Advanced

Process Owner

Paper at **gilb.com/dl799**

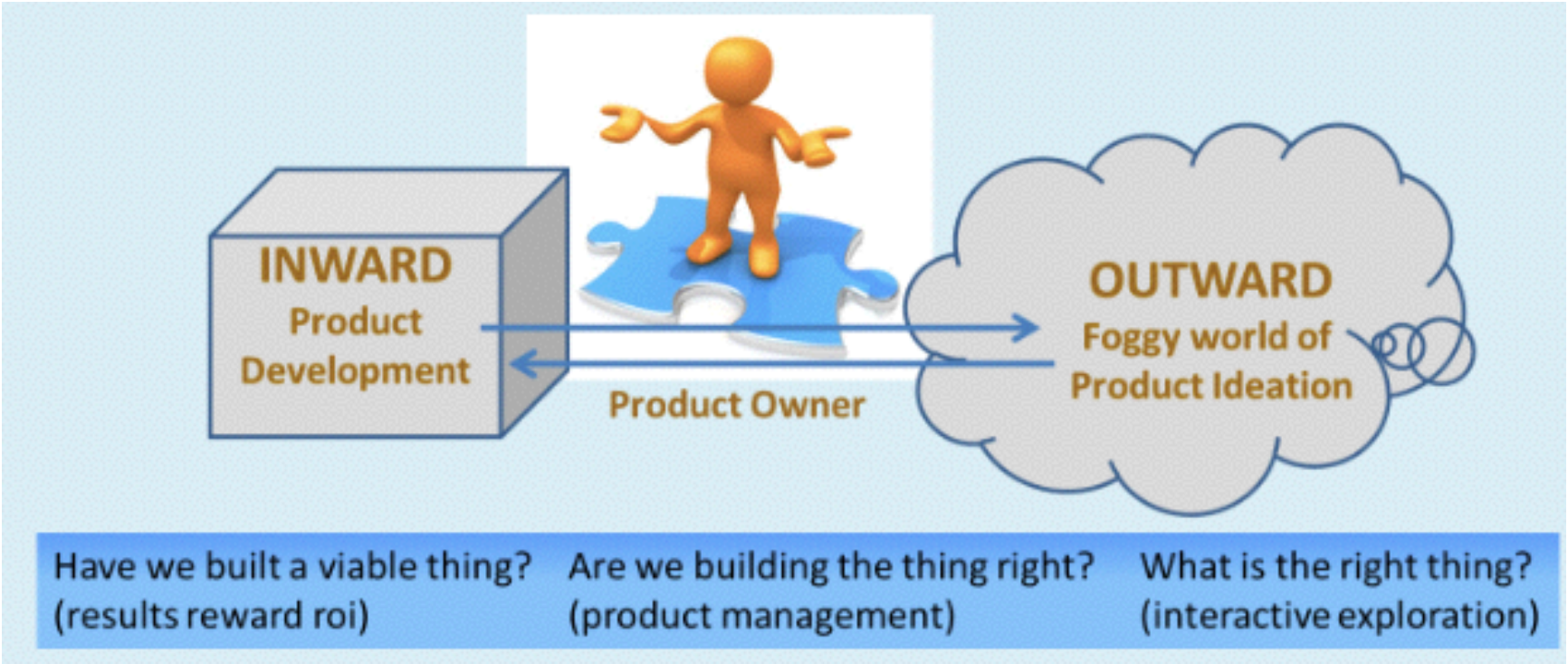
All Gilb's 11 Agile Mythodology Columns

tinyurl.com/GilbMyth

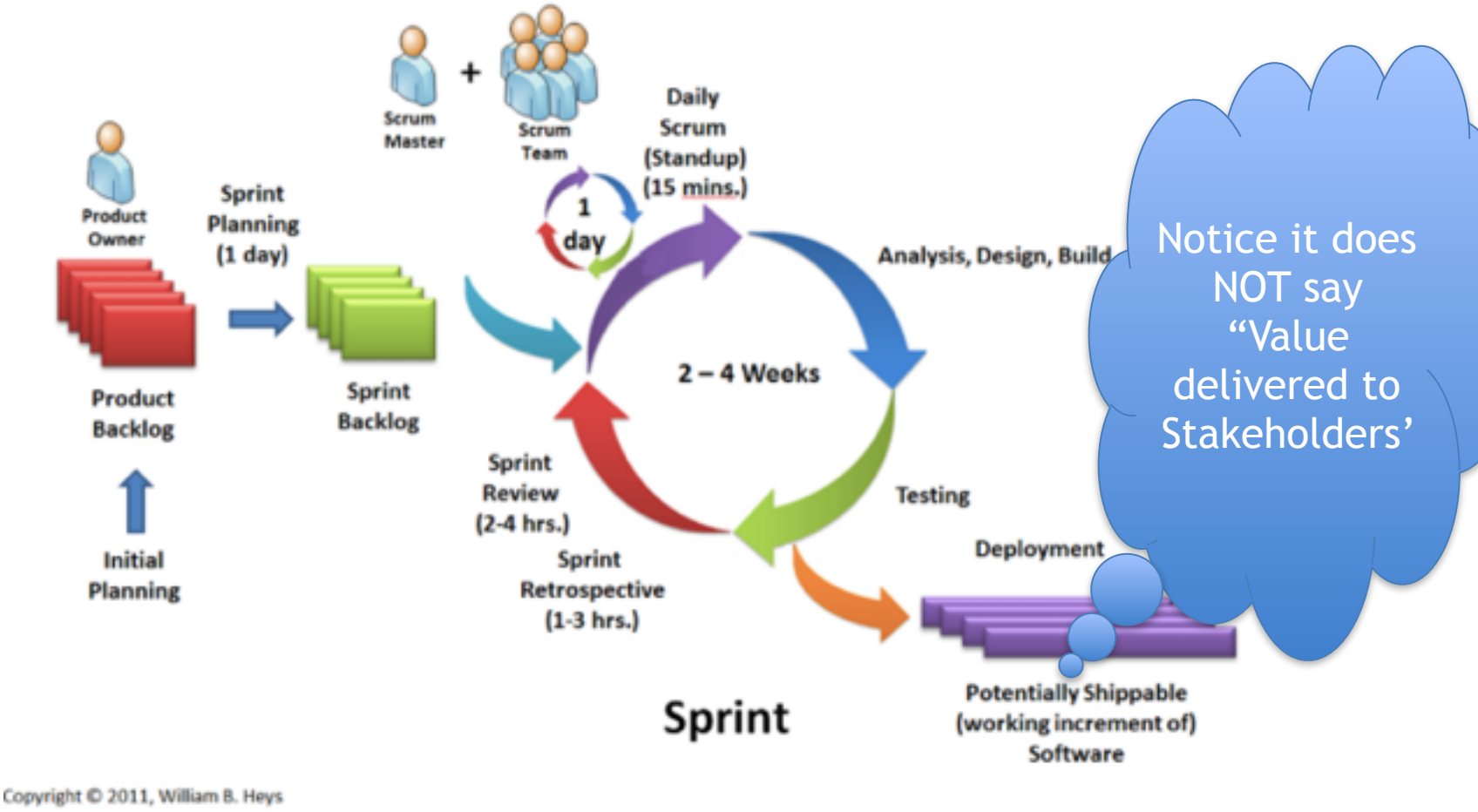
Tom Gilb
MASTER

5 minute Lightning Talk

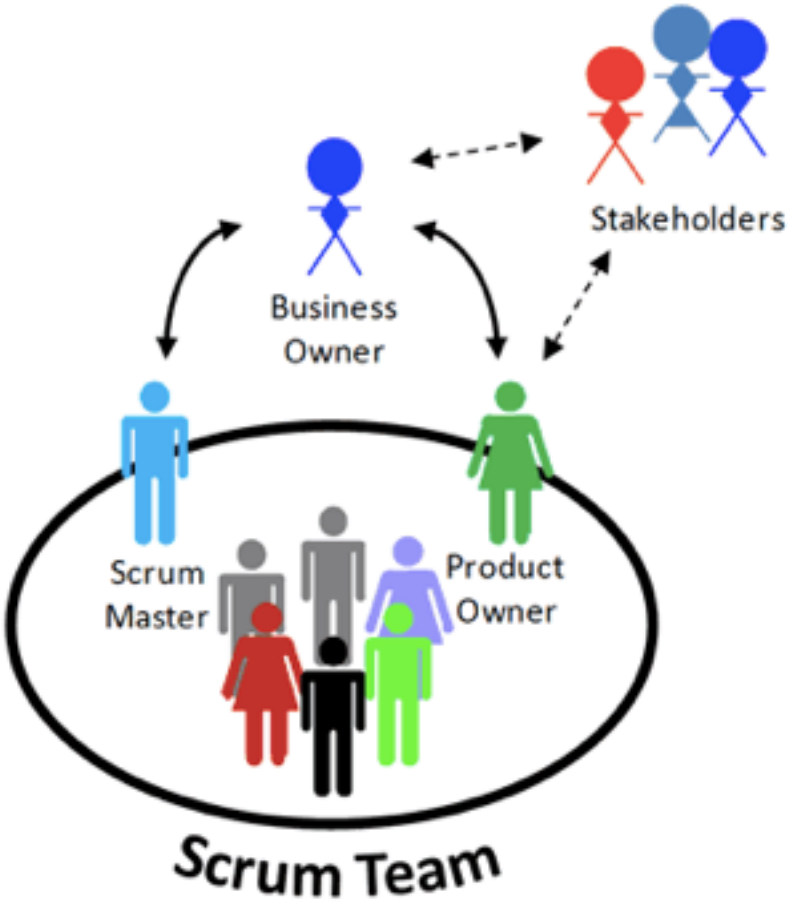
Basic Product Owner Concept



Product Owner as Input to Scrum Team



Input sources to P.O. Stakeholders and Business Owner



http://www.executivebrief.com/agile/how-to-scrum/s

Requirements and Design:
Related but Separated
and Specialized 'Engineering' Processes

Stakeholders
(as source of all requirements)

- **Requirements Engineer**
  - **Architecture**
  - **Engineer**
    - **'Backlog'**
  - **Test Engineer**
- **Business Owner**
- **As Funder and Sponsor**
  - **Users and Customers**
  - **(as recipients of VALUE from system)**

Advanced ‘Product Owner’ and the ‘Value Options List’ (VOLare!)

Stakeholders

(as source of all requirements)

- **Requirements**
- **Engineer**
  - **Architecture**
  - **Engineer**
    - **Value Options List**
    - **With Value/cost + info**
