1 Δ%: 20 % 105 ?%: 12 % (x 0.6)	Σ Δ%: 105
Sum Of Performance: Credibility - adjusted:	Σ%: 230 % 230 Σ?%: 111 %	Σ%: 109 % 339 Σ?%: 38 %	Σ%: 203 % 542 Σ?%: 75 %	Σ%: 123 % 665 Σ?%: 53 %	

Resources (Lower part of Table) WITH Credibility Factor




app.needsandmeans.com/iet/IET-AFGGXOT?subpage=graph&graph=ratios
Untitled
Untitled
Untitled
Performance to Resource Ratio (Planned) of design ProductDesign:6.05
Performance to Resource Ratio Credibility-adjusted (Planned) of design ProductDesign:1.79
Performance To Resource Ratio
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
ProductDesign
ProductDesign
Financials
MarketingStrategy
DistributionMethod
JS chart by amCharts
Designs
Performance to Resource Ratio (Planned) 6.05
Performance to Resource Ratio Credibility-adjusted (Planned) 1.79


app.needsandmeans.com/iet/IET-AFGGXOT?subpage=graph&graph=ratios
Untitled
Untitled
Untitled
Performance to Resource Ratio (Planned) of design ProductDesign:6.05
Performance to Resource Ratio Credibility-adjusted (Planned) of design ProductDesign:1.79
Performance To Resource Ratio
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
ProductDesign
ProductDesign
Financials
MarketingStrategy
DistributionMethod
JS chart by amCharts
Designs
Performance to Resource Ratio (Planned) 6.05
Performance to Resource Ratio Credibility-adjusted (Planned) 1.79

4. IT governance:

investment decisions, business case, and prioritisation etc.

See benefits/value (2) above.

Dynamic Prioritisation of investments and strategies depending on changing objectives and resources

Understanding and managing risks with decisions, in a complex culture of technology, business and international considerations

Main Points.

Resource prioritisation should not be static and up front. Prioritization must be re-evaluated frequently based on achievement of objectives and depletion of limited resources.

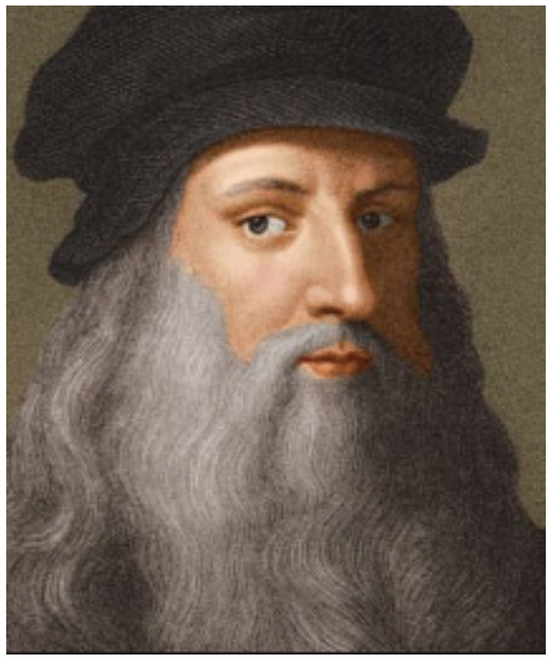
Risk evaluation is a constant and detailed planning and evaluation process.

"Dynamic Design to Cost for Value (DDtCV):

copes with imposed deadlines and fixed prices."

Tom Gilb and Kai Gilb
gilb.com
@ImTomGilb
These slides are at: **gilb.com/dl858**
Workshop at 'Smidig' (Agile) Conference, Oslo Monday 2 November 2015, 13:15-14:00
tom@gilb.com, kai@gilb.com
http://tinyurl.com/AGILEMYTHS

Leonardo da Vinci 1452-1519



Life is pretty simple:

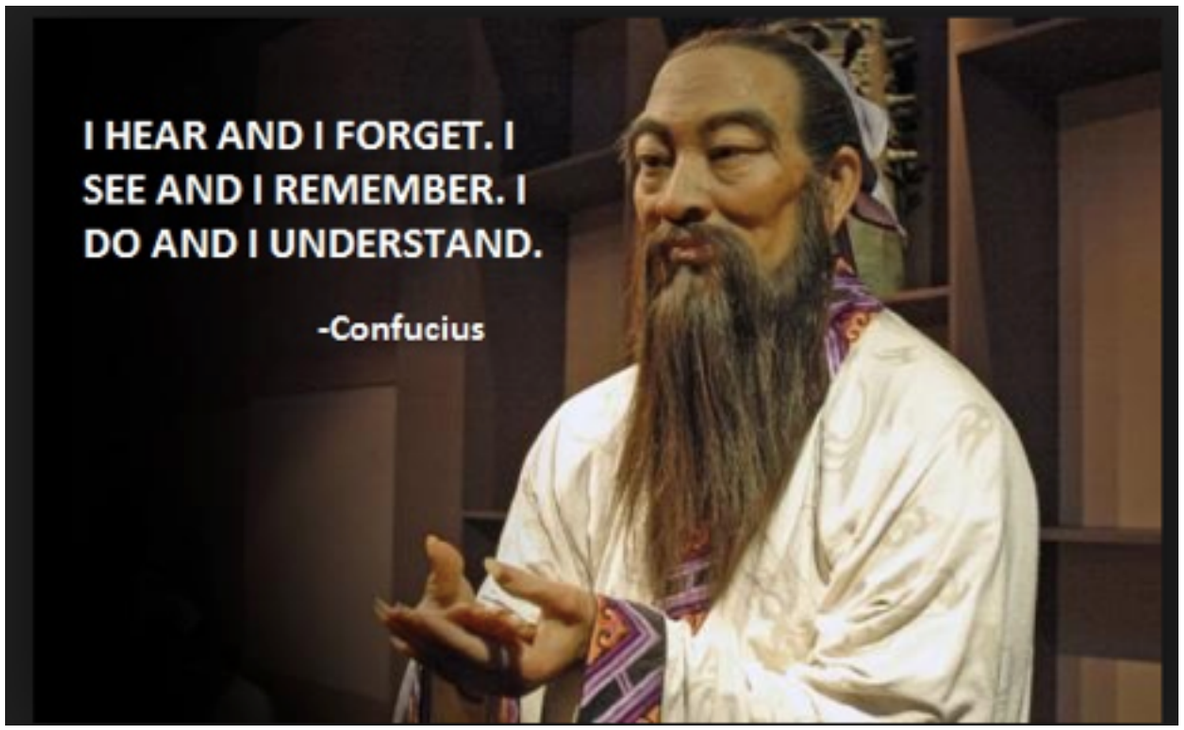
You do some stuff.
Most fails.
Some works.
You do more of what works.
If it works big, others quickly copy it.
Then you do something else.

The trick is the doing something else."

Confucius says

When it is obvious that the goals cannot be reached,
don't adjust the goals,
adjust the action steps.

Confucius (551-479 BCE)



The highest priority for human survival is:

- Water
- Air
- Food

Critical Body Priorities

Dynamic prioritization, the human body method, is a pretty **smart** prioritization method, and keeps you **alive** in **changing** conditions.

We could do worse than to use this **dynamic and logical** method
for management planning.

?

Value Decision Tables

		
Product Value 1		
Product Value 2		
Resources		

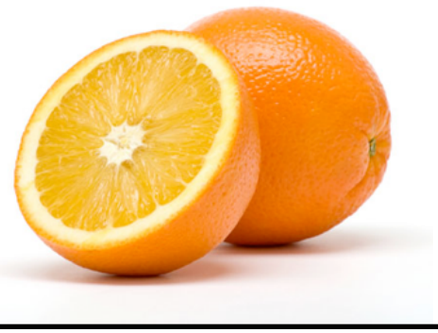
Value Decision Tables

		
Product Value 1		
Product Value 2		
Resources		

Value Decision Tables

			
Product Value 1			
Product Value 2			
Resources			

Value Decision Tables

			
Taste			
Resources			

Value Decision Tables

			
Taste			
Nutrition			
Resources			

Value Decision Tables

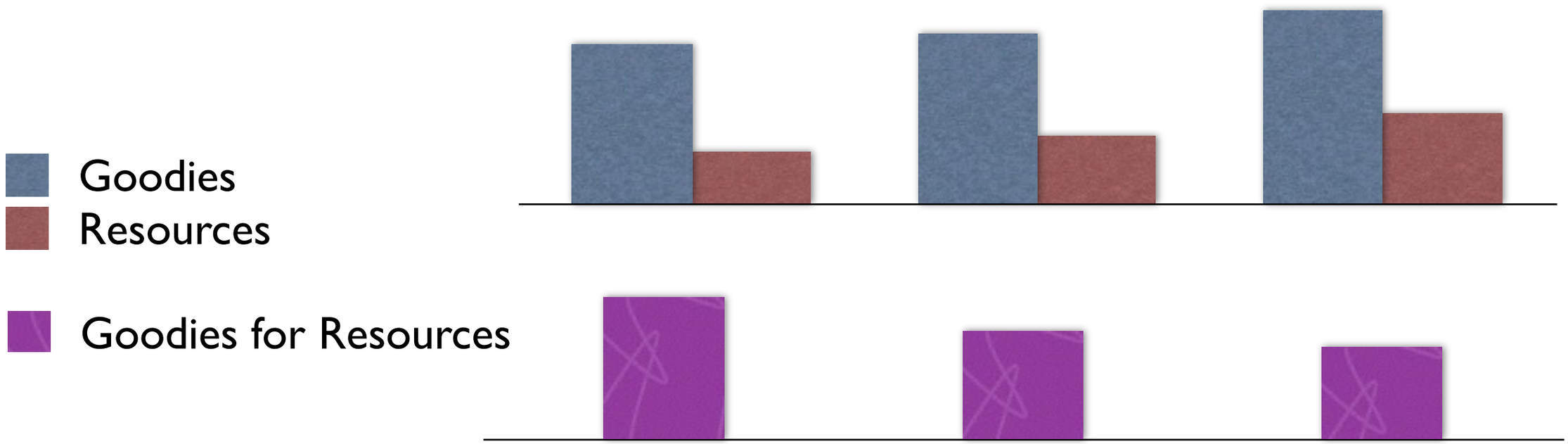
			
Taste			
Nutrition			
Shelf Life			
Resources			

Value Decision Tables

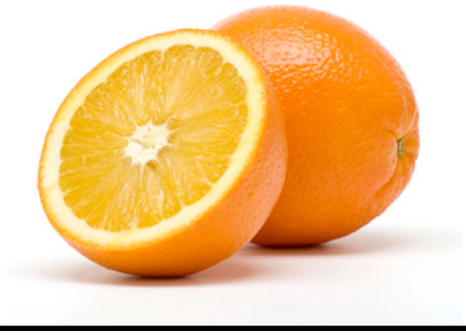
			
Taste			
Nutrition			
Shelf Life			
Sum Goodies			
Resources			

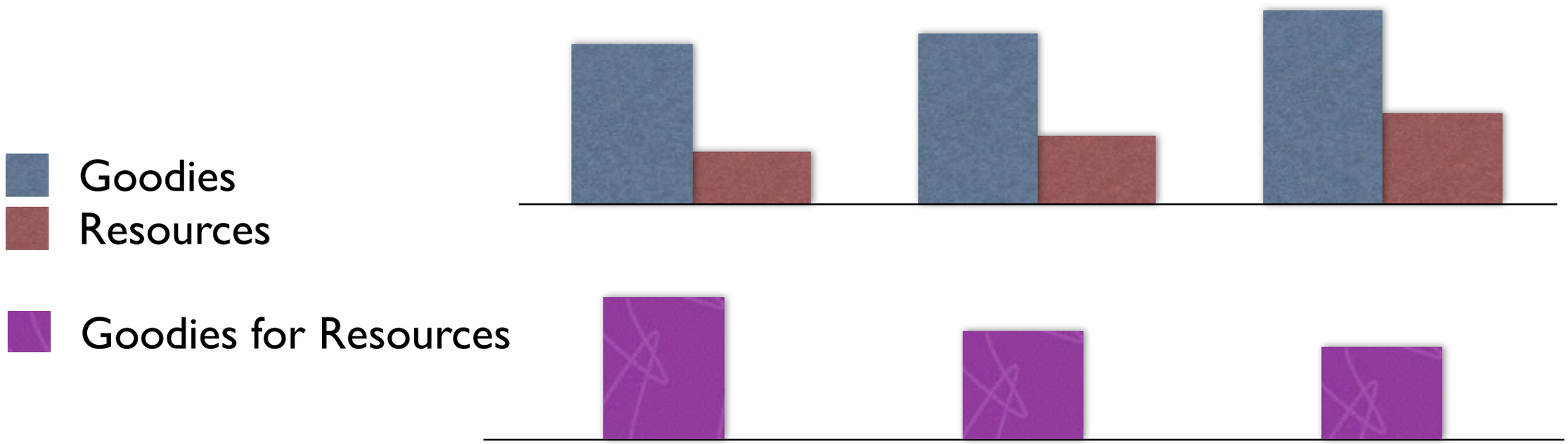
Value Decision Tables

			
Taste	0,2	0,5	0,9
Nutrition	0,3	0,7	0,9
Shelf Life	0,8	0,3	-0,1
Sum Goodies	1,3	1,5	1,7
Resources	0,4	0,6	0,8



Value Decision Tables

			
Taste	0,2	0,5	0,9
Nutrition	0,3	0,7	0,9
Shelf Life	0,8	0,3	-0,1
Sum Goodies	1,3	1,5	1,7
Resources	0,4	0,6	0,8



Confirmit: Results

Description of requirement/work task	Past	Status
Usability.Productivity: Time for the system to generate a survey	7200 sec	15 sec

Confirmit: Results

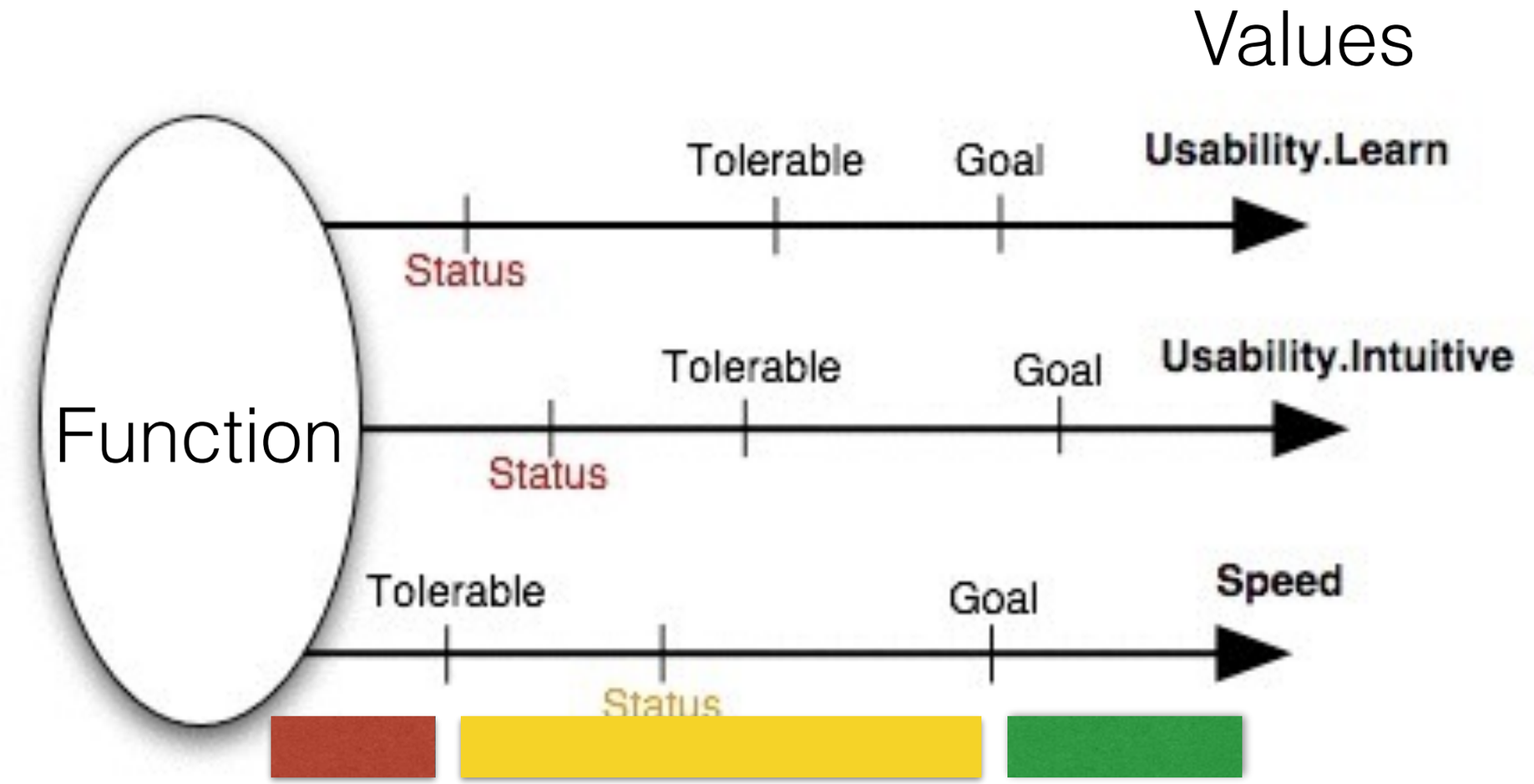
Description of requirement/work task	Past	Status
Usability.Productivity: Time for the system to generate a survey	7200 sec	15 sec
Usability.Productivity: Time to set up a typical specified Market Research-report (MR)	65 min	20 min
Usability.Productivity: Time to grant a set of End-users access to a Report 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

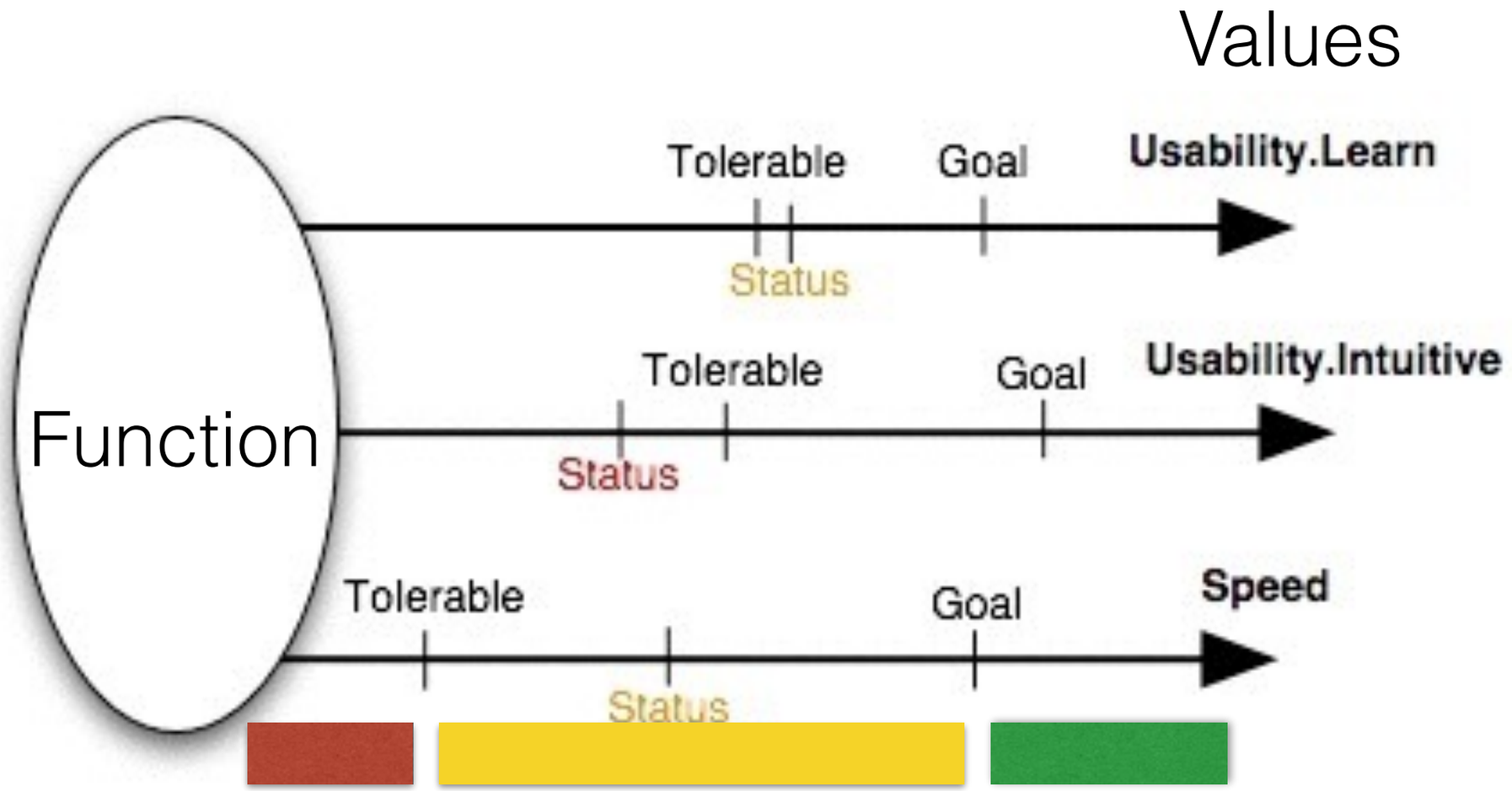
Confirmit
Snapshot End Week 9 of 12

Current Status	Improvements		Goals			Step9 Recoding			
						Estimated impact		Actual impact	
Units	Units	%	Past	Tolerable	Goal	Units	%	Units	%
			Usability.Replacability (feature count)						
1,00	1,0	50,0	2	1	0				
			Usability.Speed.NewFeaturesImpact (%)						
5,00	5,0	100,0	0	15	5				
10,00	10,0	200,0	0	15	5				
0,00	0,0	0,0	0	30	10				
			Usability.Intuitiveness (%)						
0,00	0,0	0,0	0	60	80				
			Usability.Productivity (minutes)						
20,00	45,0	112,5	65	35	25	20,00	50,00	38,00	95,00
			Development resources						
	101,0	91,8	0		110	4,00	3,64	4,00	3,64

Confirmit
Snapshot End Week 9 of 12

Current Status	Improvements		Goals			Step9 Recoding Estimated impact		Actual impact	
Units	Units	%	Past	Tolerable	Goal	Units	%	Units	%
			Usability.Replacability (feature count)						
1,00	1,0	50,0	2	1	0				
			Usability.Speed.NewFeaturesImpact (%)						
5,00	5,0	100,0	0	15	5				
10,00	10,0	200,0	0	15	5				
0,00	0,0	0,0	0	30	10				
			Usability.Intuitiveness (%)						
0,00	0,0	0,0	0	60	80				
			Usability.Productivity (minutes)						
20,00	45,0	112,5	65	35	25	20,00	50,00	38,00	95,00
			Development resources						
	101,0	91,8	0		110	4,00	3,64	4,00	3,64





Confirmit
Snapshot End Week 9 of 12

Current Status	Improvements		Goals			Step9 Recoding Estimated impact		Actual impact	
Units	Units	%	Past	Tolerable	Goal	Units	%	Units	%
			Usability.Replacability (feature count)						
1,00	1,0	50,0	2	1	0				
			Usability.Speed.NewFeaturesImpact (%)						
5,00	5,0	100,0	0	15	5				
10,00	10,0	200,0	0	15	5				
0,00	0,0	0,0	0	30	10				
			Usability.Intuitiveness (%)						
0,00	0,0	0,0	0	60	80				
			Usability.Productivity (minutes)						
20,00	45,0	112,5	65	35	25	20,00	50,00	38,00	95,00
			Development resources						
	101,0	91,8	0		110	4,00	3,64	4,00	3,64

Confirmit
Snapshot End Week 9 of 12

Dynamic Priority Metric

Weekly Progress

Constraint

Target

Current Status	Improvements		Goals			Step9 Recoding			
						Estimated impact		Actual impact	
Units	Units	%	Past	Tolerable	Goal	Units	%	Units	%
			Usability.Replacability (feature count)						
1,00	1,0	50,0	2	1	0				
			Usability.Speed.NewFeaturesImpact (%)						
5,00	5,0	100,0	0	15	5				
10,00	10,0	200,0	0	15	5				
0,00	0,0	0,0	0	30	10				
			Usability.Intuitiveness (%)						
0,00	0,0	0,0	0	60	80				
			Usability.Productivity (minutes)						
20,00	45,0	112,5	65	35	25	20,00	50,00	38,00	95,00
			Development resources						
	101,0	91,8	0		110	4,00	3,64	4,00	3,64

Estimates

Weekly Testing

Confirmit

4 product areas were attacked in all: **25 Qualities** concurrently

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		?	?
			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

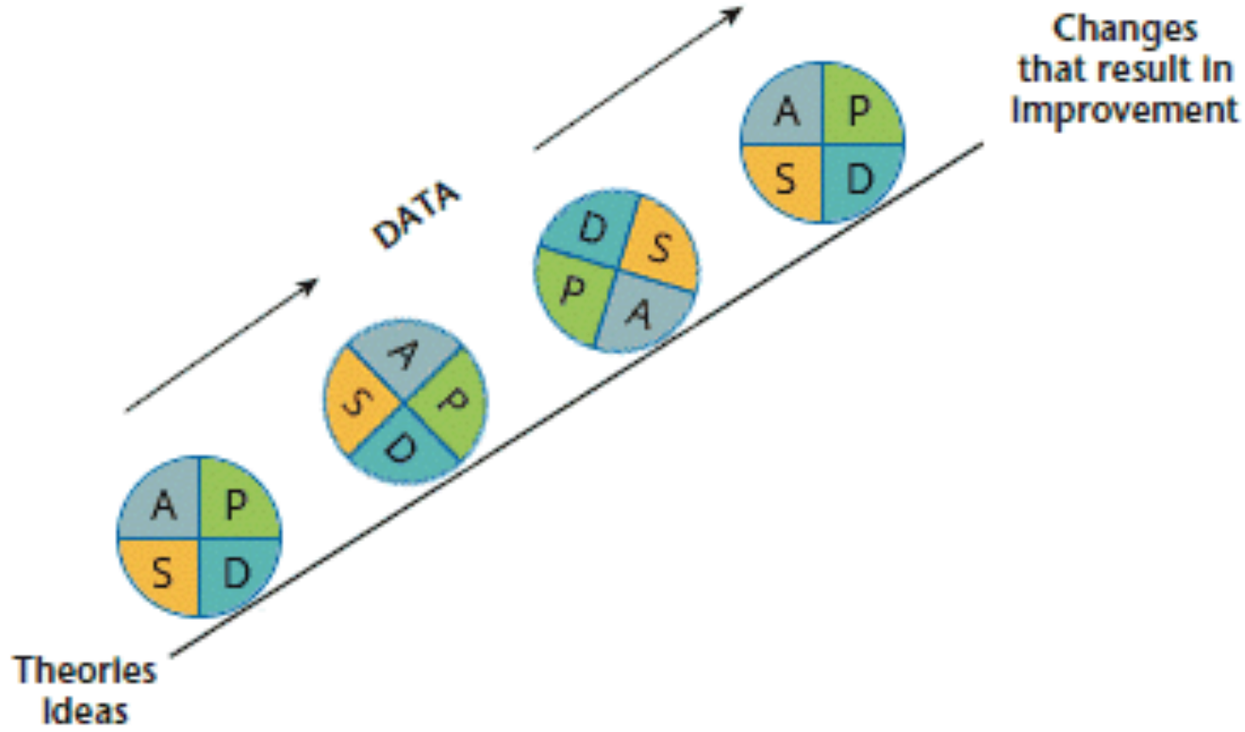
Current Status	Improvements		XML Web Services		
Units	Units	%	Past	Tolerable	Goal
			TransferDefinition.Usability.Efficiency		
7,0	9,0	81,8	16	10	5
17,0	8,0	53,3	25	15	10
			TransferDefinition.Usability.Response		
943,0	-186,0	#######	170	60	30
			TransferDefinition.Usability.Intuitiveness		
5,0	10,0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

#NoEstimates

- "Estimation: A Paradigm Shift Toward Dynamic Design-to Cost and Radical Management"
  - Volume 13 Issue 2 of SQP journal - the March 2011 version.
    - Software Quality Professional, USA
    - The American Society for Quality (ASQ)
  - http://www.gilb.com/tiki-download_file.php?fileId=460
  - Slides: For BCS SPA, London
    - http://www.gilb.com/tiki-download_file.php?fileId=470

The basic process: DDtCV

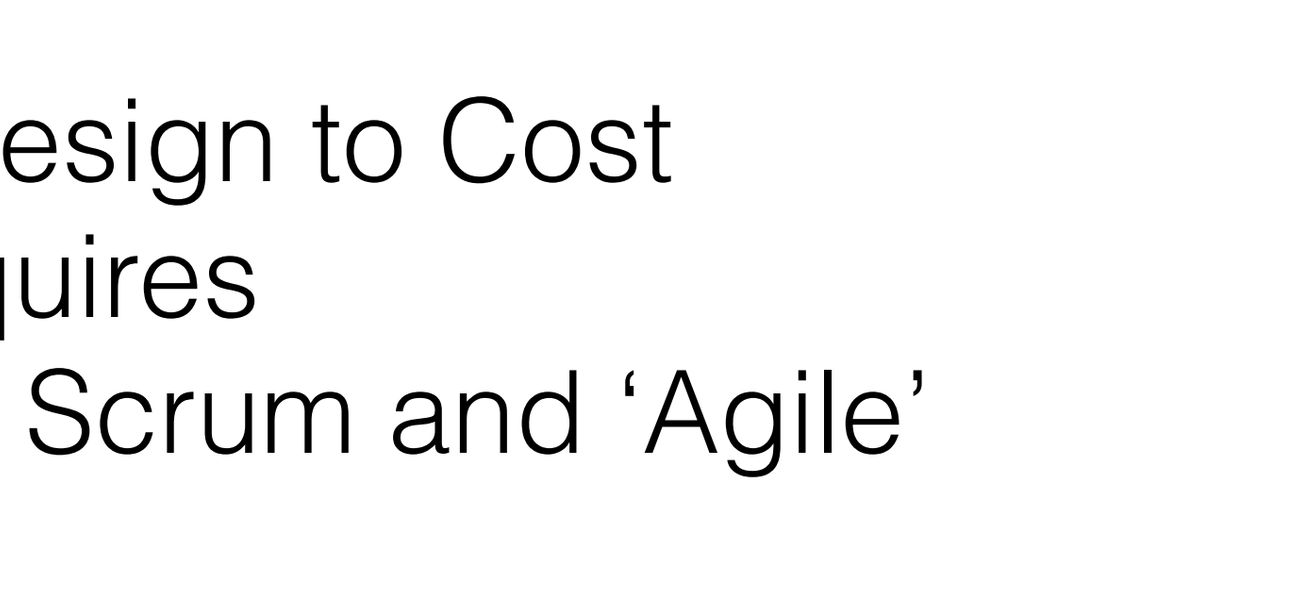
- **If all is 'on track'**
  - x% value**s**, for
  - X% cost**s**
- **Do a new value delivery cycle**
- **If not on track, then '*change something*'; to get back on track**



PDSA: Plan Do Study Act
Deming Cycle

Dynamic Design to Cost
requires
things absent in Scrum and ‘Agile’

- Multiple resource constraints
  - deadline, money, people, space
- Multiple measurable values
  - qualities, savings
- Cycle Decomposition by Value
- Measurement of Value each cycle
- Design to cost



Attributes of Dynamic Design to Cost (DDC)

- **Ability to deliver on time**
- **Ability to deliver to budget**
- **Ability to delivery to multiple ambitious quality targets**
- **Ability to learn what works early**
- **Ability to experiment with high promise architecture, at low risk**
- **Ability to experiment, low risk, with development processes**
- **Fits a no cure no-pay contracting model**
  - **flexiblecontracts-com**

Dynamic Design to Cost as a defence against arbitrary budgets and deadlines.

in 4.5 VP

‘Dynamic design to cost’ as a **management process**, is particularly interesting to understand,
when you do not have the luxury *to estimate how much you need or want, for your own scheduling and funding purposes.*
You are not *asked*, you are ***told*** **the costs and deadlines.**

The government client, or other powerful forces, set a deadline for you; and they allocated a fixed-cost budget.
Your salespeople ‘happily’ won, as low bidder of a fixed-price contract.
You, however, are then stuck with **the problem of ‘making it happen’, on time, under budget.**

Principle 6.2

DYNAMIC PRIORITY

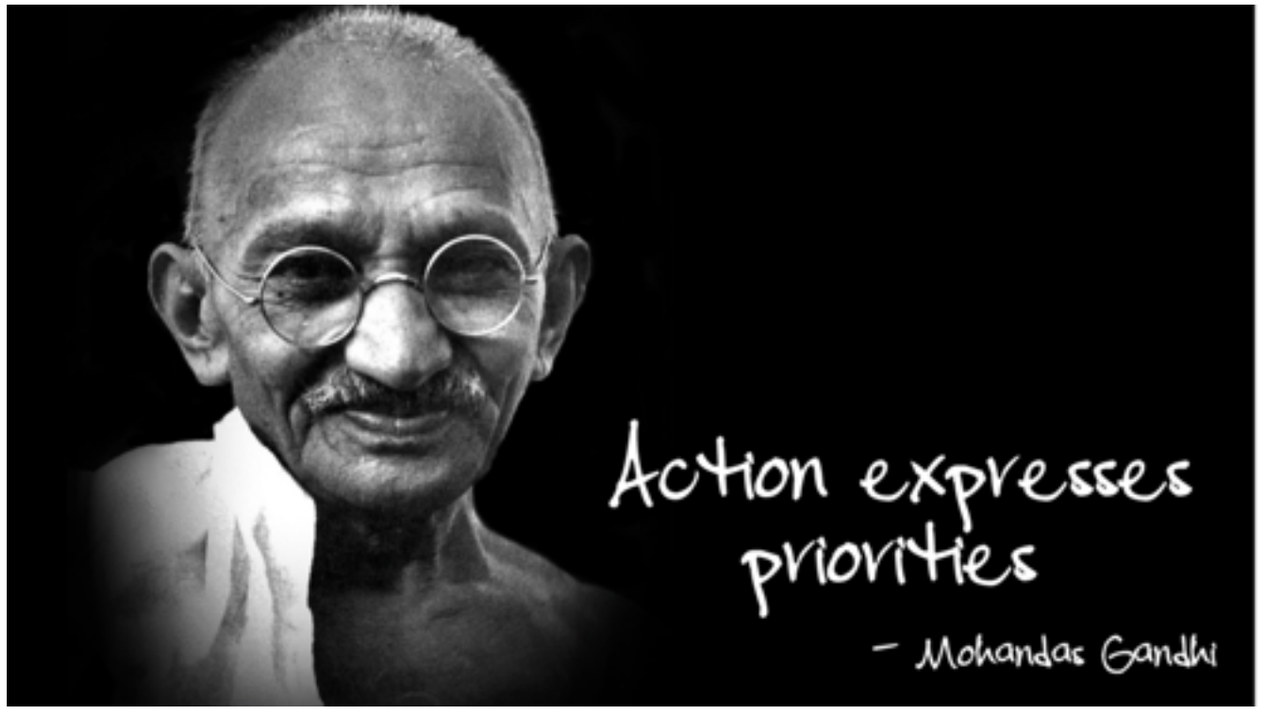
(VP book):

Static initial prioritization is unrealistic –

things change

Why
Priority
must be
Dynamic

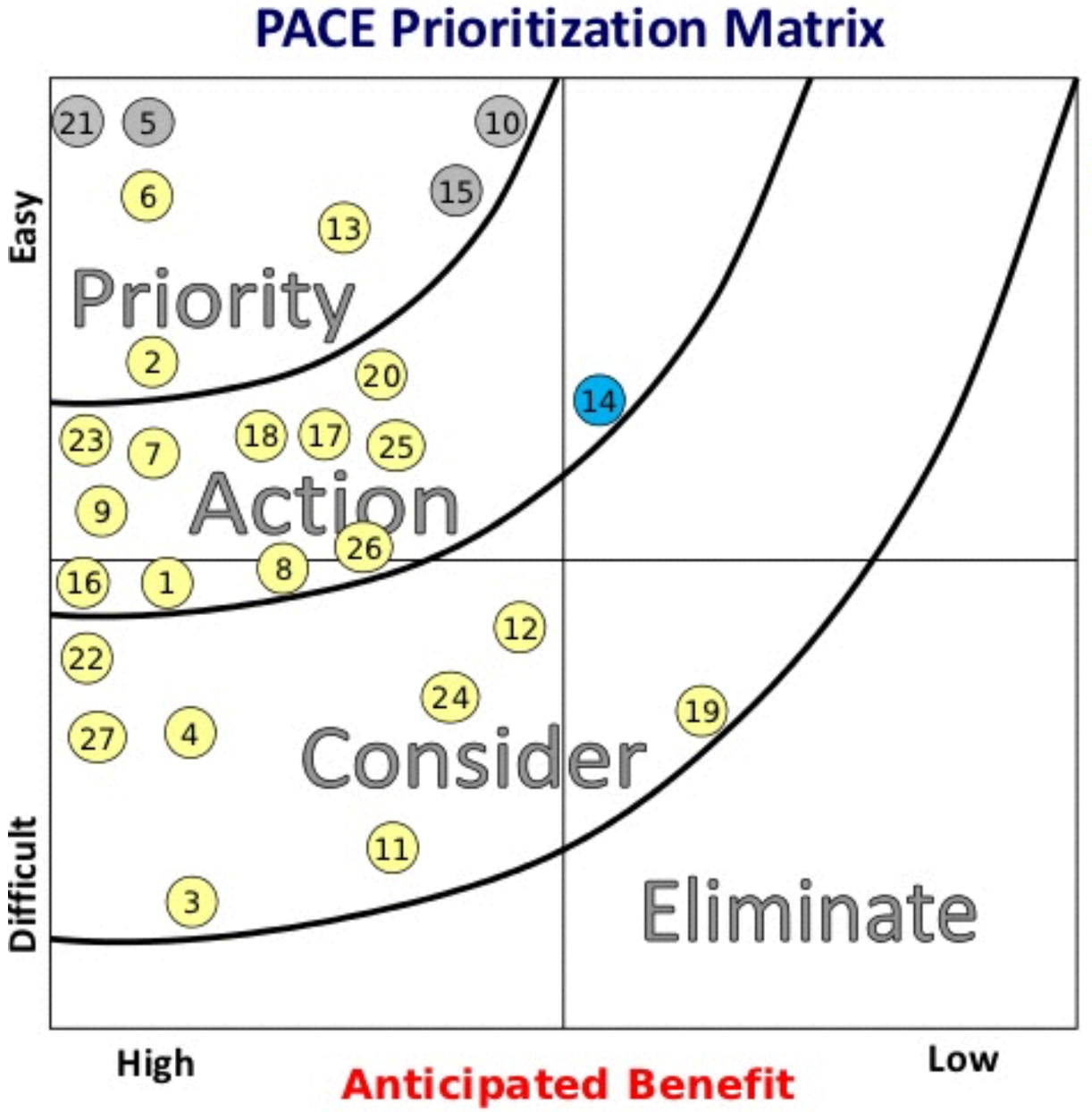
- The **facts needed** to determine your current priority,
  - o are **constantly and arbitrarily changing**
- The facts needed are:
  - o ***remaining limited resources, and remaining distance to Goals***
- Only when these facts are available, can you search for a 'suitable strategy':
  - o *one that will move you towards your Goals as much as possible,*
  - o *within the (weekly) cycle duration,*
  - o *with as little use of other resources, like money, as possible.*
- We can **prioritize** any strategy, which we can find,
  - o that gives best **progress**, towards *residual* **Goal** levels,
  - o at the lowest consumption of residual resources.



Conditions for Logical Prioritization

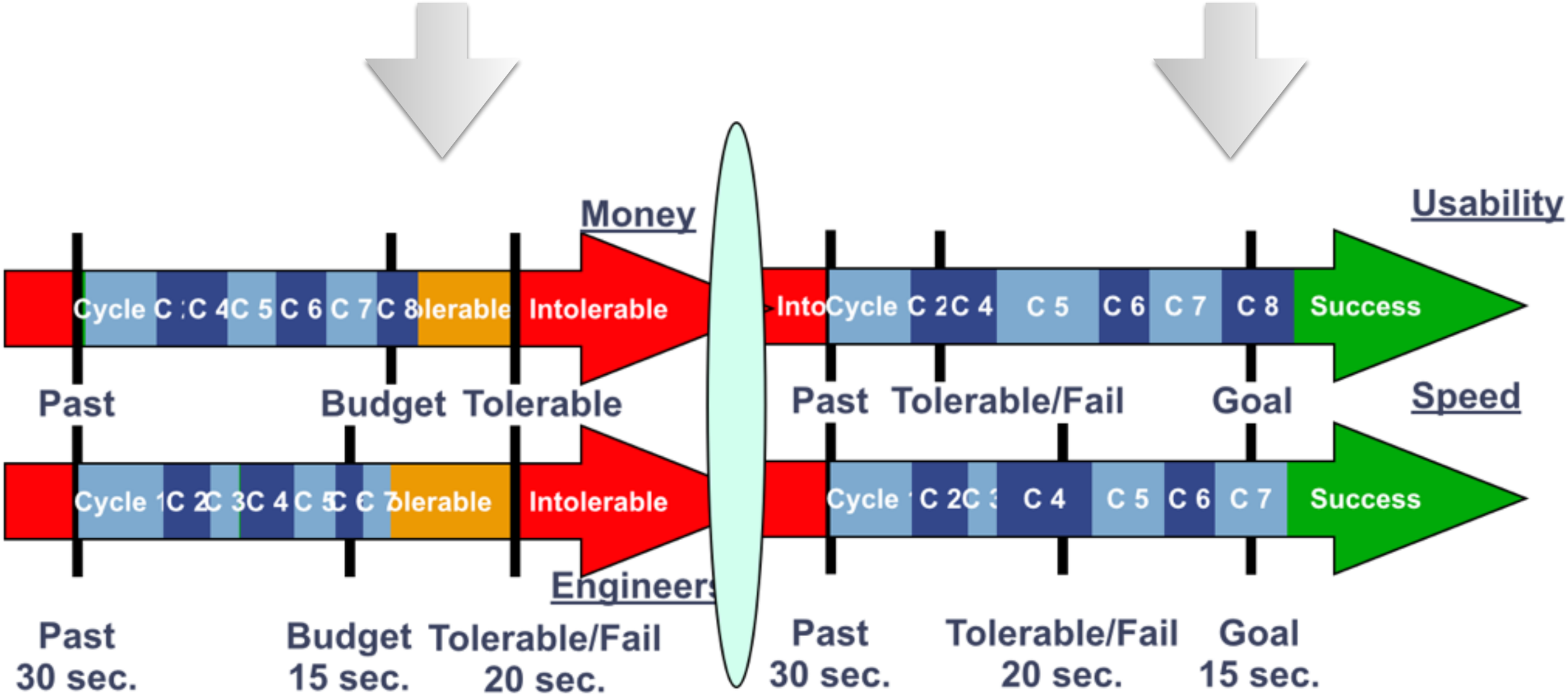
VP 6.8

1. **Critical Objectives identified**
2. **Objectives Quantified**
3. **Constraints ID & Quantified**
4. **Clear detailed strategies**
5. **Estimates of Strategy Impacts & Costs**
6. **Risks and Uncertainties ID**
7. **Policy for deciding what to prioritize (Value / € ?), Risk**