      - **Dev Team (s)**
      - **Prioritize Value options to real Value Targets**
  - **Test Engineer**
- **Business Owner**
- **As Funder and Sponsor**
  - **Users and Customers**
  - **(as recipientsof VALUE from system)**

Advanced: = 'Evo' Agile Method *

Advanced Product Owner

- Value Focussed
- Real Engineering
- Requirements = Value
- Stakeholder Focussed (all 50+ !)
- Qualities Focussed (all 30)
- Measurable Value Stream
- Architecture Engineering

Conventional 'Product Owner'

- Code Focussed
- Craft ('Softcraft')
- Reqts = Function, Story
- User Customer Focussed (all 2)
- Bug Focussed (not even MTBF)
- Code Stream
- No clear design concept

* CE book, Chapter 10: Evolutionary Project Management: http://www.gilb.com//tiki-download_file.php?fileId=77

POo
(A Wave to Milne)

- *The 'Owner of Product,' made stories*
- *So that Burndown was ferocious velocities*
- *But the Value delivered*
- *Made Stakeholders so shivered*
- *That the Owner turned into a Loner*

Cheers Milne!

- There once was a ‘soft engineer’
- Who knew no ‘complexity fear’
- He *sorted* a project
- That beggared his logic
- So, ‘**Done**’!
  - who’s having a beer

Last Slide

Want the detail free?

- Email me
  - Tom @ Gilb . Com
  - Subject: **BOOK**
  - And or
  - Subject: **COURSES**

Book For Mature IT Engineers
Not For Softcrafters

The Policy

- Advanced Product Owner' Policy: System 'Requirements Engineer' (RE).
  - **Background**: this policy defines the expectations for a 'Product Owner' (PO) for serious, critical, large, and complex systems.
    - This implies that it is not enough to manage a simple stream (Backlog) of 'user stories' fed to a programming team.
    - It is necessary to communicate with a systems engineering team, developing or maintaining the 'Product'.
      - *System* implies management of all technological components, people, data, hardware, organization, training, motivation, and programs.
      - *Engineering*: means systematic and quantified, 'real' engineering processes, where proactive design is used to manage system performance (incl. all qualities) attributes and costs.