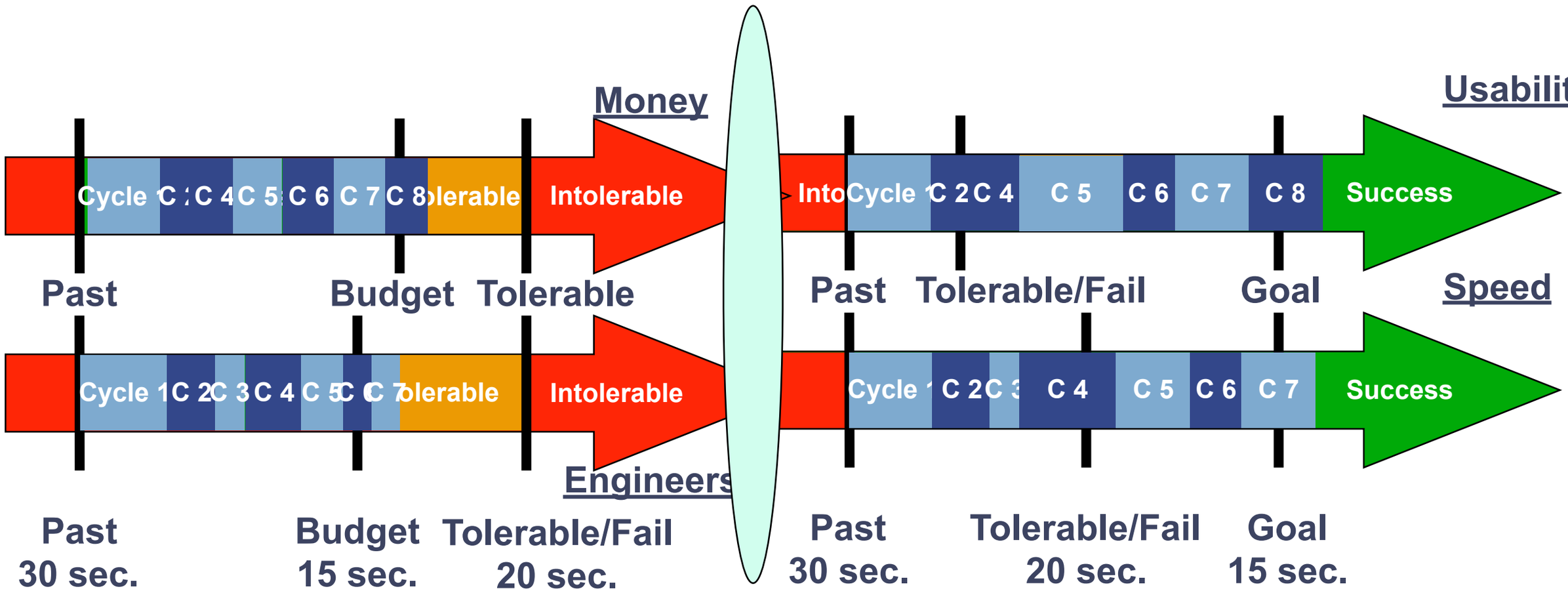


http://www.slideshare.net/KarenMartinGroup/08-232012-value-stream-mapping

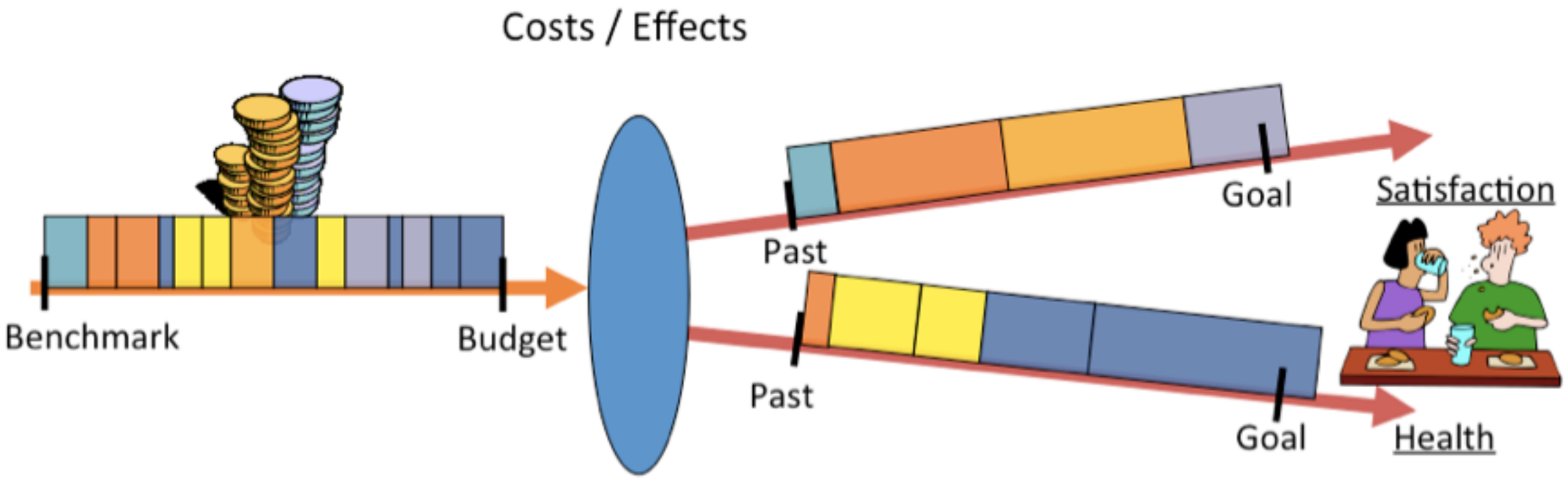
Multiple Constraints and Multiple Objectives (Static)

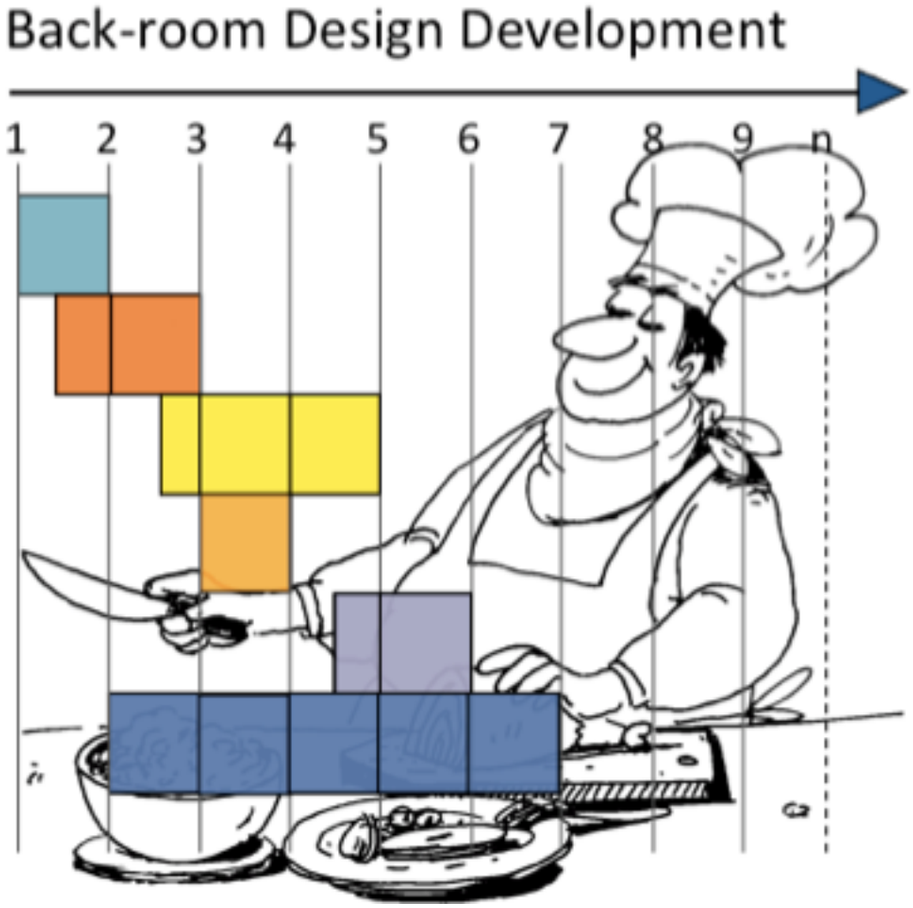


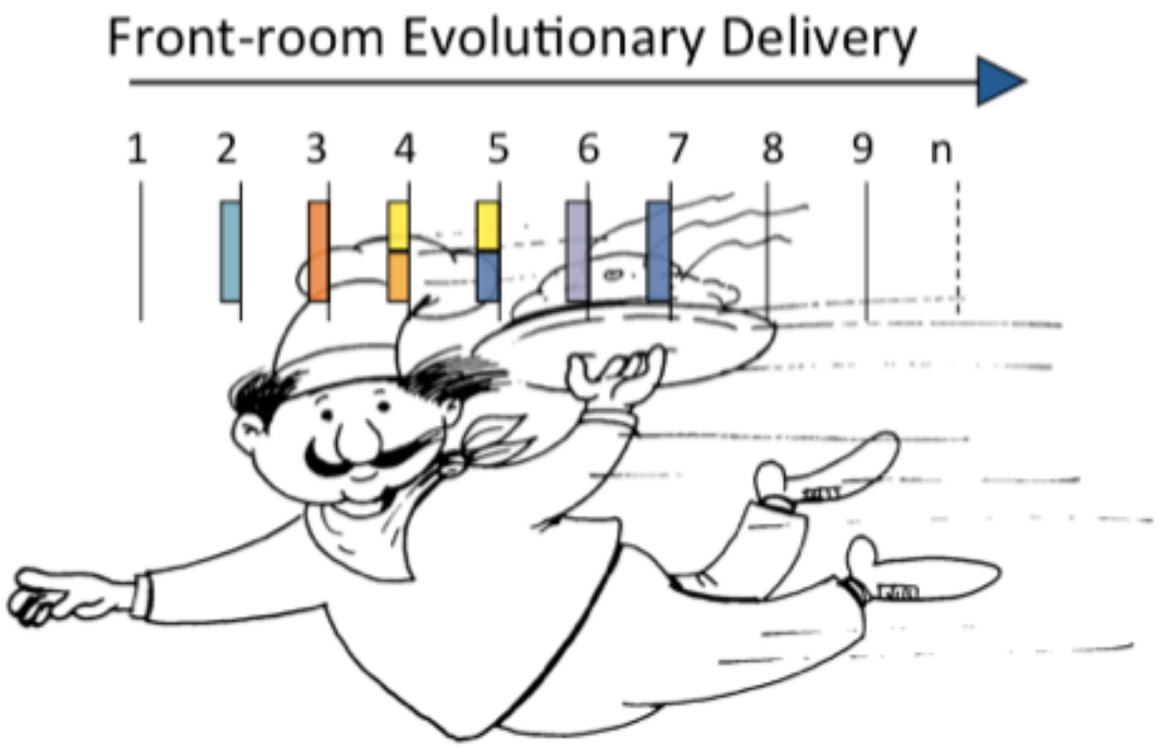
Each Evolutionary Cycle uses a constrained budget of Development Resources



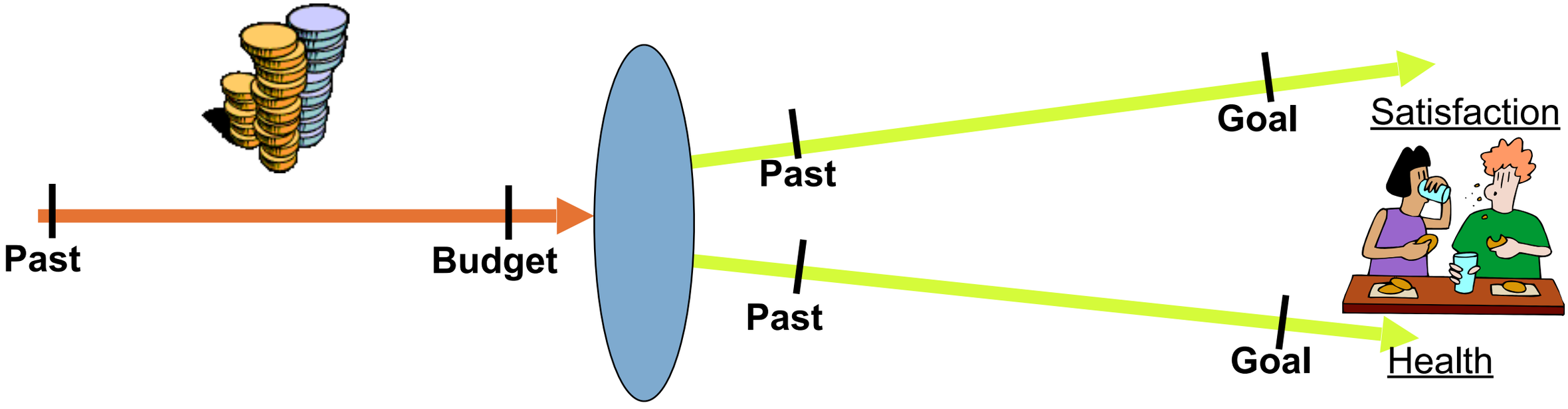
Dynamic ‘Restaurant’ Prioritization (Static)