1. COMPLETE REQUIREMENTS:

- The RE (Requirements Engineer) is responsible for **absolutely all requirements specification** that the system must be aware of, and be responsible for to all critical or relevant stakeholders.
  - In particular, the RE is
    - not narrowly responsible for requirements from users and customers alone.
    - They are responsible for all other stakeholders,
      - » such as operations, maintenance, laws, regulations, resource providers, and more.

2. **QUALITY REQUIREMENTS:**

- The RE is **responsible for the quality level**, *in relation to official standards*, of all requirements they transmit to others.
  - They are consequently responsible for making sure the quality of incoming raw requirements, needs, values, constraints etc. is good enough to process. No GIGO.
  - If input is not good quality,
    - they are responsible for making sure it is better quality,
    - or at least clearly annotated where there is
      - » doubt, incompleteness, ambiguity and any other potential problems, they cannot resolve yet.

3. ARCHITECTURE:

- The Requirements Engineer is NOT responsible for any architecture or design process itself.
  - This will be done by professional engineers and architects.
- They are however very much responsible for a *complete* and *intelligible quality* set of requirements,
  - transmitted to the designers and architects.
- The are also responsible for transmitting quality-controlled architecture or design specifications to any relevant system builders.
  - These are the designs which are input requirements to builders. Effectively they are ‘design constraints requirements’.

4. Priority Information:

- The Requirements Engineer is NOT responsible for *prioritization* of requirements.
- Prioritization is done dynamically
  - at the project management (PM) level,
  - based on prioritization signals in the requirements,
  - and on current feedback and experience in the value delivery cycles (Sprints).
- The primary responsibility of the Requirements Engineer,
  - is to *systematically and thoroughly collect and disseminate* all relevant priority signals, into the requirement specification;
  - so that intelligent prioritization can be done at any relevant level, and at any time.

End of Summary in Detail

The Long Version of the Talk for those who want detail that cannot be given in 5 minutes

ADVANCED PRODUCT OWNER Col. 12
GILBS MYTHODOLOGY COLUMN Agile Record 18 Feb 2014
We are going to argue that the normally defined role of Product Owner (PO) is inadequate for projects that have serious multiple quality requirements, and consequent architecture processes, to deliver the necessary levels of performance and quality.