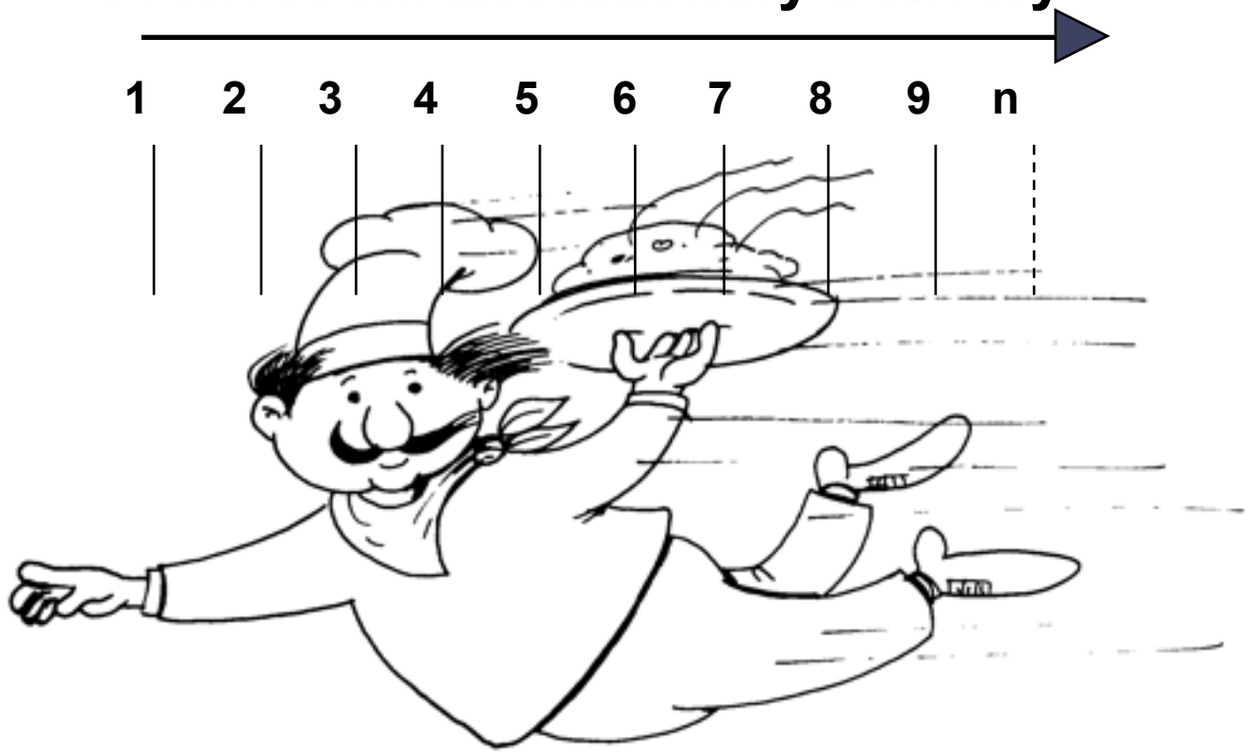


Costs / Effects

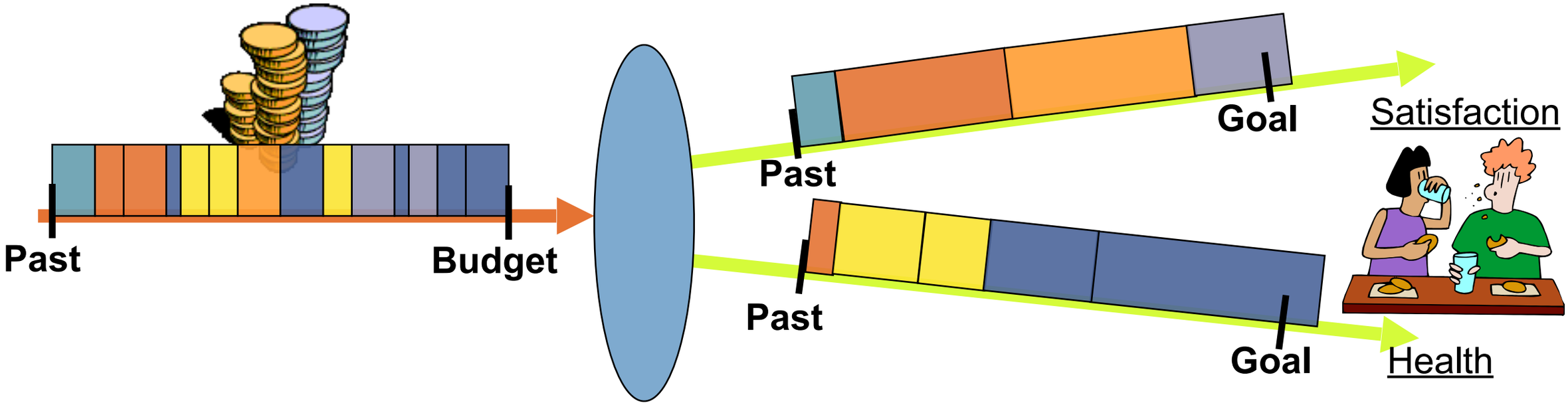


Back-room Design Development

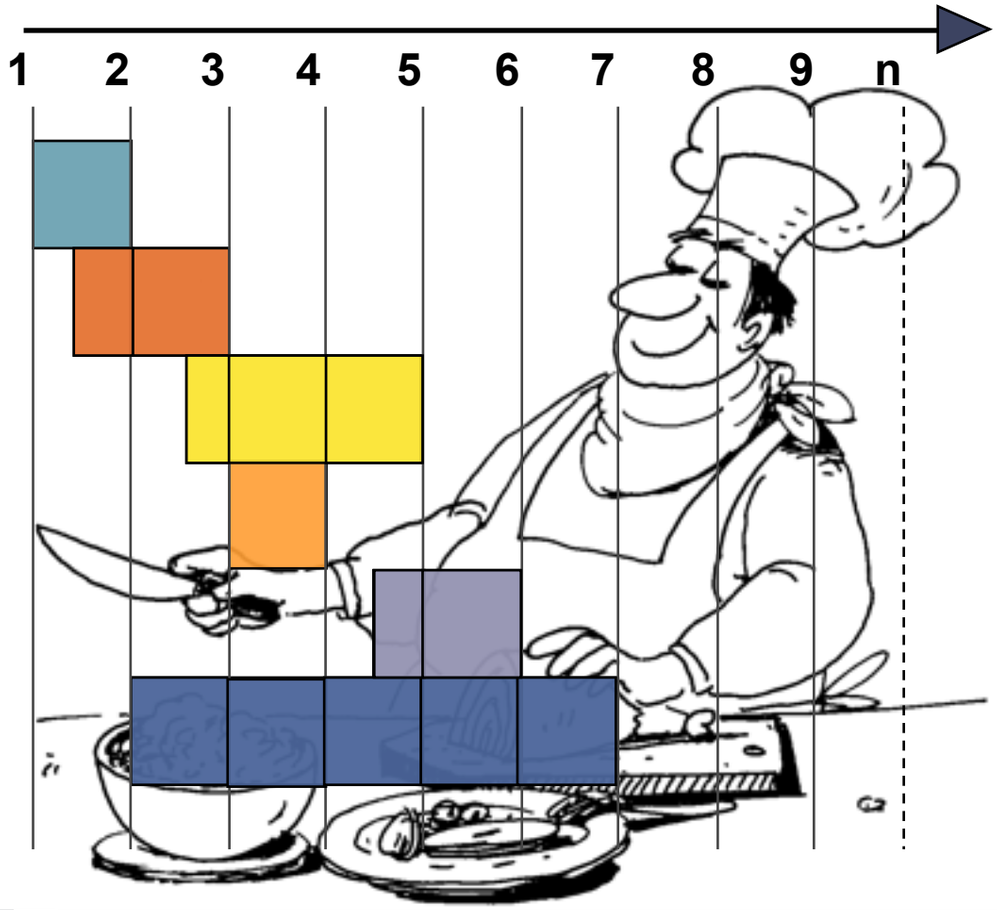
Front-room Evolutionary Delivery



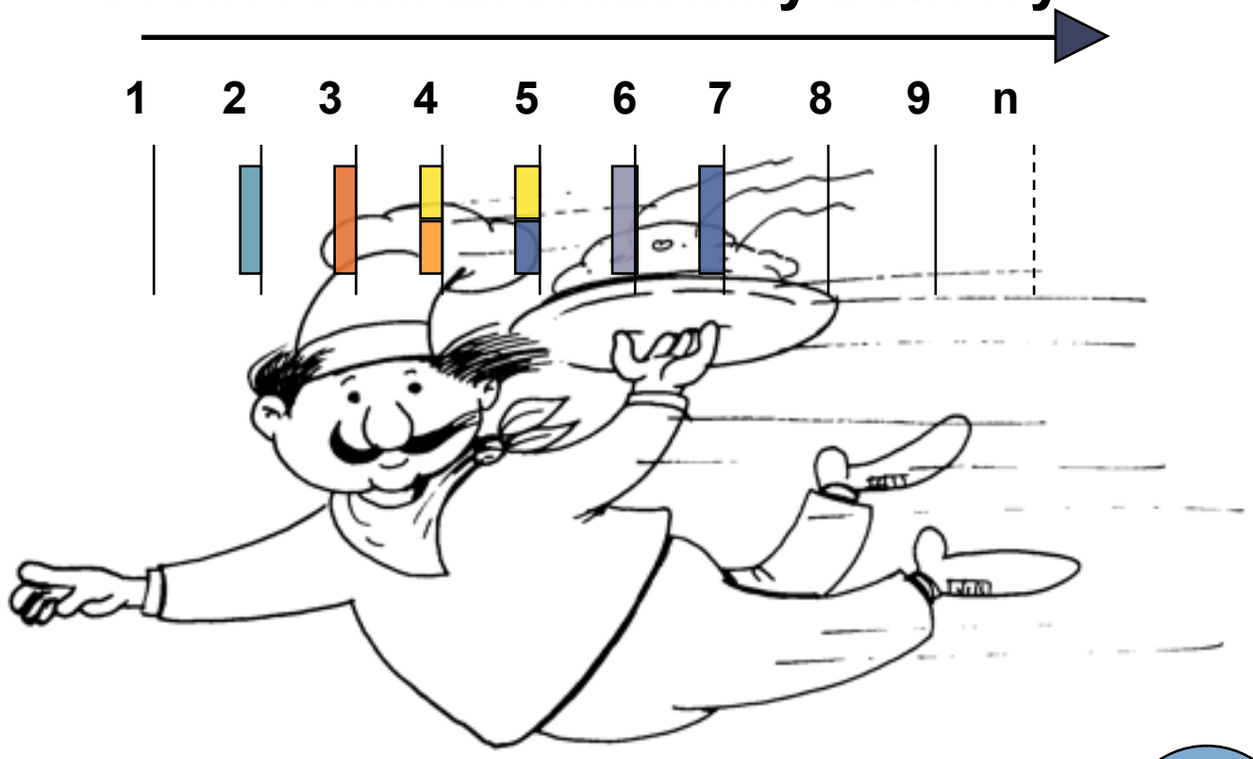
Costs / Effects



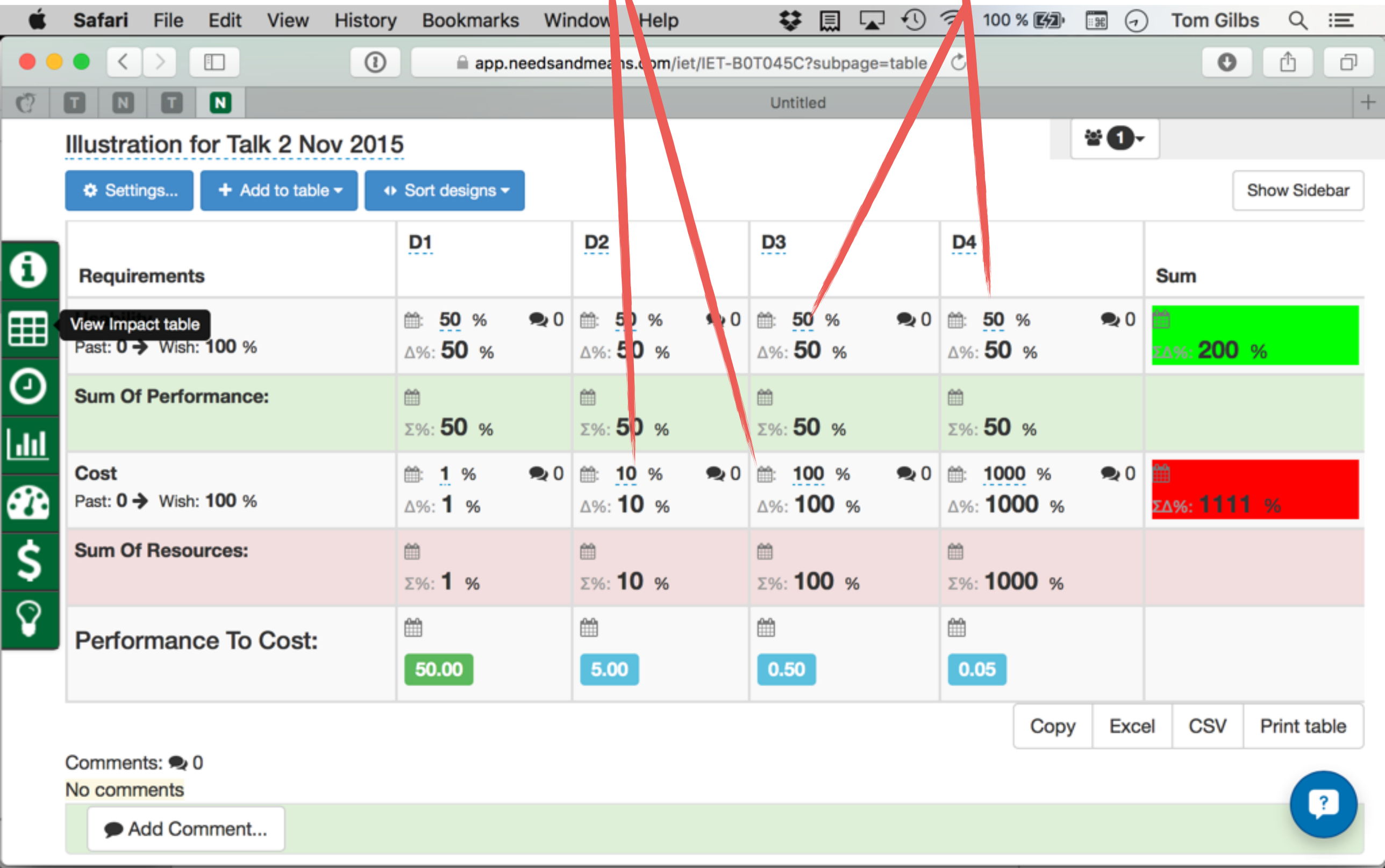
Back-room Design Development



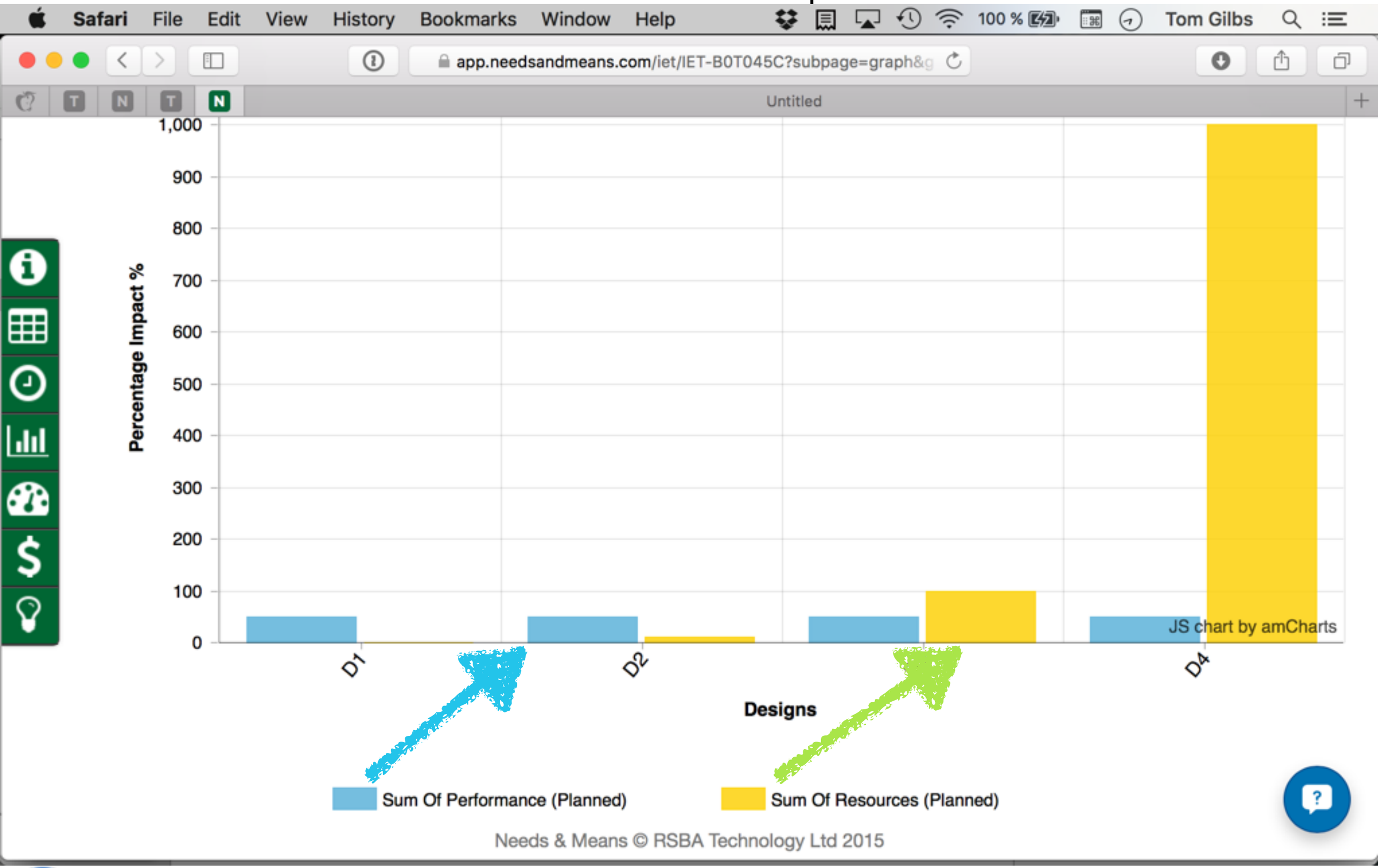
Front-room Evolutionary Delivery



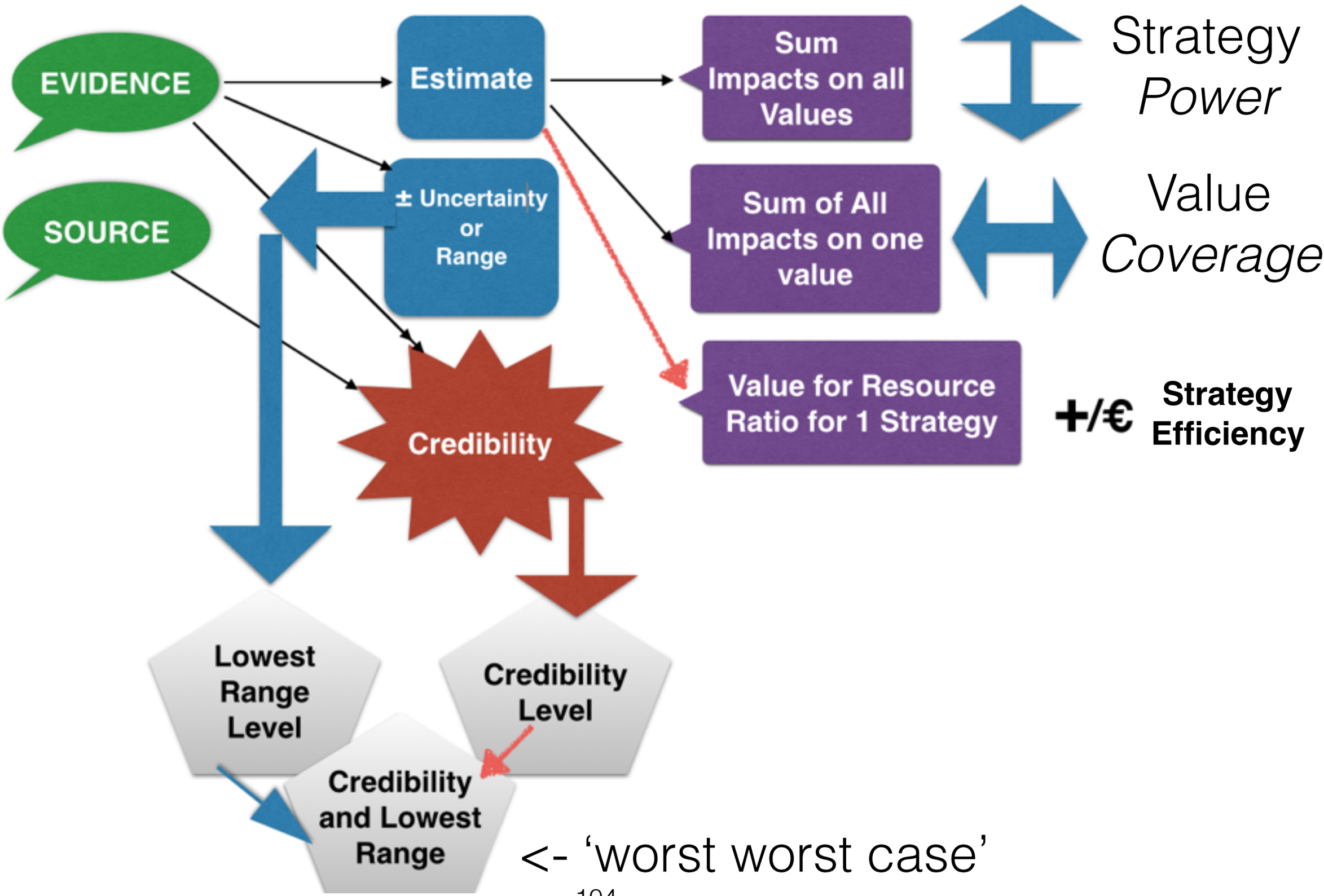
Impact Table with highly varied costs, for ‘same impact’ on requirements



Bar Chart from the Impact Table



Dynamic Prioritizing with Risks using IE Table



<- ‘worst worst case’

Impact Table with Risks

Safari File Edit View History Bookmarks Window Help 100 % Tom Gilbs

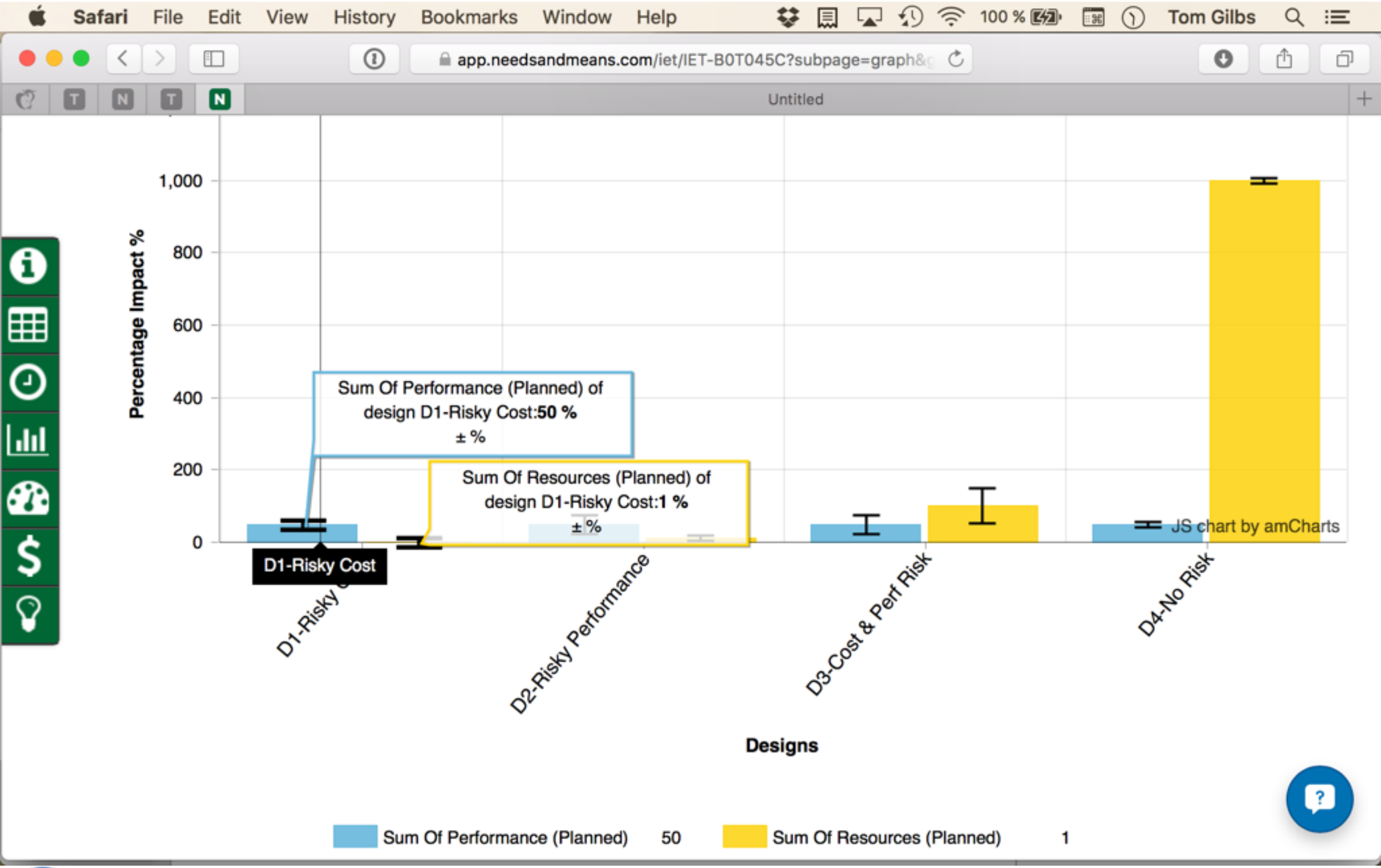
app.needsandmeans.com/iet/IET-B0T045C

Untitled

Settings... | + Add to table | Sort designs | Show Sidebar

Requirements	D1-Risky Cost	D2-Risky Performance	D3-Cost & Pe...	D4-No Risk	Sum
Usability Past: **0** → Wish: **100 %**	50 ± 0 % 0 Δ%: **50 ± 0 %** ?%: *50 % (x 1.0)*	50 ± 49 % 0 Δ%: **50 ± 49 %** ?%: *5 % (x 0.1)*	50 ± 49 % 0 Δ%: **50 ± 49 %** ?%: *5 % (x 0.1)*	50 ± 0 % 0 Δ%: **50 ± 0 %** ?%: *50 % (x 1.0)*	ΣΔ%: **200 ± 98 %**
Sum Of Performance: Credibility - adjusted:	Σ%: **50 ± 0 %** Σ?%: *50 %*	Σ%: **50 ± 49 %** Σ?%: *5 %*	Σ%: **50 ± 49 %** Σ?%: *5 %*	Σ%: **50 ± 0 %** Σ?%: *50 %*	
Cost Past: **0** → Wish: **100 %**	1 ± 0.9 % 0 Δ%: **1 ± 1 %** ?%: *2 % (x 0.1)*	10 ± 0 % 0 Δ%: **10 ± 0 %** ?%: *20 % (x 0.0)*	100 ± 99 % 0 Δ%: **100 ± 99 %** ?%: *190 % (x 0.1)*	1000 ± 0 % 0 Δ%: **1000 ± 0 %** ?%: *1000 % (x 1.0)*	ΣΔ%: **1111 ± 100 %**
Sum Of Resources: Credibility - adjusted:	Σ%: **1 ± 1 %** Σ?%: *2 %*	Σ%: **10 ± 0 %** Σ?%: *20 %*	Σ%: **100 ± 99 %** Σ?%: *190 %*	Σ%: **1000 ± 0 %** Σ?%: *1000 %*	
Performance To Cost:	50.00	5.00	0.50	0.05	
Ratio (Worst Case)	25.00	0.10	0.01	0.05	
Ratio (Cred. - adjusted)	26.32	0.25	0.03	0.05	

Bar Graph of the Impact Table with Risks



The 2 Estimation Elements in 'Design to Cost'.

VP 4.5

1. You estimate, and then re-estimate, repeatedly, based on 'costs to date'.

You *extrapolate* and say something like 'if we continue with these strategies, then we will run over budget, and past the deadline.

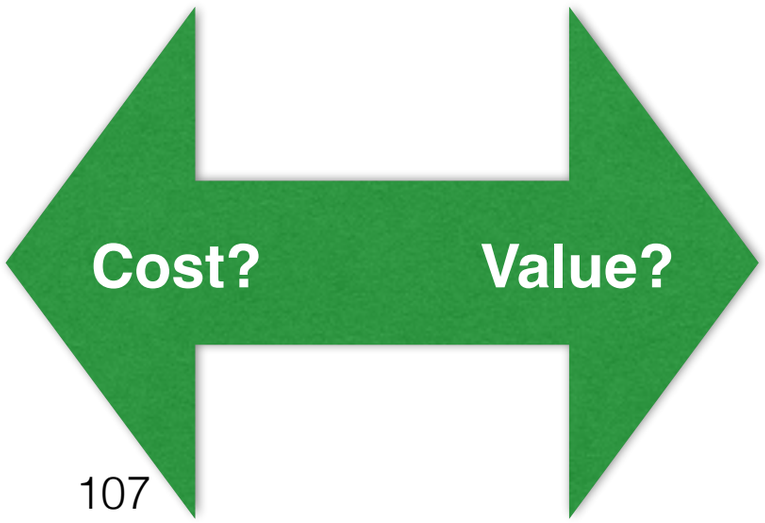
So, we must change strategies, and we must do it **now**.'

2. In addition to the cost and value extrapolation,

based on incremented facts, and on hard credible evidence,

we use a *second* sort of estimation:

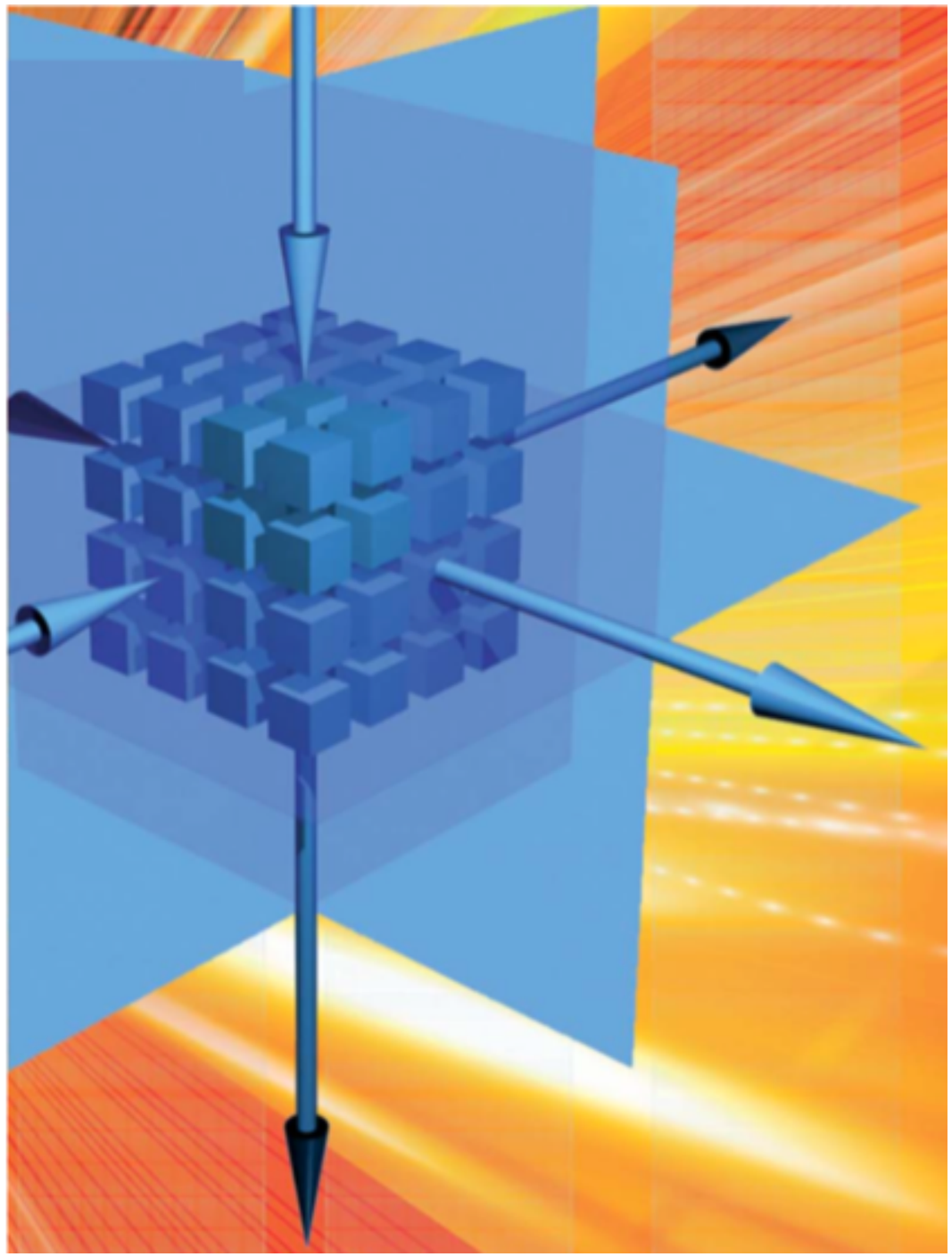
'what **will** candidate strategy X cost,
in time and/or money?



Decomposition

Separating out
small stakeholder-delivery
value increments

from your top-level
Architecture/Strategies



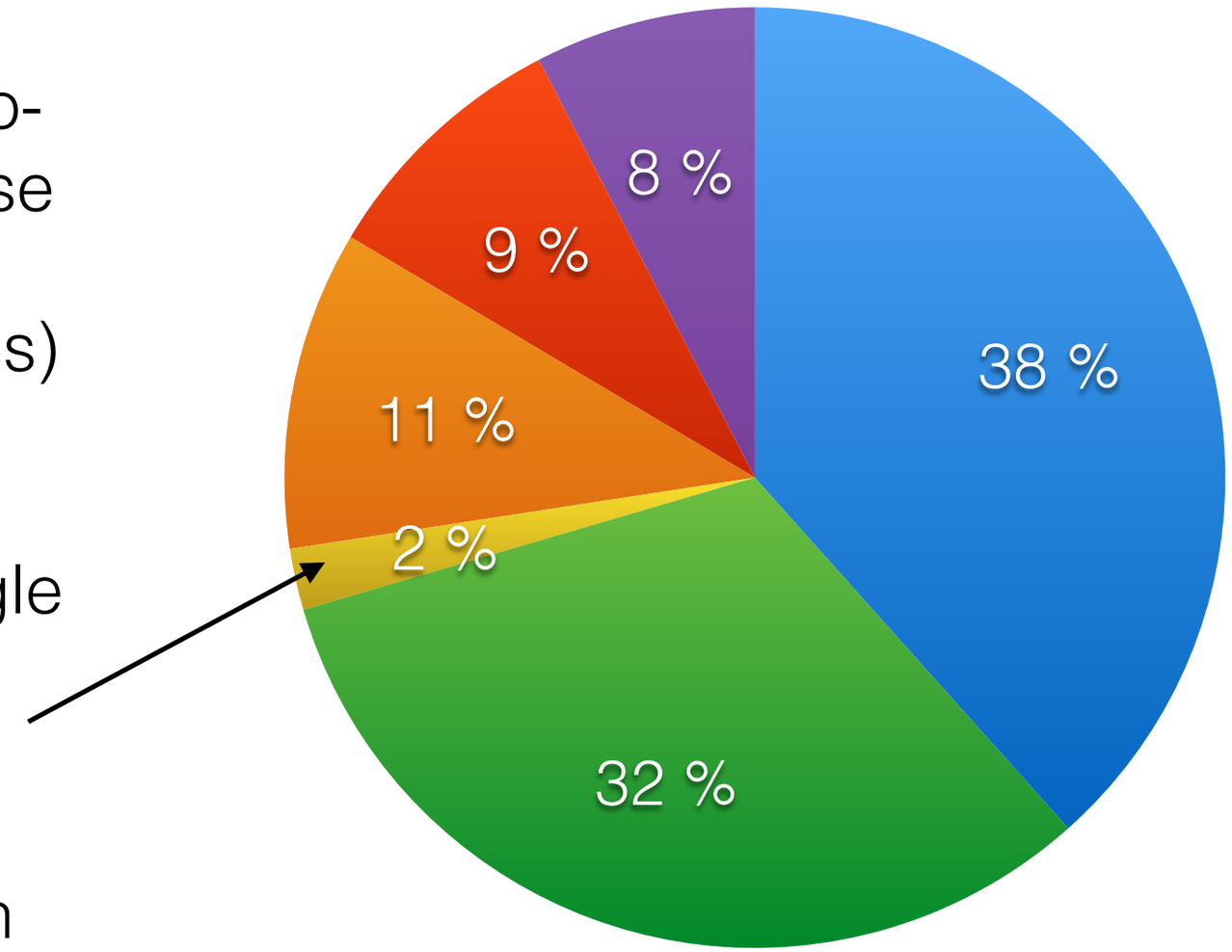
Ideal Separation of a Value-Delivery Step

1. No dependencies, that are not already existing in the to-be-incremented system base

2. Will give measurable value(s) to some stakeholder (s)

3. Can be completed in a single value delivery cycle (2% of time to deadline, a week)

4. Acceptable risk of deviation (±30% ?) from estimated values and costs



Methods for Extraction

1. Just ask: 'what could we do next week to deliver some value' ?

2. Use an Impact Estimation Table to decompose and see high value opportunities

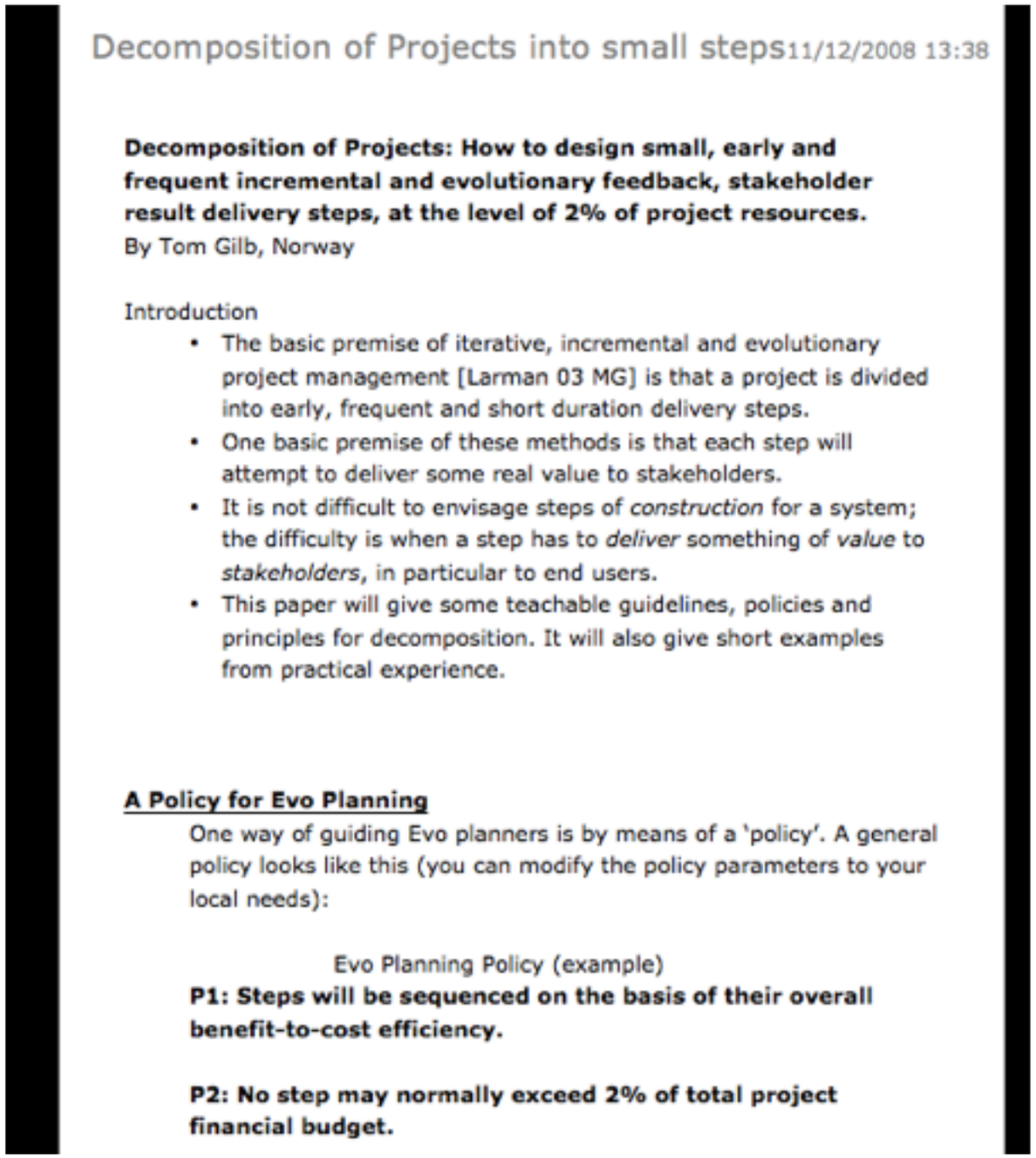
3. Use 20 Principles of Decomposition (CE Ch 10, VP)

US Army Example: PERSINSCOM: Personnel System

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

29.5 to 1

Decomposition Principles
A Teachable Discipline



Decomposition of Projects into small steps11/12/2008 13:38

Decomposition of Projects: How to design small, early and frequent incremental and evolutionary feedback, stakeholder result delivery steps, at the level of 2% of project resources.

By Tom Gilb, Norway

Introduction

- The basic premise of iterative, incremental and evolutionary project management [Larman 03 MG] is that a project is divided into early, frequent and short duration delivery steps.
- One basic premise of these methods is that each step will attempt to deliver some real value to stakeholders.
- It is not difficult to envisage steps of *construction* for a system; the difficulty is when a step has to *deliver* something of *value* to *stakeholders*, in particular to end users.
- This paper will give some teachable guidelines, policies and principles for decomposition. It will also give short examples from practical experience.

A Policy for Evo Planning

One way of guiding Evo planners is by means of a 'policy'. A general policy looks like this (you can modify the policy parameters to your local needs):

Evo Planning Policy (example)

P1: Steps will be sequenced on the basis of their overall benefit-to-cost efficiency.

P2: No step may normally exceed 2% of total project financial budget.

How to decompose systems into small evolutionary steps:

some principles to apply:

1• *Believe* there is a way to do it, you just have not *found* it yet!

2• *Identify* obstacles, but don't use them as excuses: use your imagination to get *rid* of them!

3• Focus on *some usefulness* for the user or customer, however small.

4• Do not focus on the design ideas themselves, they are distracting, especially for small initial cycles. Sometimes you have to ignore them entirely in the short term!

5• Think; one customer, tomorrow, one interesting improvement.

6• Focus on the *results* (which you should have defined in your goals, moving toward target levels).

7• Don't be afraid to use temporary-scaffolding designs. Their cost must be seen in the light of the value of making some progress, and getting practical experience.

8• Don't be worried that your design is inelegant; it is results that count, not style.

9• Don't be afraid that the customer won't like it. *If* you are focusing on results *they want*, then by definition, *they* should like it. If you are not,

10• Don't get so worried about "what might happen afterwards" t
no practical progress.

11• You cannot foresee everything. Don't even *think* about it!

12• If you focus on helping your customer in practice, *now*, wher
need it, you will be forgiven a lot of 'sins'!

13• You can understand things much better, by getting *some* pra
(and removing *some* of your fears).

14• Do *early* cycles, on willing local mature parts of your user co

15• When some cycles, like a purchase-order cycle, take a long time, initiate them early, and do other useful cycles while you wait.

16• If something seems to need to wait for 'the big new system', ask if you cannot usefully do it with the 'awful old system', so as to pilot it realistically, and perhaps alleviate some 'pain' in the old system.

17• If something seems too costly to buy, for limited initial use, see if you can negotiate some kind of 'pay as you really use' contract. Most suppliers would like to do this to get your patronage, and to avoid competitors making the same deal.

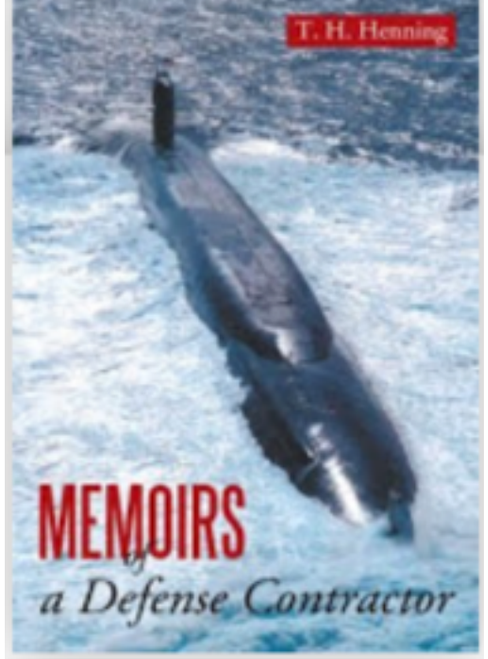
18• If *you* can't think of some useful small cycles, then talk directly with the real 'customer' or end user. They probably have dozens of suggestions.

19• Talk with end users in *any* case, they have insights you need.

20• Don't be afraid to use the old system and the old 'culture' as a launching platform for the radical new system. There is a lot of merit in this, and many people overlook it.

I have never seen an exception in 33 years of doing this with many varied cultures. Oh Ye of little faith!

http://www.gilb.com/tiki-download_file.php?fileId=41



LAMPS Sub.

Cleanroom Method
Robert Quinnan
uses Dynamic Design to Cost
on 2% (monthly) steps
and result is years of always on time under budget for 10 years on end.

On Military and Space Projects:
the highest state of art qualities

Cleanroom: IBM FSD, Federal Systems Division
(Agile ‘as it should be’: 1980-1990)
IBM SJ 4/1980, http://trace.tennessee.edu/utk_harlan/18/

Harlan Mills

DESIGN
The first guarantee of quality

"The first guarantee of **quality in design**
is in well-informed, well-educated, and well-motivated **designers**.

Quality must be built into designs, and cannot be inspected in or tested in.

Nevertheless, any prudent development process **verifies quality** through **inspection and testing.**

Inspection by peers in design, by users or surrogates, by other financial specialists concerned with cost, reliability, or maintainability
not only increases confidence in the design at hand,
but also provides designers with valuable lessons and insights to be applied to future designs.

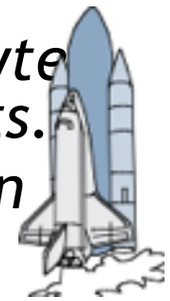
The very fact that **designs face inspections**
motivates even the most conscientious designers
to greater care, deeper simplicities, and more precision in their work."

inIBM sj 4 80 p.419
In

Mills, H. 1980. The management of software engineering: part 1: principles of software engineering. IBM Systems Journal 19, issue 4 (Dec.):414-420.
Direct Copy
http://trace.tennessee.edu/cgi/viewcontent.cgi?article=1004&context=utk_harlan
Library header
http://trace.tennessee.edu/utk_harlan/5/

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

- *"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 byte 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."*

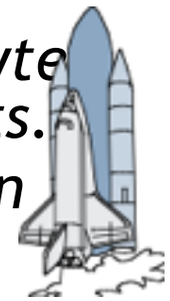


In the Cleanroom Method, developed by IBM's Harlan Mills (1980) they reported:

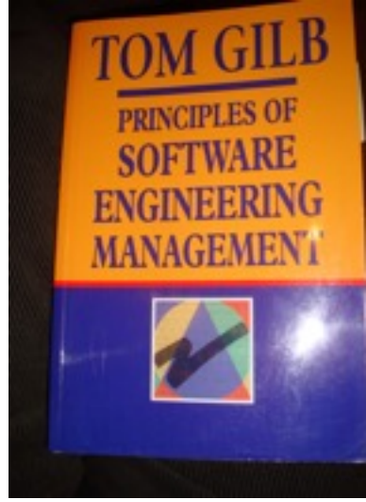
- *"Software Engineering began to emerge in FSD" (IBM Federal Systems Division, f[illegible] about*
- *T[illegible]cts – cost overruns, late deliveries, unreliable and incomplete software*
- *Today [Ed. 1980!], management has learned to expect on-time, within budget deliveries of high-quality software. 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 s[illegible]rocessors distrib[illegible]veries [Ed. Note 2[illegible]dget*
- *A mor[illegible]'*
- *- Whe[illegible]-years of softwe[illegible]million byte of pro[illegible]en projects.*
- *- Ther[illegible]e 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



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)

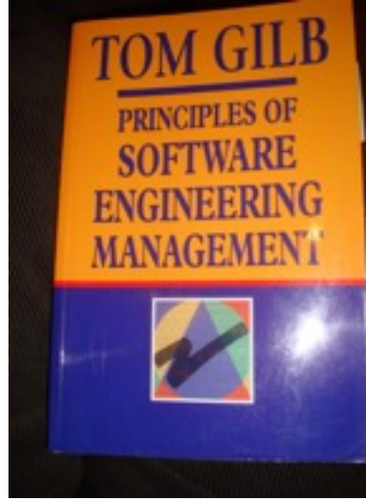
It is clear from this that they avoid the big bang cost estimation approach. Not only do they iterate in seeking the appropriate balance between cost and design for a single increment, but they iterate through a series of increments, thus reducing the complexity of the task, and increasing the probability 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. . . anagement farther by introducing design-t tegrated way to ensure that software technical m k by Figure 7.10] consists of developing a design,

-

_____He goes on to r by sacrificing 'planned capability.' When a sa t of each increment can proceed concurrently with the

'Design is an iterative

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

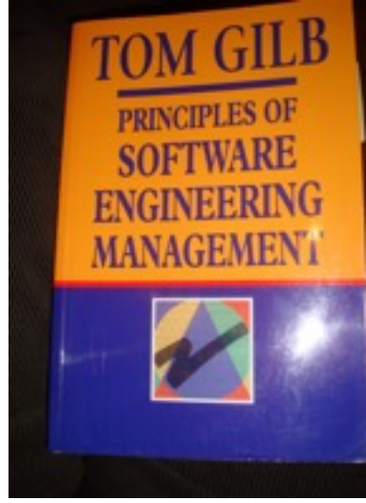
It is clear from this that they avoid the big bang cost estimation approach. Not only do they iterate in seeking the appropriate balance between cost and design for a single increment, but they iterate through a series of increments, thus reducing the complexity of the task, and increasing the probability 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 proceed concurrently with the

'Design is an iterative

It is clear from **in seeking the appropriate balance between cos** **thus reducing the complexity of the task, and increas** **d as the true cost of the increment becomes a**

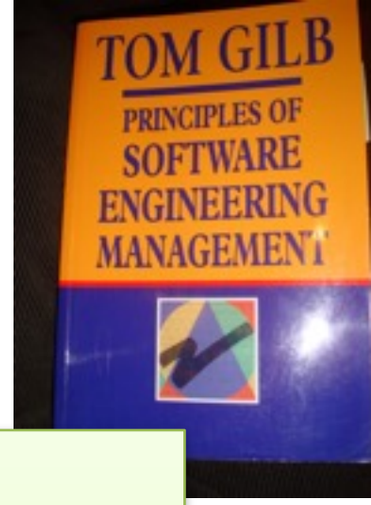
'When the developme **crements is computed.' (p. 474)**

Source: Robert E. Quir 1980, pp. 466~77

This text is cut from G

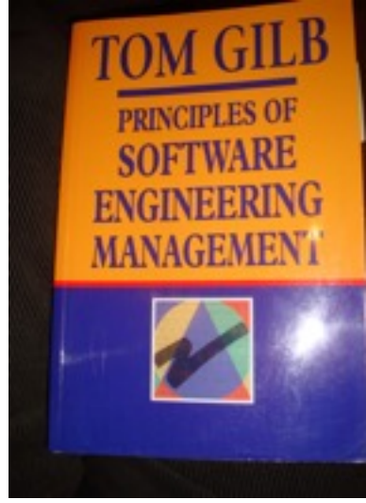
iteration process trying to meet cost targets by either *redesign* or by *sacrificing* 'planned capability'

Quinnan: IBM FSD Cleanroom
Dynamic Design to Cost



Design is an iterative process

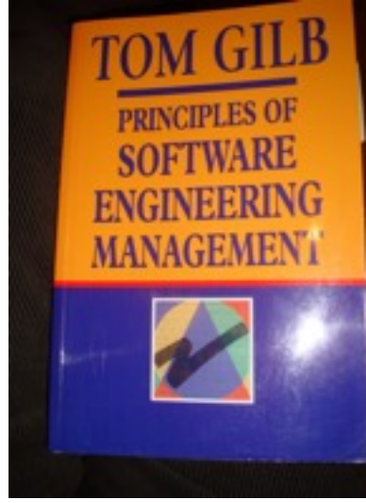
Quinnan: IBM FSD Cleanroom
Dynamic Design to Cost



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.

ed

of

)

Citibank London Case
Using Gilb's Evo & Planguage

Notice that designs that do not work
are immediately swapped
with hopefully better designs

20 Sept, 2015 Report on Gilb Evo method (Richard Smith, Citigroup)

- 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 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

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

Previous PM 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

Tom Gilb & Kai Gilb

www.Gilb.com

Our Column
http://tinyurl.com/AGILEMYTHS

5. Budget/cost/funding:

reduction, justifying, management (with or without recession).