http://www.gilb.com/dl799

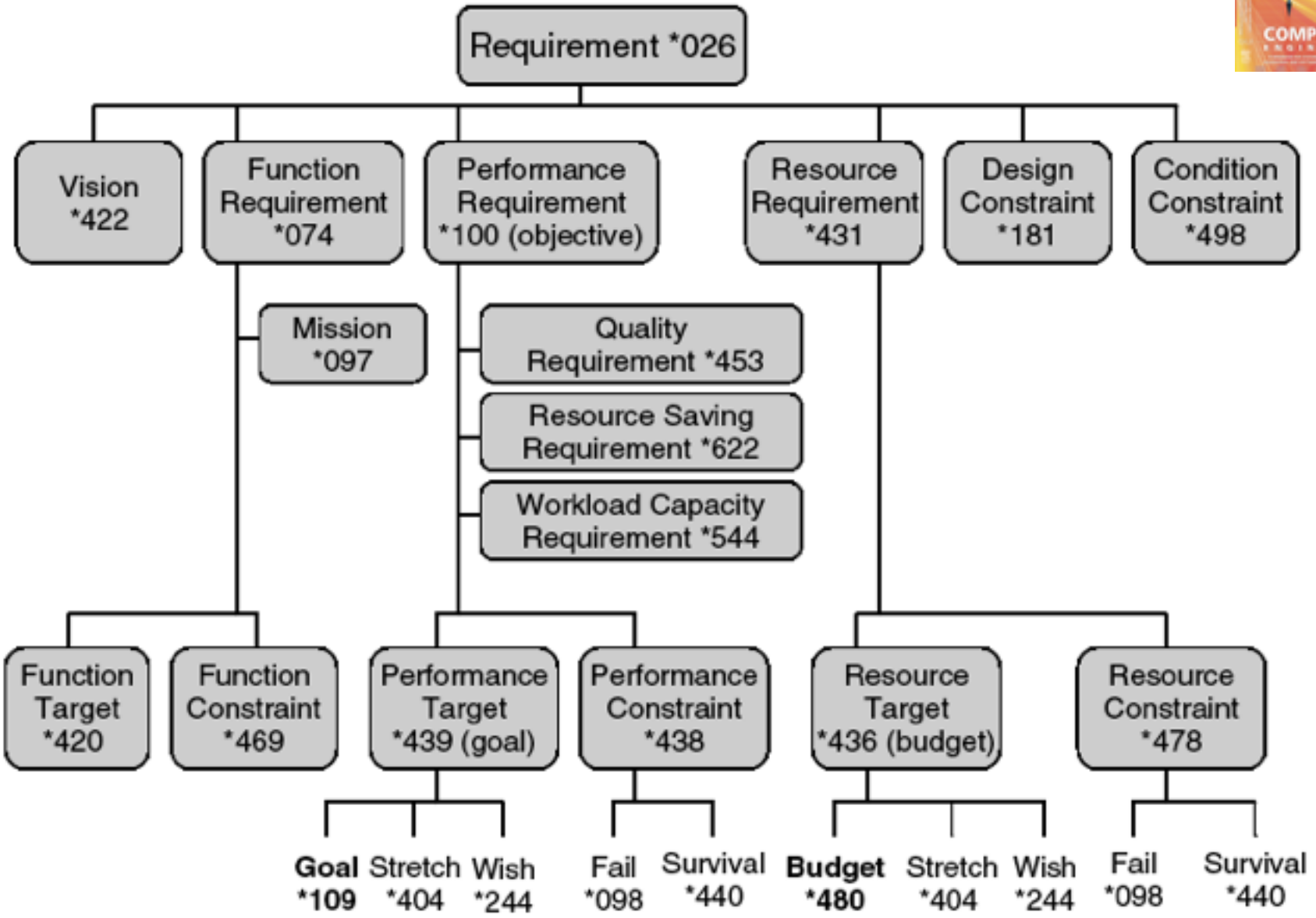
The Policy

- Advanced Product Owner' Policy: System 'Requirements Engineer' (RE).
  - **Background**: this policy defines the expectations for a 'Product Owner' (PO) for serious, critical, large, and complex systems.
    - This implies that it is not enough to manage a simple stream (Backlog) of 'user stories' fed to a programming team.
    - It is necessary to communicate with a systems engineering team, developing or maintaining the 'Product'.
      - *System* implies management of all technological components, people, data, hardware, organization, training, motivation, and programs.
      - *Engineering*: means systematic and quantified, 'real' engineering processes, where proactive design is used to manage system performance (incl. all qualities) attributes and costs.