Why conventional IT estimation of project costs and duration cannot actually work satisfactorily.

Unconventional estimation. Dynamic design to cost.

A process for delivering to arbitrary and inconvenient deadlines and budgets; even surprisingly changing resource constraints: and still apparently delivering planned quantified stakeholder value goals, on time, under budget - and even surprisingly early in practice

Main Points.
Advance cost-estimation for IT systems cannot be sufficiently accurate for purpose.

There are far too many cost-drivers (60) which are far too little understood.

There is however a simple, proven, little known method for getting control over resources, budgets and deadlines

“Dynamic Agile Feedback and Change”

“Estimation: A Paradigm Shift Toward Dynamic Design-to Cost and Radical Management”

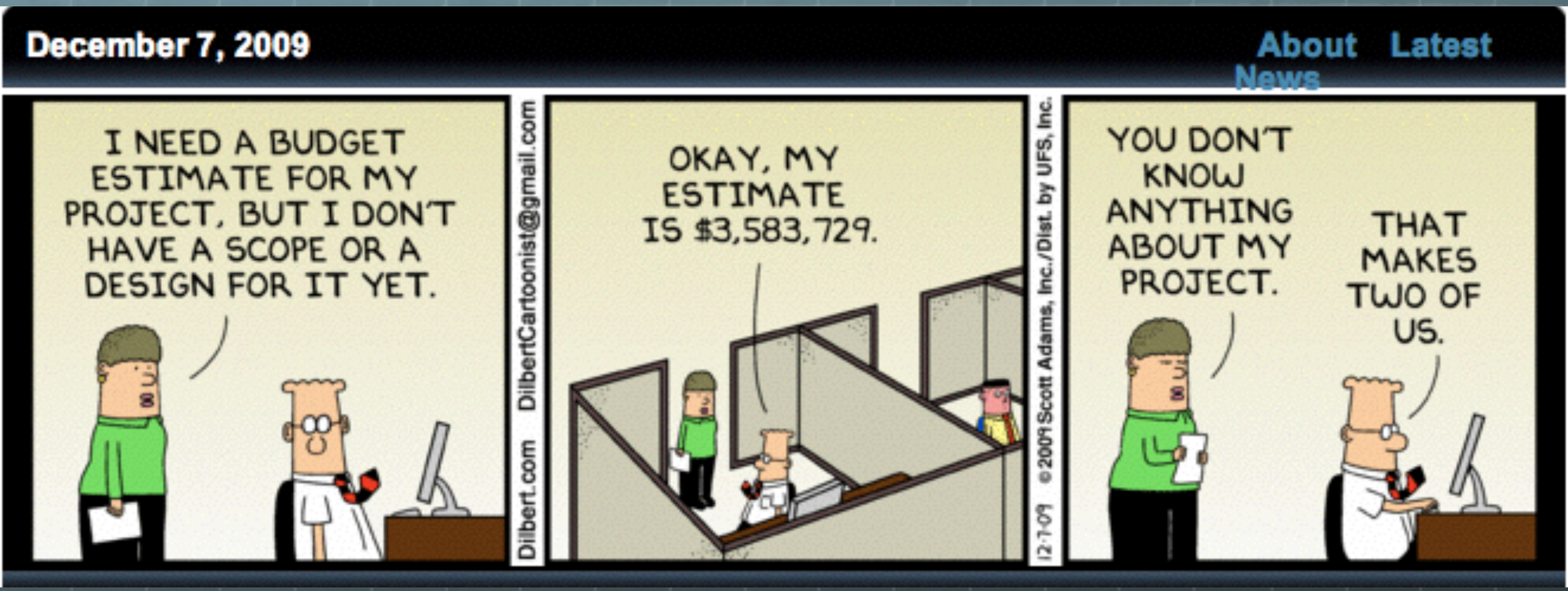
By Tom Gilb MASTER
Tom@Gilb.com
www.GILB.com
added or edited
21 Aug 2015

Based On A Paper

"Estimation: A Paradigm Shift Toward Dynamic Design-to Cost and Radical Management"

- Volume 13 Issue 2 of SQP journal - the March 2011 version.
  - Software Quality Professional, USA
  - The American Society for Quality (ASQ)
- http://www.gilb.com/tiki-download_file.php?fileId=460

The Obligatory Dilbert



The Risk Principles

- 1. DRIVERS: If you have not specified all critical performance and quality levels numerically – you cannot estimate project resources for those vague requirements.
- 2. EXPERIENCE: If you do not have experience data, about the resources needed for your technical solutions, then you cannot estimate the project resources.
- 3. ARCHITECTURE: If you implement your project solutions *all at once*, without learning their costs and interactions incrementally – you cannot expect to be able to understand the results of many interactions.
- 4. STAFF: If a complex and large professional project staff is an unknown set of people, or changes mid-project – you cannot expect to estimate the costs for so many human variables.
- 5. SENSITIVITY: If even the *slightest change* is made, after an 'accurate' estimation, to *any* of the requirements, designs or constraints – then the estimate might need to be changed *radically*. And – you probably will not have the information necessary to do it, nor the insight that you *need* to do it.

The Risk Principles (in Detail)

- The point being
  - that I want you to lose faith in convention notions of project estimation

- The risk of being very wrong is very high!

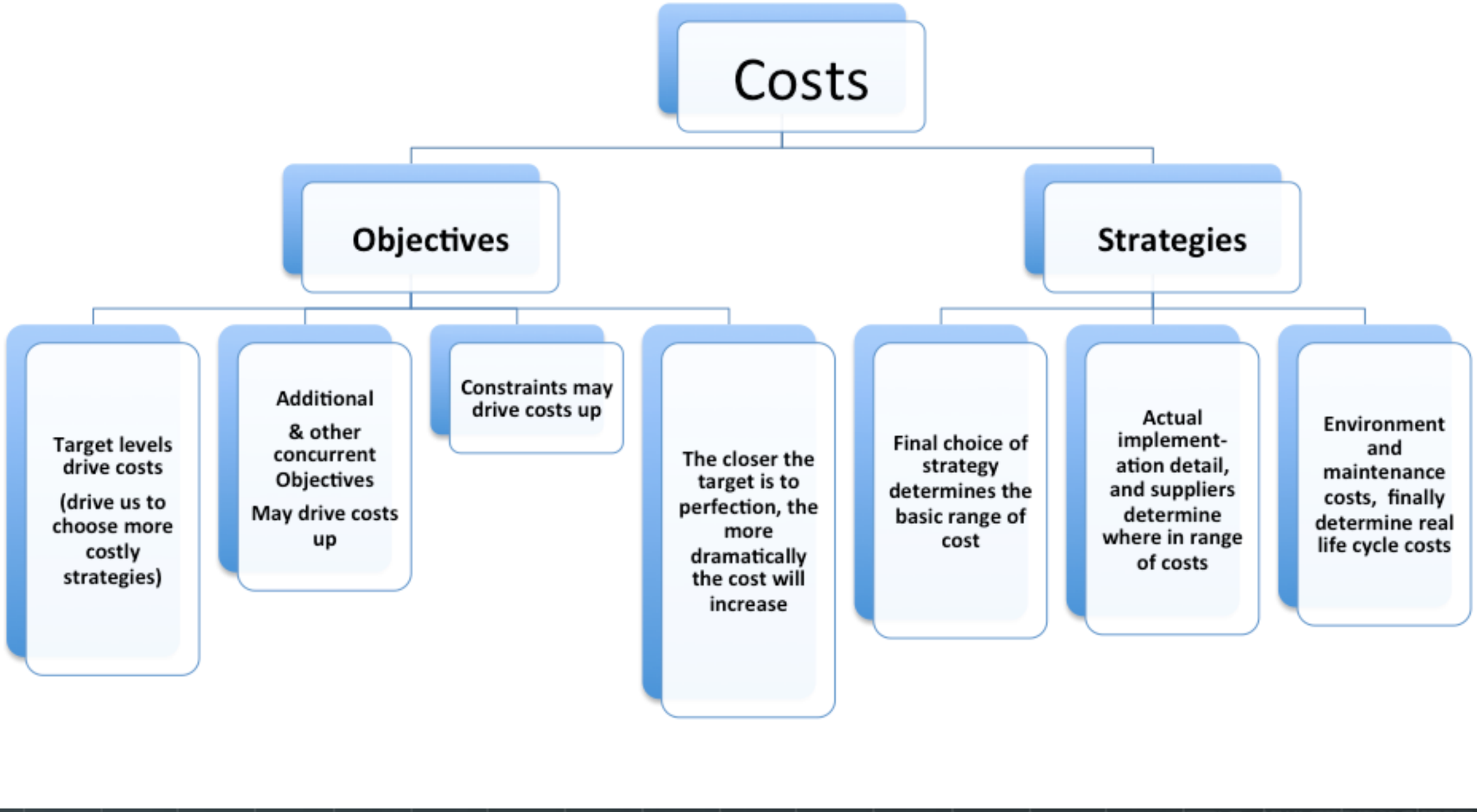
- The probability of being reasonably right is as big as you winning the Euro Lottery prize this week

- In fact if you sometime experience being ‘right1, it is Not due to estimation
  - Just probably due to slamming on the brakes, when the resources are used up.

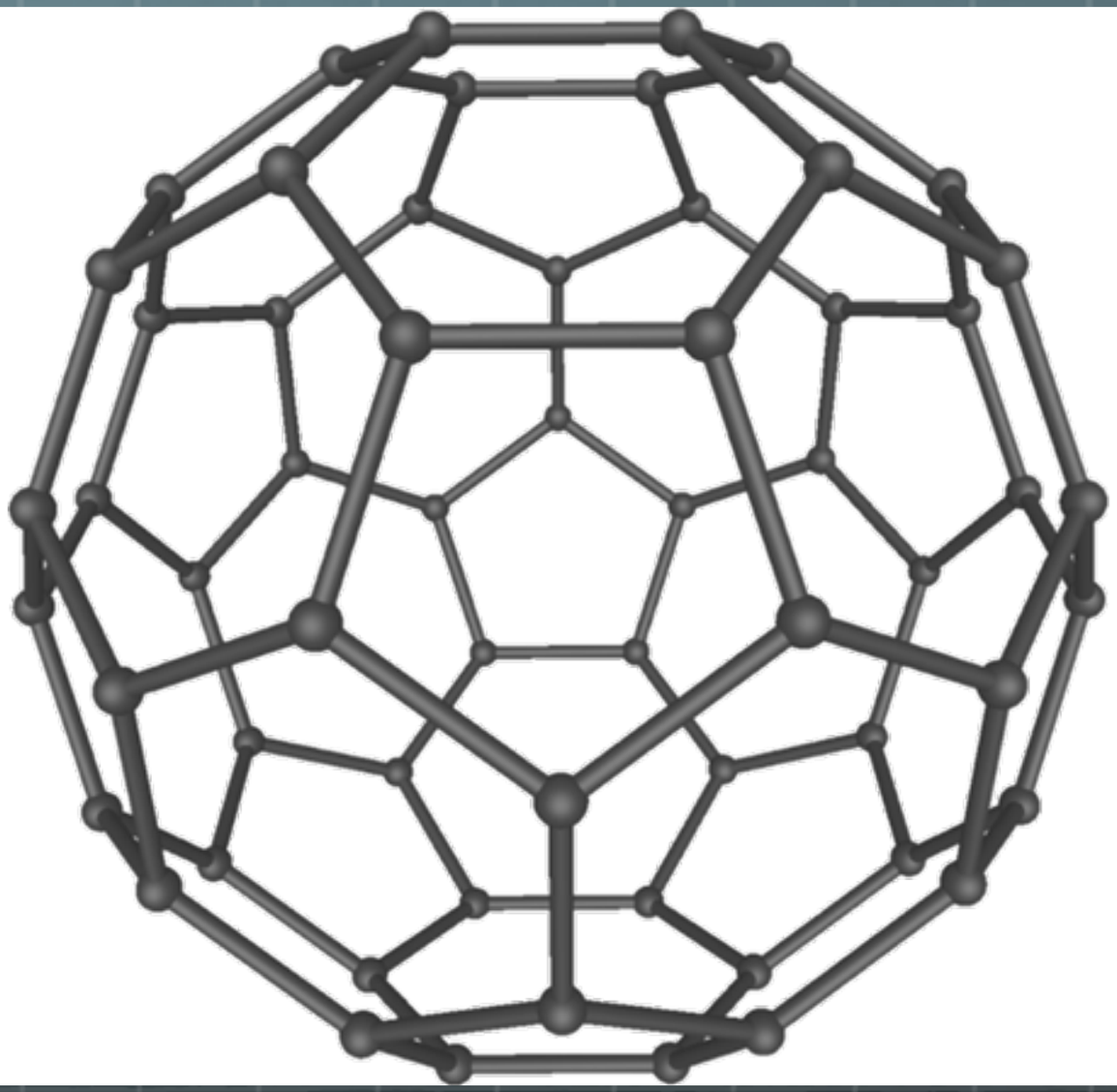
1. DRIVERS

- If you have not specified
  - all critical performance and quality levels numerically
    -
  - you *cannot* estimate project resources for those vague requirements.

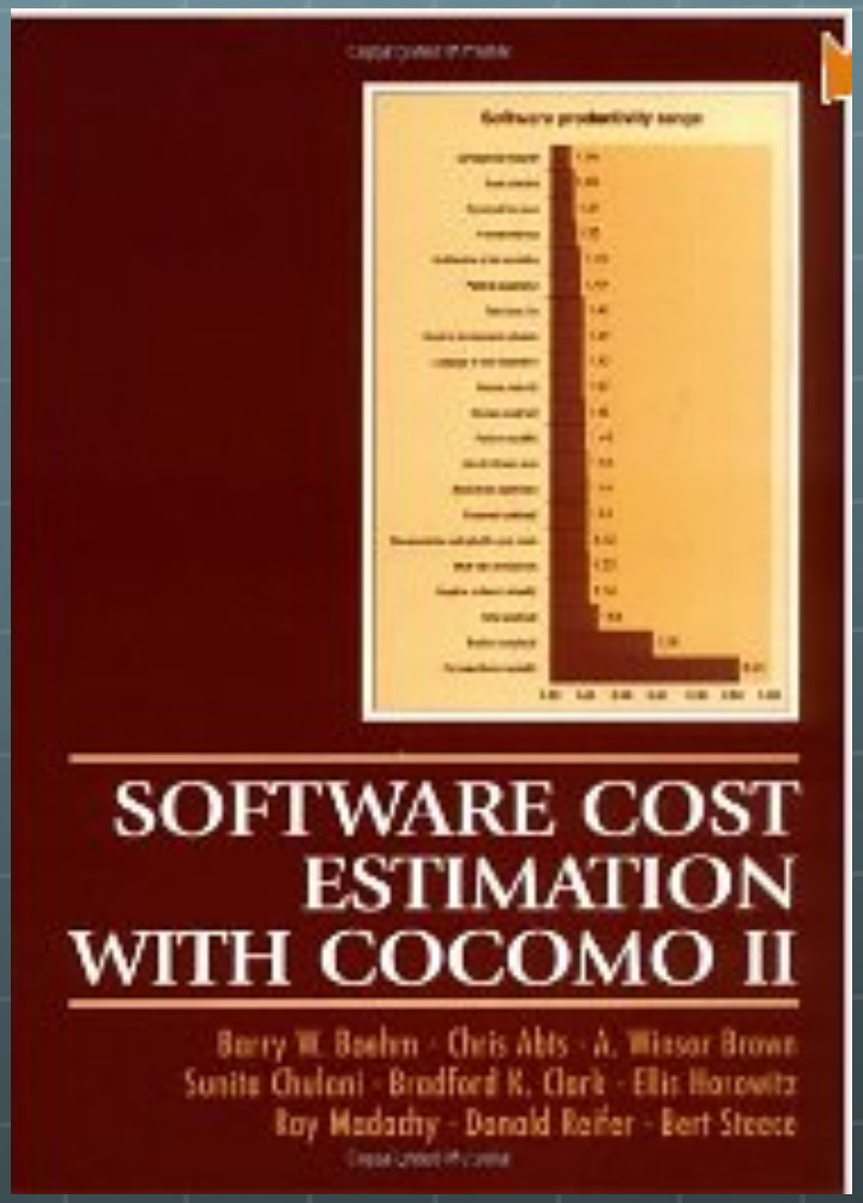
Drivers


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

Buckyball 60 points as many as Boehm's COCOMO Cost Drivers

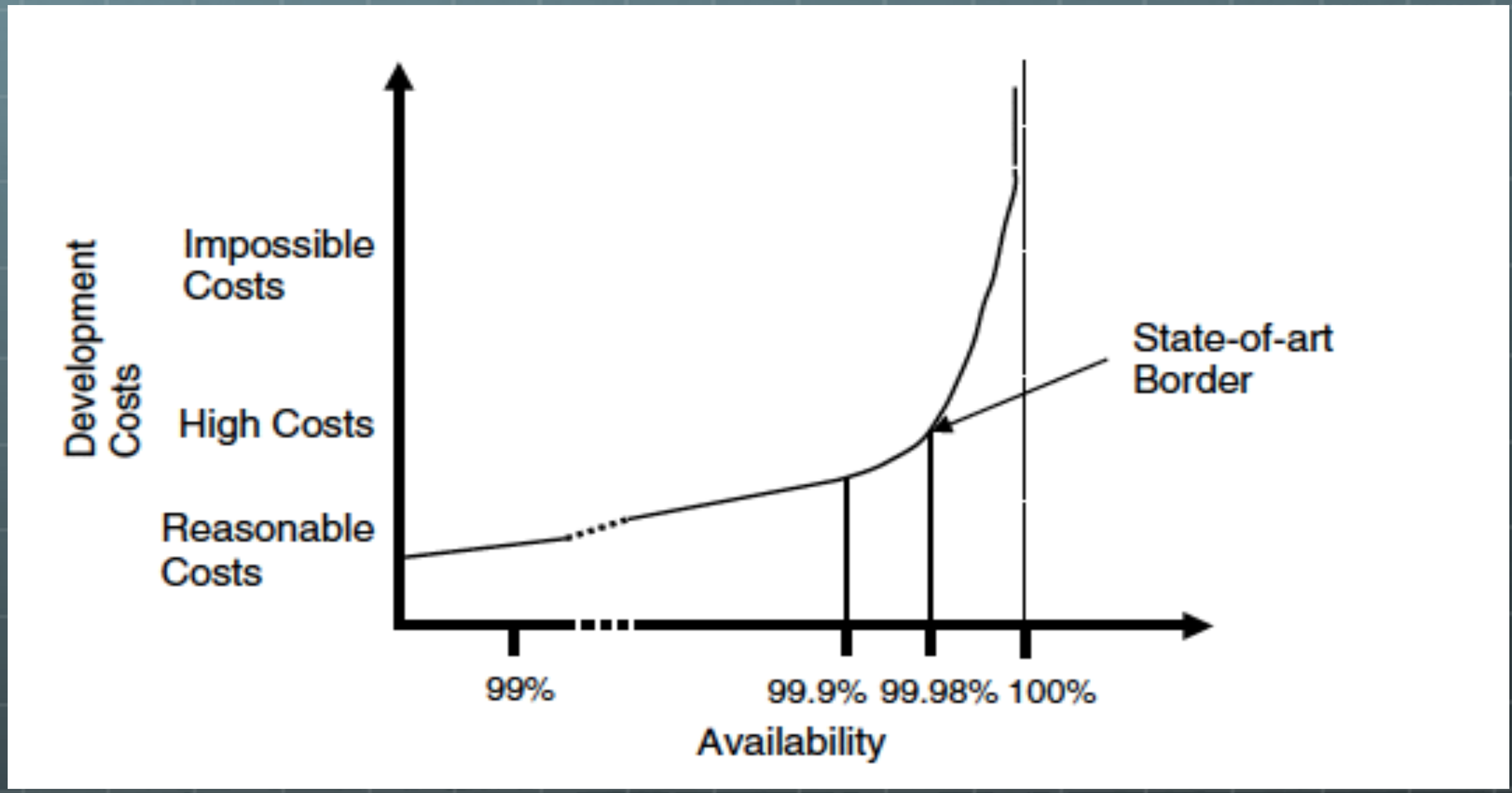


COCOMO & Boehm



http://csse.usc.edu/csse/research/COCOMOII/cocomo_books.htm

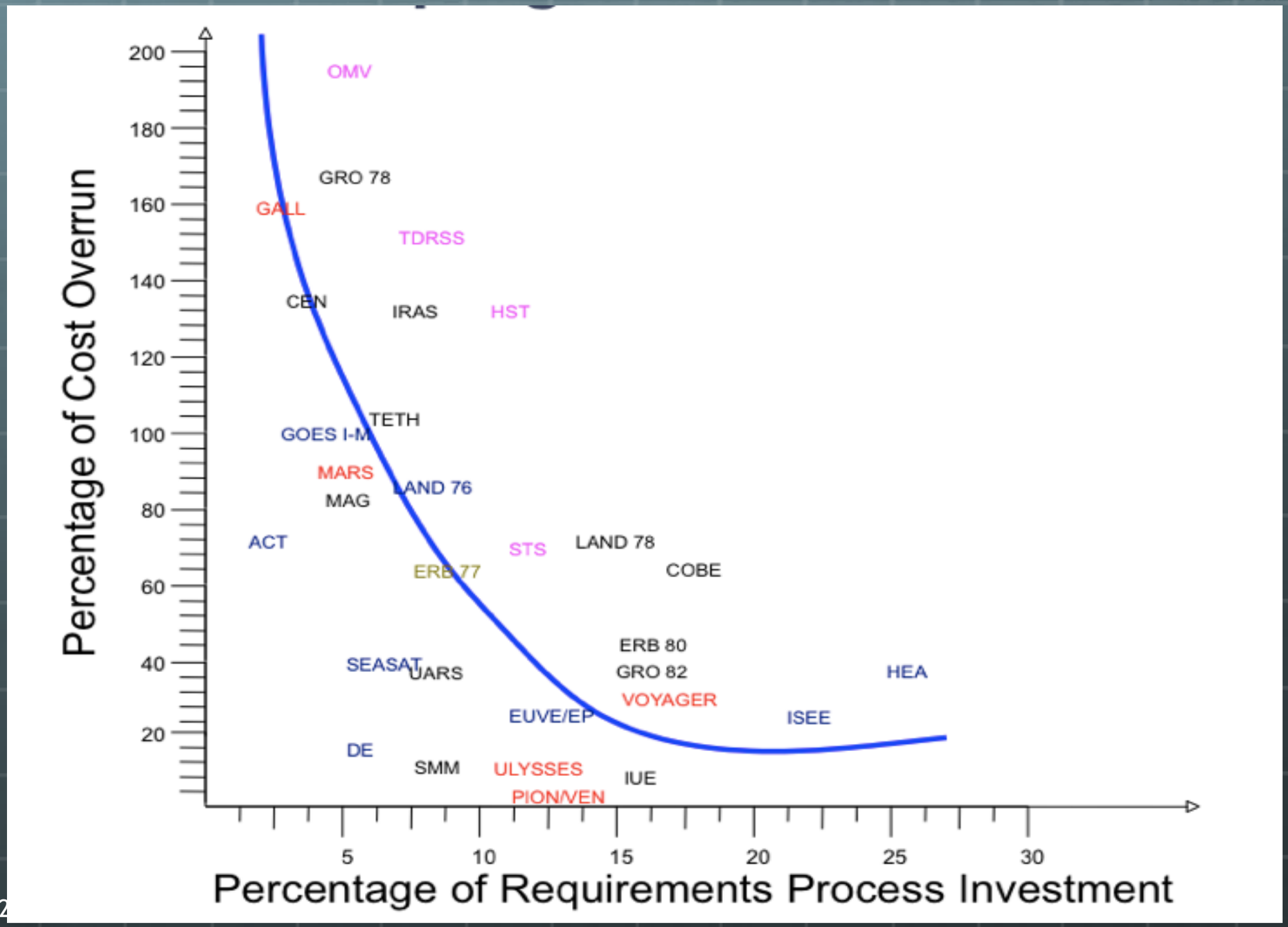
How much will ‘High Availability’ Cost?