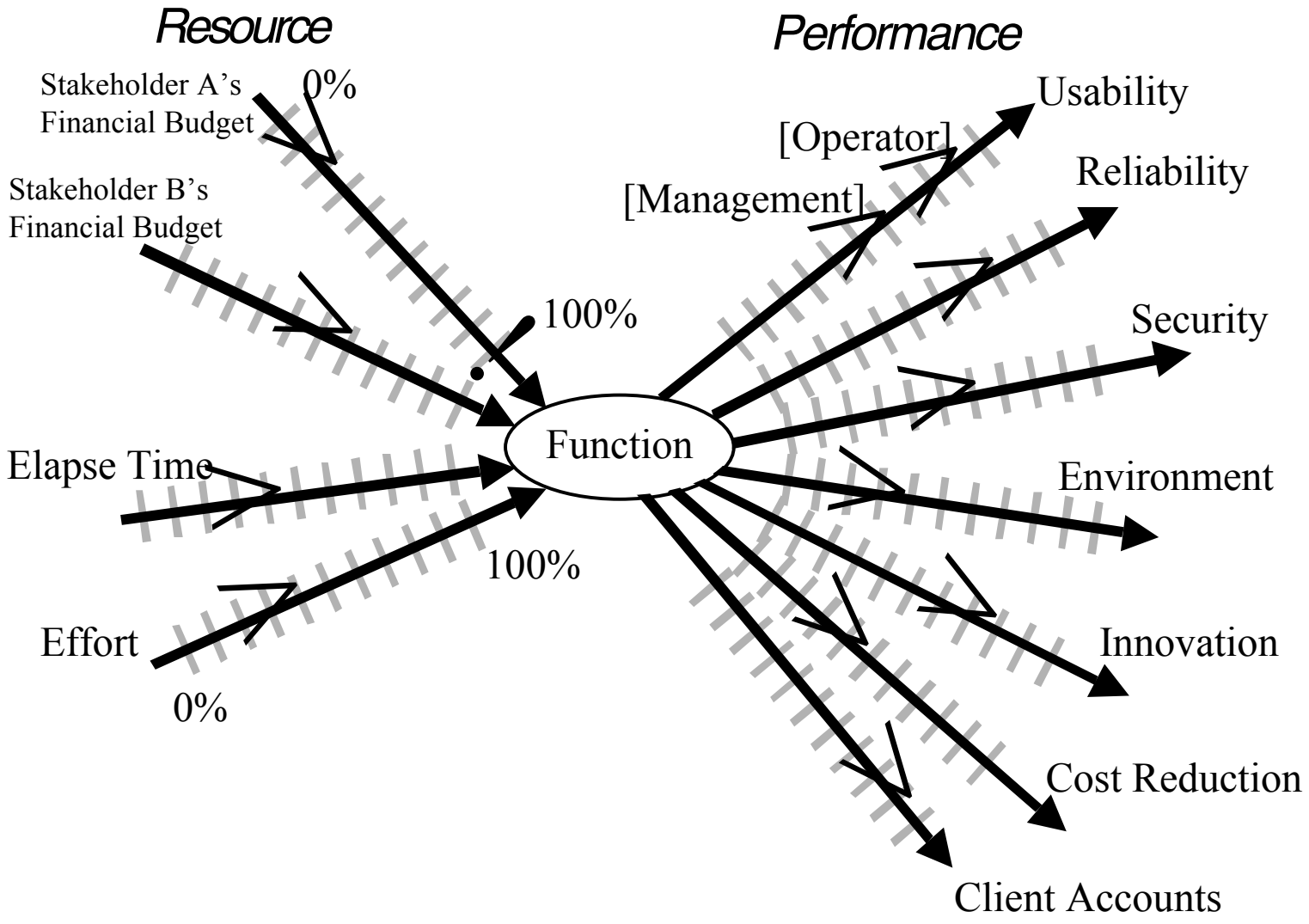
1. COMPLETE REQUIREMENTS:

- The RE (Requirements Engineer) is responsible for **absolutely all requirements specification** that the system must be aware of, and be responsible for to all critical or relevant stakeholders.
  - In particular, the RE is
    - not narrowly responsible for requirements from users and customers alone.
    - They are responsible for all other stakeholders,
      - » such as operations, maintenance, laws, regulations, resource providers, and more.

Rich and Complete Requirement Concepts

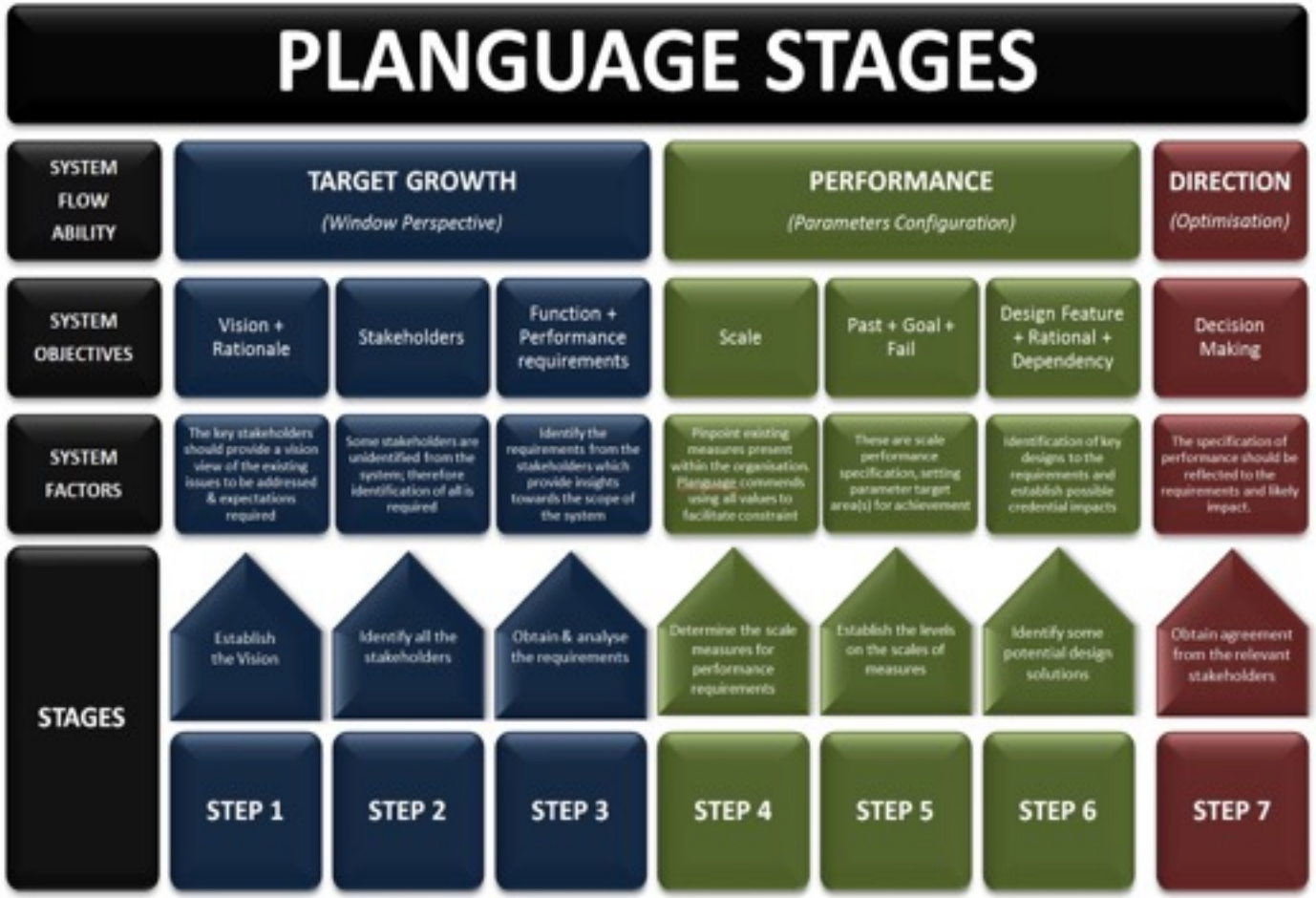


Multiple Required Performance and Cost Attributes
are the basis for architecture selection and evaluation



Planguage stages

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



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

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

9

8

3

3

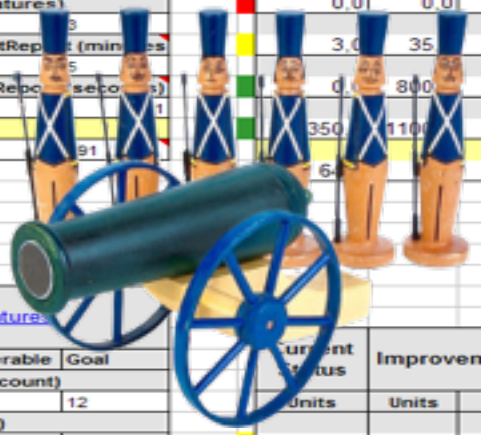
Impact Estimation Table: Reportal codename "Hyggen"

Current Status	Improvements		Reportal - E-SAT features		
Units	Units	%	Past	Tolerable	Goal
			Usability.Intuitivness (%)		
75,0	25,0	62,5	50	75	90
			Usability.Consistency.Visual (Elements)		
14,0	14,0	100,0	0	11	14
			Usability.Consistency.Interaction (Components)		
15,0	15,0	107,1	0	11	14
			Usability.Productivity (minutes)		
5,0	75,0	96,2	80	5	2
5,0	45,0	95,7	50	5	1
			Usability.Flexibility.OfflineReport.ExportFormats		
3,0	2,0	66,7	1	3	4
			Usability.Robustness (errors)		
1,0	22,0	95,7	7	1	0
			Usability.Replacability (nr of features)		
4,0	5,0	100,0	8	5	3
			Usability.ResponseTime.ExportRep (min		es
1,0	12,0	150,0	13	13	5
			Usability.ResponseTime.ViewRepo	seco	s)
1,0	14,0	100,0	15		1
			Development resources		
203,0			0		91

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

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

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



Real Bank Project : Project Progress Testability
Quantification of the most-critical project objectives on day 1

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

Speed-To-Deliver: **Scale**: average Calendar days needed from New Idea Approved until Idea Operational, for given Tasks, on given Markets.
Past [2009, Market = EURex, Task =Bond Execution] **2-3 months** ?
Goal [Deadline =End 20xz, Market = EURex, Task =Bond Execution] **5 days**