2. EXPERIENCE

- If you do not have experience data,
  - about the resources needed for your technical solutions,
  - then you *cannot* estimate the project resources.

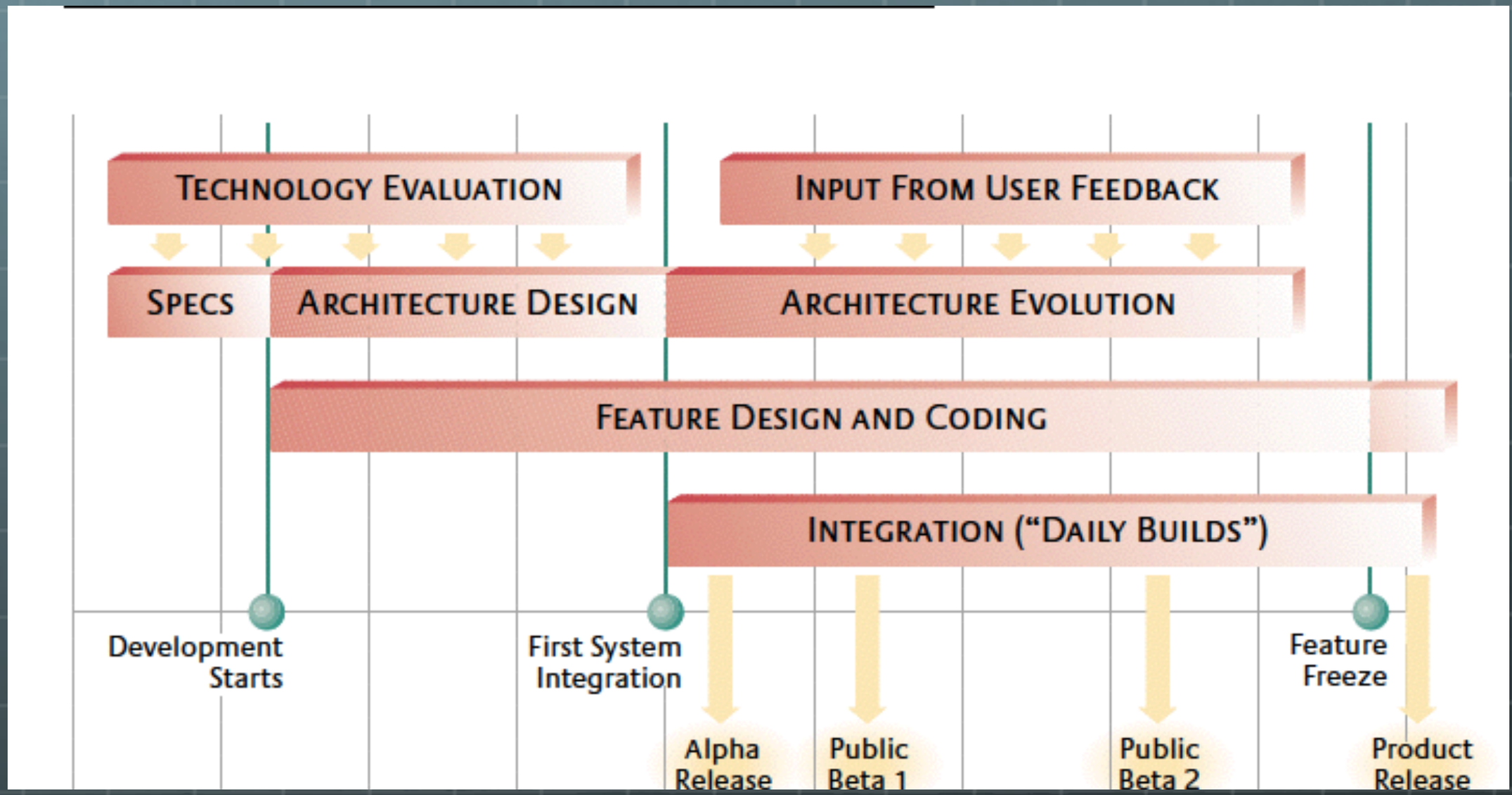
What is the cost difference if we use 5% for requirements, rather than 25%, if we are NASA?



3. ARCHITECTURE

- If you implement your project solutions *all at once*,
  - without learning their costs and interactions incrementally -
  - you cannot expect to be able to understand the results of many interactions.

Big Bang Fails: you don't know *exactly* why!



Small Delivery Steps Give Better Control: Cause and effect of failure is clearer

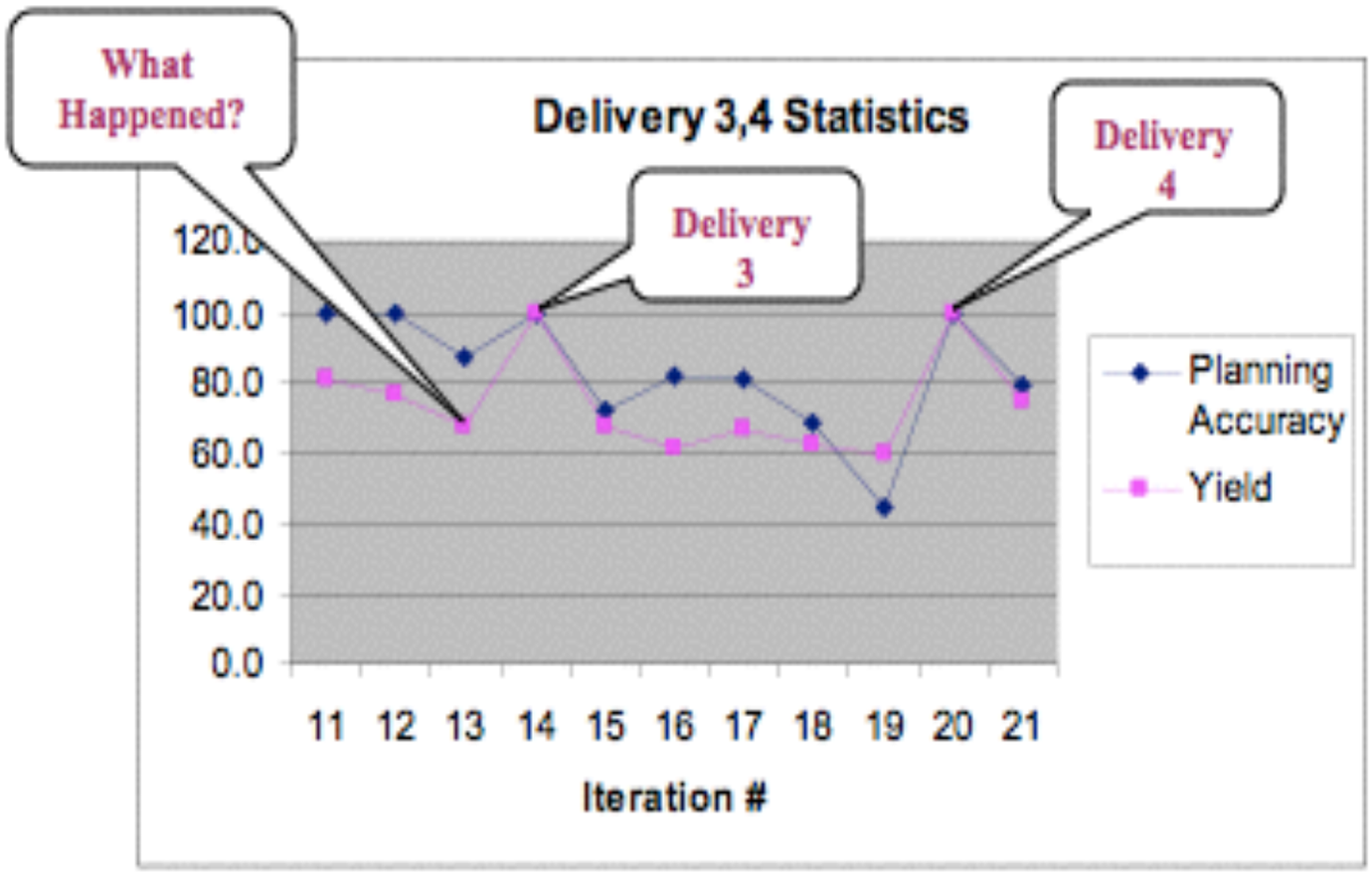
	Design Idea: Step 9 - Recoding			
Requirements	**Estimated Scale Impact**	**Estimated % Impact**	**Actual Scale Impact**	**Actual % Impact**
Objectives				
Usability.Productivity 65 <-> 25 minutes Past: 65 minutes. Tolerable: 35 minutes. Goal: 25 minutes.	**65 – 20 = 45 minutes**	**50%**	**65 - 38 = 27 minutes**	**95%**
Resources				
Development Cost 0 <-> 110 days	**4 days**	**3.64%**	**4 days**	**3.64%**

4. People

- If a complex and large professional project staff is
  - an unknown set of people,
  - or changes mid-project -
  - you cannot expect to estimate the costs for so many human variables.

Real Case:

Iterative measures, detected bad staff change (Honeywell, Berntsen)



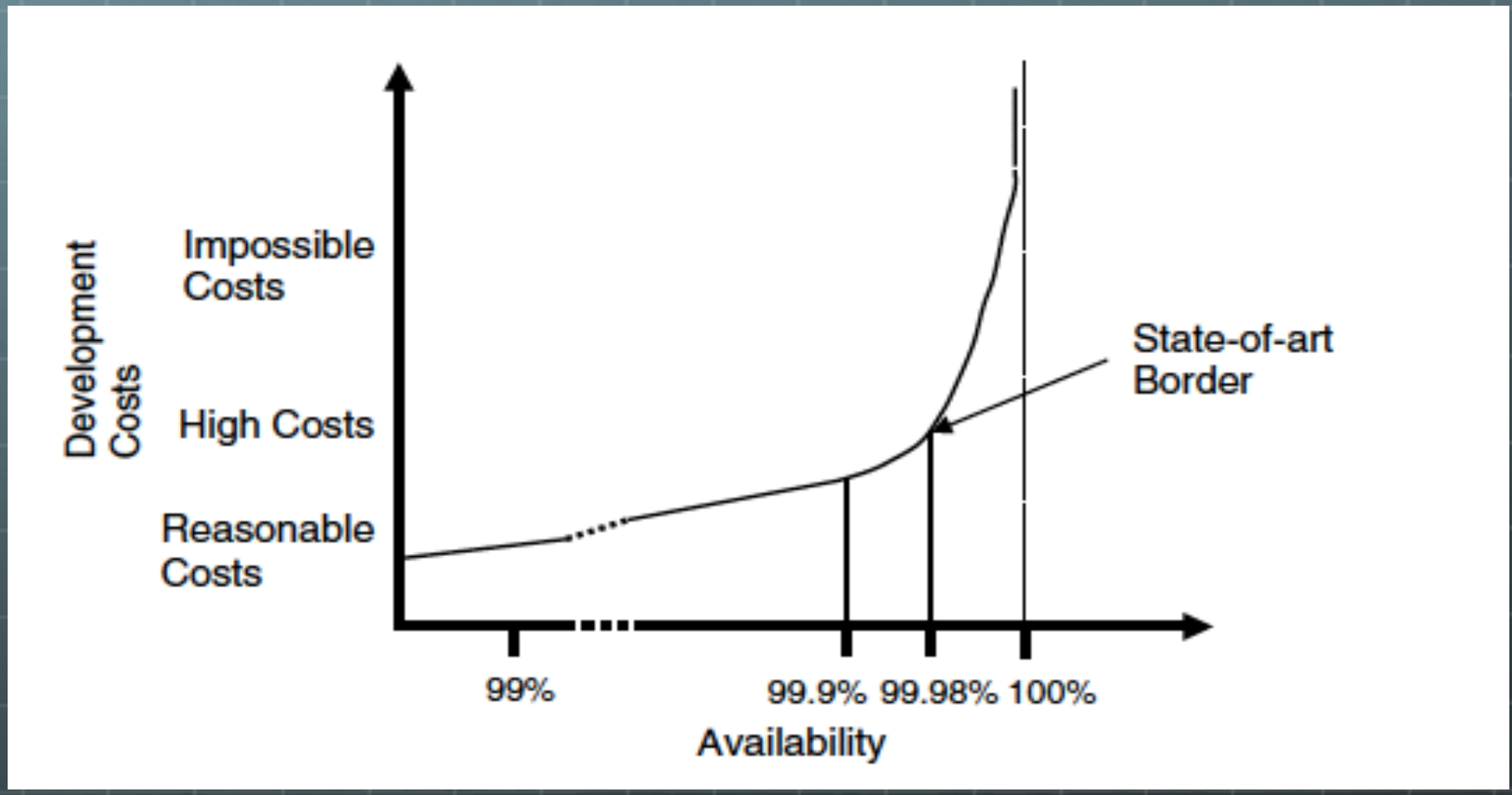
Measures

- Planning Accuracy - % of planned work that was completed.
- Build Yield - % of completed work that passed verification testing.

5. SENSITIVITY: to small changes in goals

- If even the *slightest change* is made,
  - after an ‘accurate’ estimation,
  - to *any* of the requirements, designs or constraints ,
    - then the estimate might need to be changed *radically*.
    - And – you probably will not have the information necessary to do it,
      - nor the insight that you *need* to do it.

99.98 - 99.90 = 00.08
80% to infinite costs



Real! : Primary Objectives for a £100 mill. Project

- Central to the Corporation's business strategy is to be the world's premier integrated <domain> service provider
- Will provide a much more efficient user experience
- 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
- Make the system much easier to understand and use than has been the case for the previous system
- A primary goal is to provide a much more productive systems development environment than was previously the case
- Will provide a richer set of functionality for supporting next-generation logging tools and applications
- Robustness is an essential system requirement
- Major improvements in data quality over current practices.

Why COCOMO Estimation Method is doomed to fail

- Availability
  - Very High
    - 99.90%
    - 99.98%
  - High
  - Medium
  - Low

Why COCOMO Estimation Method is doomed to fail

- Availability
  - Very High
    - 99.90%
    - 99.98%
  - High
  - Medium
  - Low

8 years x 2 to 3,000 people
(AT&T Case 5 ESS)

The Control Principles: the Good News

6. LEARN SMALL: Carry out projects in small increments of delivering requirements – so you can measure results and costs, against (short term) estimates.

7. LEARN ROOT: If incremental costs for a given requirement level (and its designs) deviate negatively from estimates – analyze the root cause, and change anything about the next increments that you believe might get you back on track.

8. PRIORITIZE CRITICAL: You will have to prioritize your most critical requirements and constraints: there is no guarantee you can achieve them all. Deliver 'high-value for resources-used' first.

9. RISK FAST: You should probably implement the design ideas with the highest value, with regard to cost and risk, early.

10. APPLY NOW: Learn early, learn often, learn well; and apply the learning to your current project.

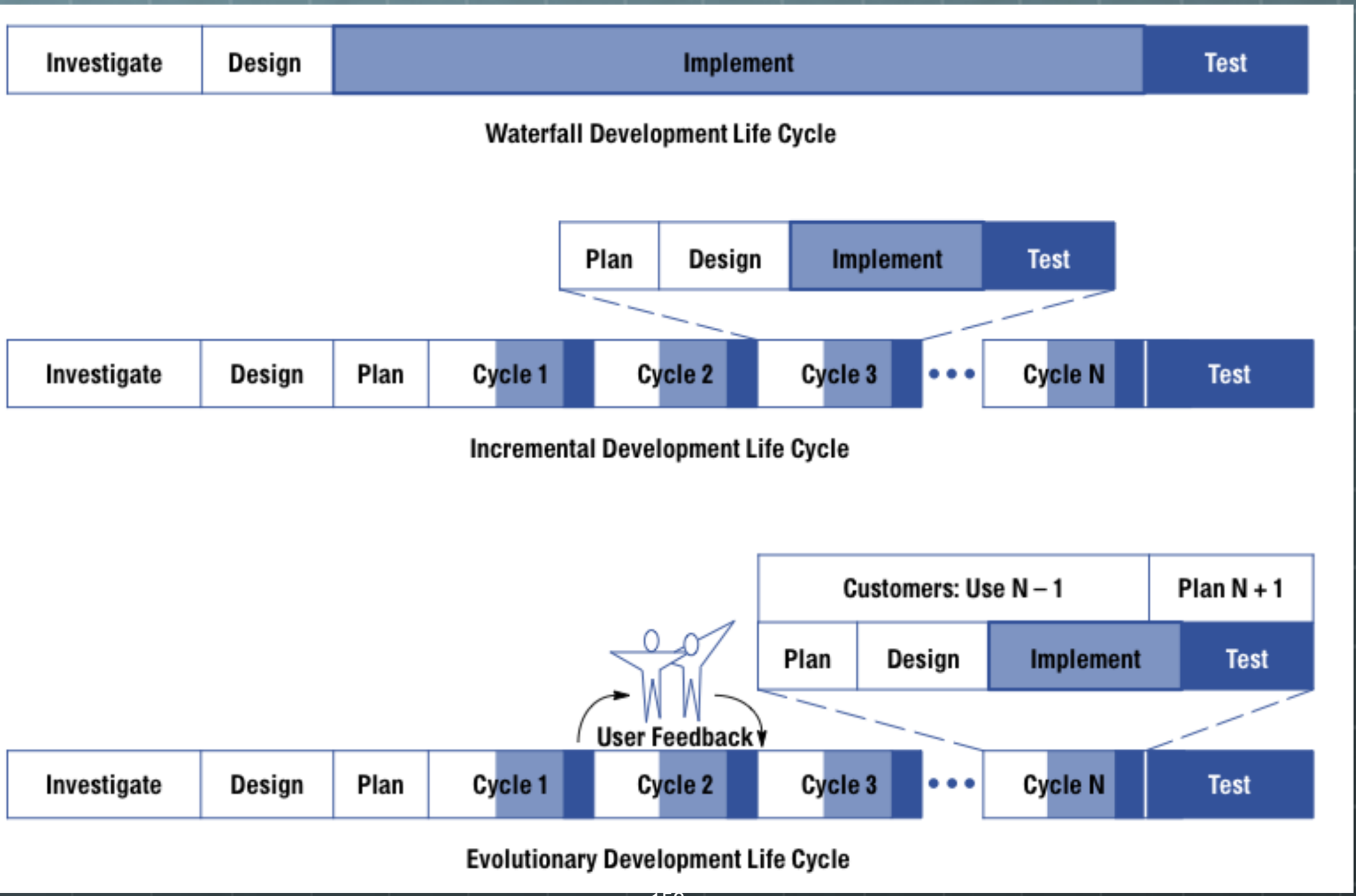
The Control Principles (shorter summary)

- The point here is that :
  - Given *any* arbitrary estimate of reasonable resources
  - You should be able to deliver so much prioritised value
    that you will stay in business, forever (meaning)
  - People will want to feed you money!

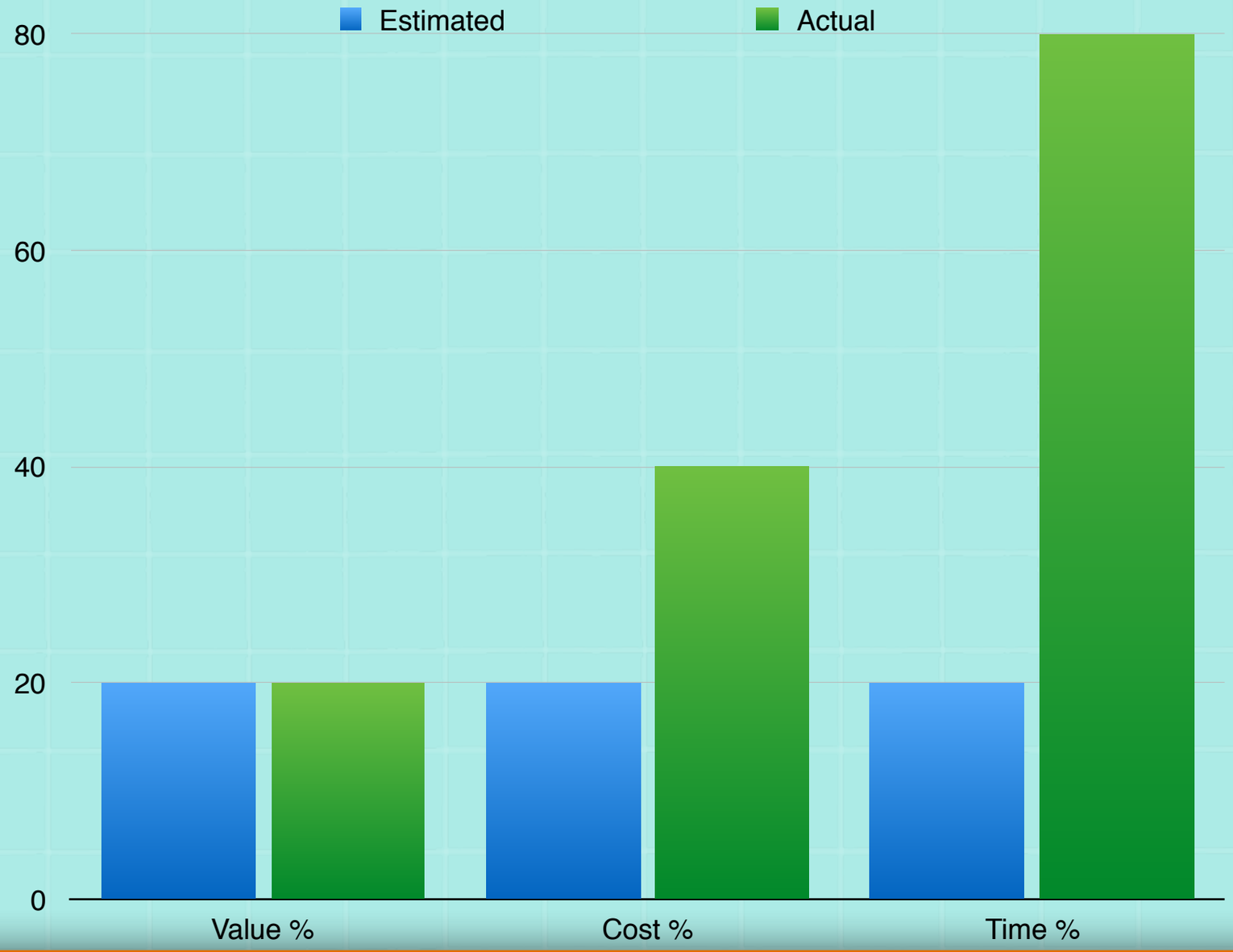
6. LEARN SMALL

- Carry out projects in *small increments* of delivering requirements -
- so you can measure *results* and *costs*,
- against (short term) estimates.
- And see cause and effect in useful detail

Breaking Result Deliveries into Small Chunks (Evo, HP, 1988 on)



Learning
20% done at 80% time, 40% cost

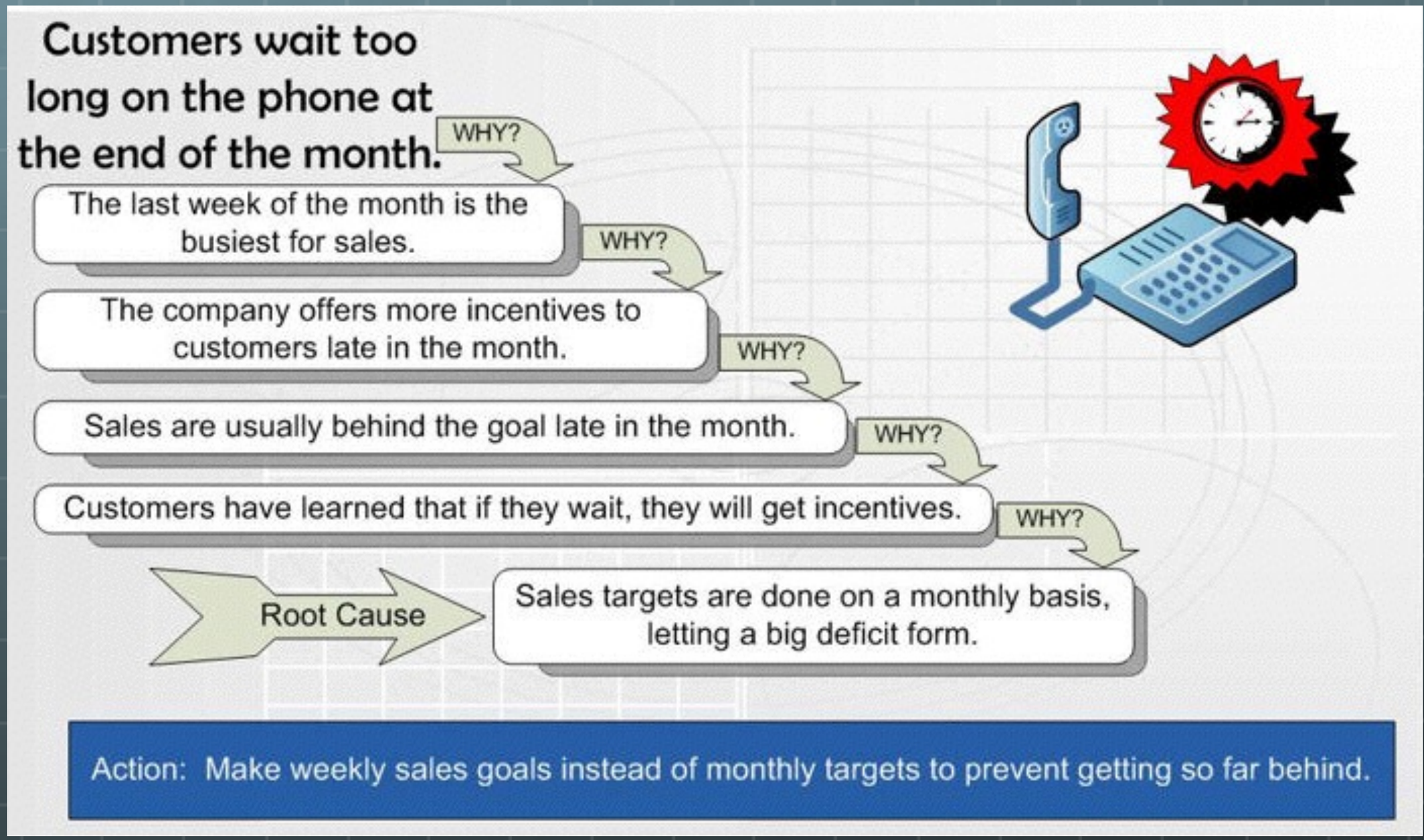


VP Book, Chart 4.5. At 20% of planned value delivery cycles (10 of 50 planned 2% iterations), we delivered 20% of value, as planned with current strategies.
But actual incremental costs are far too much. And if we do not act decisively now, change to cheaper strategies,
we will fail to deliver planned value by the deadline, and and/or fail to deliver planned value when we run our of budgeted money.

7. Learn the Root Cause (not unlike ‘Lean Startup’ !)

- If incremental costs for a given requirement level (and its designs) deviate negatively from estimates -
  - analyze the root cause, and
  - change anything
    - about the next increments
    - that you believe might get you back on track.

5 ‘Why’s find roots



8. Prioritize the Critical Value Deliveries

- You will have to
  - *prioritize* your most critical requirements ('deliveries')
  - and respect your resource constraints:
    - there is no guarantee you can achieve them all.
- Deliver:
  - 'high-value for resources-used'
  - *first*.

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

- *"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 developme developing and integrating over a hundred million bytes of program and data for ground and spa 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."*

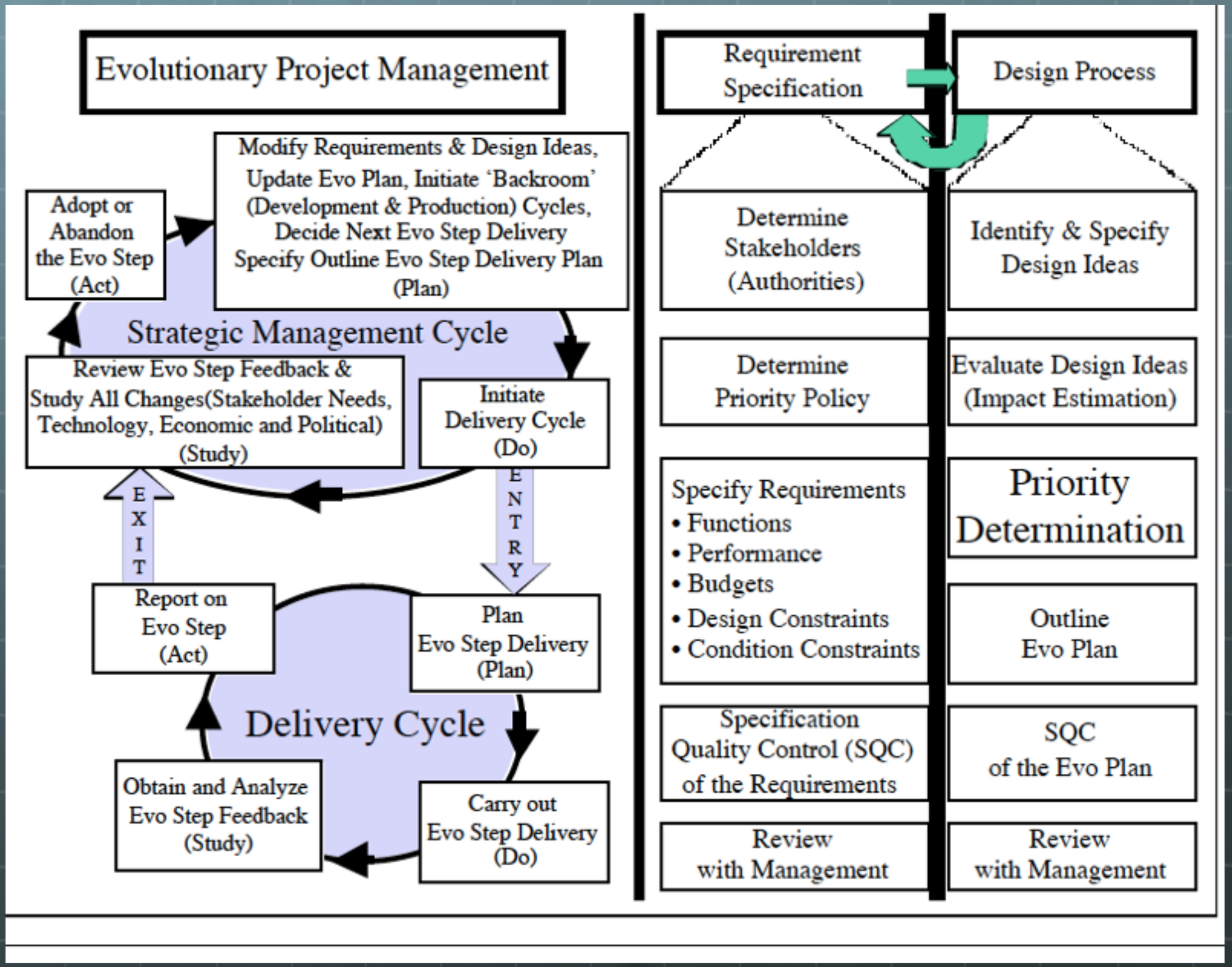
In the ‘Cleanroom’ Method, developed by IBM’s Harlan Mills (1980) : Early ‘Agile’ in practice! (1970’s)

- *“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 deliverie [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.”***

Cleanroom also uses

- Dynamic Design to Cost
  - See Quinnan in IBM SJ 4/1980 for details
- Like my friends at Confirmit in Oslo
- See Confirmit Case Studies at gilb.com/Downloads

Dynamic Prioritisation



9. Deliver Highest Value Early

- You should probably implement the design ideas (architecture components)
- with the highest value,
- with regard to cost and risk,
- early.

Which Designs are 'Risky' ?

Design Ideas

On-line Support: Gist: Provide an optional alternative user interface, with the users' task information for defined task(s) embedded into it.

On-line Help: Gist: Integrate the users' task information for defined task(s) into the user interface as a 'Help' facility.

Picture Handbook: Gist: Produce a radically changed handbook that uses pictures and concrete examples to *instruct*, without the need for *any* other text.

Access Index: Gist: Make detailed *keyword indexes*, using *experience* from *at least ten* real users learning to carry out the defined task(s). What do *they* want to look things up under?

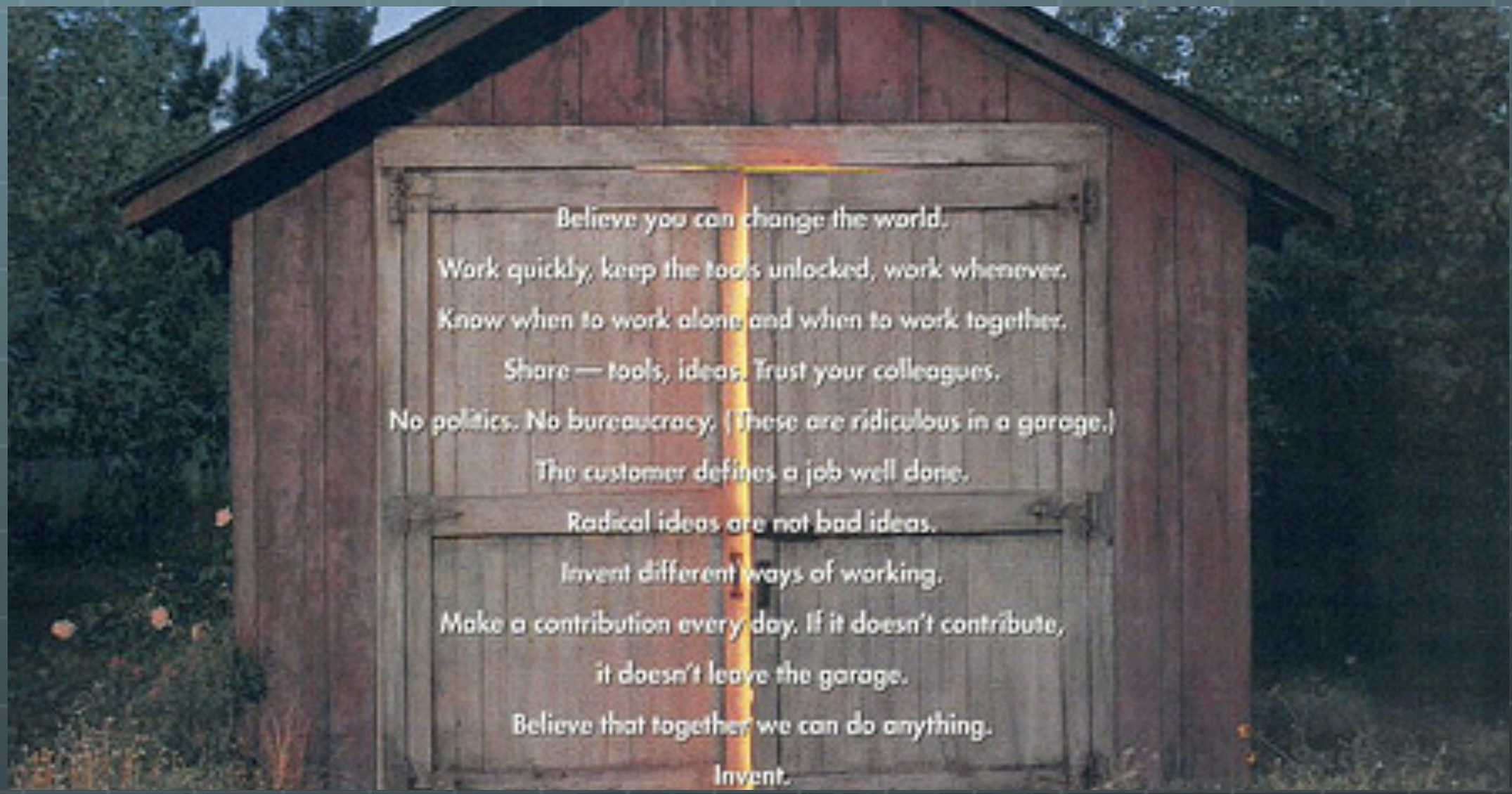
‘Impact Estimation’ Making ‘Risk’ Visible

	On-line Support	*On-line Help*	*Picture Handbook*	*On-line Help + Access Index*
Learning 60 minutes <-> 10 minutes				
Scale Impact	5 min.	10 min.	30 min.	8 min.
Scale Uncertainty	±3 min.	±5 min.	±10 min.	±5 min.
Percentage Impact	110%	100%	60%	104%
Percentage Uncertainty	±6% (3 of 50 minutes)	±10%	±20%?	±10%
Evidence	Project Ajax: 7 minutes	Other Systems	Guess	Other Systems + Guess
Source	Ajax Report, p.6	World Report, p.17	John B	World Report, p.17 + John B
Credibility	0.7	0.8	0.2	0.6
Development Cost	120 K	25 K	10 K	26 K
Performance to Cost Ratio	110/120 = 0.92	100/25 = 4.0	60/10 = 6.0	104/26 = 4.0
Credibility-adjusted Performance to Cost Ratio (to 1 decimal place)	0.92*0.7 = 0.6	4.0*0.8 = 3.2	6.0*0.2 = 1.2	4.0*0.6 = 2.4

10. APPLY NOW
(does this sound like 'Lean Startup' ?

- Learn early,
- learn often,
- learn well;
- and apply the learning to your *current* project.

“Make a contribution every day”



HP Rules of the garage

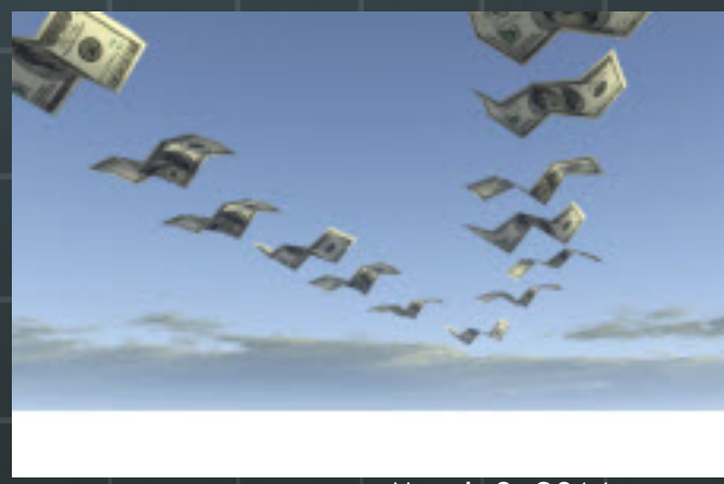
HP Garage Rules

(does this sound like ‘Lean Startup’ ?

- Believe you can change the world.
- Work quickly, keep the tools unlocked, work whenever.
- Know when to work alone and when to work together.
- Share tools, ideas. Trust your colleagues.
- No Politics. No bureaucracy. (These are ridiculous in a garage).
- The customer defines a job well done.
- Radical ideas are not bad ideas.
- Invent different ways of working.
- Make a contribution every day.
- If it doesn’t contribute, it doesn’t leave the garage.
- Believe that together we can do anything.
- Invent.

Simplified ‘Control Principles’

- 1. Do valuable stuff quickly
- 2. Measure values & costs
- 3. Adjust plans, if necessary
- Repeat 1-3 , until no net value



Advantages with Control Principles

- 1. You can*not* waste much time or money before you realize that you have false ideas
- 2. You *can* deliver value early, and keep people happy
- 3. You are forced to think about the *whole* system, including *people* (not just code)
- 4. So you are destined to see the true costs of delivering value - not just the code costs
- 5. You will learn a general method that you can apply for the rest of your career.

Disadvantages Control Principles

- 1. You cannot hide your ignorance from yourself any longer
- 2. You might have to do something not taught at school, or not taught in textbooks
- 3. There will always be people who criticize anything different or new
- 4. You cannot continue to hide your lack of ability to produce results, inside a multi-year delayed project.

Estimation ?

- Estimate, and re-estimate In small increments
- Make the most of *value* delivery
  - What does value actually cost?
- If you cannot deliver incremental value, <u>stop</u>
- A large estimate, or budget, is NOT important
  - But delivering value for money is far more important

Thanks

If you request by email,

Subject: 'Estimation Books/Papers'

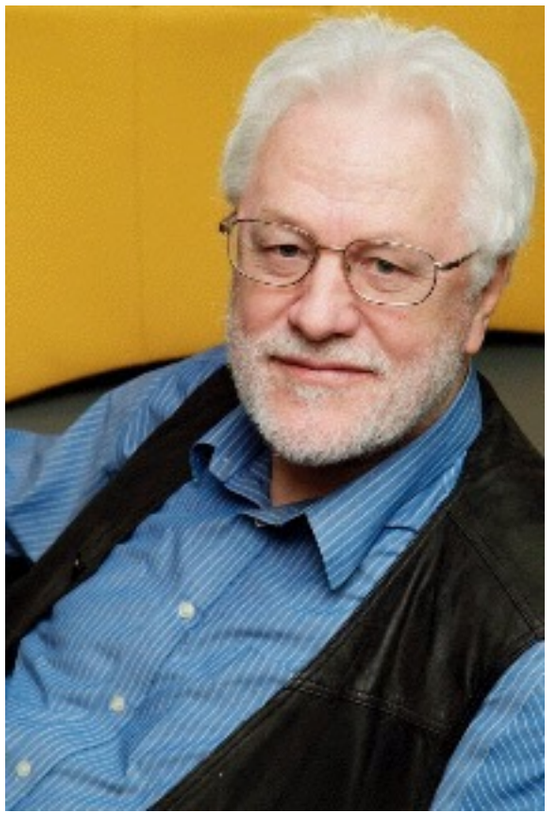
Tom@Gilb.com

I'll send you 2 free books (CE, VP) and some papers

www.Gilb.com

Sources

- Tiny url.com/ValuePlanning
  - quantifying critical objectives
  - impact estimation tables



- gilb.com
- tom@Gilb.com
- @ImTomGilb