Operational-Control: **Scale**: % of trades per day, where the calculated economic difference between OUR CO and Marketplace/Clients, is less than "1 Yen"(or equivalent).
Past [April 20xx] **10%** change this to 90% NH **Goal** [Dec. 20xy] **100%**

Operational-Control.Consistent: **Scale**: % of defined [Trades] failing full STP across the transaction cycle. **Past** [April 20xx, Trades=Voice Trades] **95%**
Past [April 20xx, Trades=eTrades] **93%**
Goal [April 20xz, Trades=Voice Trades] <**95 ± 2%**>
Goal [April 20xz, Trades=eTrades] **98.5 ± 0.5 %**

Operational-Control.Timely.End&OvernightP&L Scale: number of times, per quarter, the P&L information is not delivered timely to the defined [Bach-Run].
Past [April 20xx, Batch-Run=Overnight] 1 **Goal** [Dec. 20xy, Batch-Run=Overnight] <**0.5**> **Past** [April 20xx, Batch-Run= **T+1**] **1 Goal** [Dec. 20xy, Batch-Run=End-Of-Day, Delay<1hour] **1**

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

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

Front-Office-Trade-Management-Efficiency Scale: Time from Ticket Launch to trade updating real-time risk view
Past [20xx, Function = Risk Mgt, Region = Global] **~ 80s** +/- **45s** ??
Goal [End 20xz, Function = Risk Mgt, Region = Global] **~ 50%** better?
Managing Risk – Accurate – Consolidated – Real Time

Risk.Cross-Product Scale: % of financial products that risk metrics can be displayed in a single position blotter in a way appropriate for the trader (i.e. – around a benchmark vs. across the curve).
Past [April 20xx] **0%** 95%. **Goal** [Dec. 20xy] **100%**

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

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

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

Cost-Per-Trade Scale: % reduction in Cost-Per-Trade
Goal (EOY 20xy, cost type = I 1 – REGION = ALL) **Reduce cost by 60%** (BW)
Goal (EOY 20xy, cost type = I 2 – REGION = ALL) **Reduce cost by x %**
Goal (EOY 20xy, cost type = E1 – REGION = ALL) **Reduce cost by x %**
Goal (EOY 20xy, cost type = E 2 – REGION = ALL) **Reduce cost by 100%**
Goal (EOY 20xy, cost type = E 3 – REGION = ALL) **Reduce cost by x %**

Detailed Example

- **Operational-Control.Consistent :**
  - **Scale: % of defined [Trades] failing full STP across the transaction cycle.**
  - **Past [April 20xx, Trades=Voice Trades] 95%**
    Past [April 20xx, Trades=eTrades] 93%
  - **Goal [April 20xz, Trades=Voice Trades] <95 ± 2%>**
    Goal [April 20xz, Trades=eTrades] 98.5 ± 0.5 %

Impacts On …
The Requirements in Planguage

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

	Impacts [Functions]	Impacts [Intended Performance Requirements]	Impacts [Intended Scale]	Impact Past	Impact Tolerable	Impact Goal
I1	Enter Content [Consumer] request details	Efficiency. Effort Saving. Reduce Time for [User] to produce request	Average time taken for define [request type: default=user]	[<2012, HH, User, 180 minutes]	30 minutes	5 minutes
I2	Submit [Content] Request	Efficiency. Effort Saving. Reduce Time for [User] to enter request	Average time taken for define [request type: default=user]	[<2012, HH, User, 30 minutes]	15 minutes	10 minutes
I3	Process a [User] Request	Efficiency. Elapse Time Saving. Reduce [TIME] to process user request	Average time taken for define [request type: default =processor]	[<2012, HH, User, 70 minutes]	30 minutes	15 minutes
I4	Usability [Sheet] Type	Average Number of [Sheet] Completed Manually Monthly	1412 sheets	[<2012, HH, Completed Sheets, 1412]	1000 lines	850 lines
I5	Usability. Reduce number of Content [Errors]	Average Number [Errors] of Content	353 errors per week	[<2012, HH, User, 353 per week]	100 per week	30 per week
I6	Update [Process] rules	Efficiency. Elapse Time Saving. Reduce [TIME] to update the rules	Average time taken for [Content Validation]	[<2012, HH, Verifier, 50 minutes]	35 minutes	20 minutes
I7	Distribution [Location]	Accessibility. Elapse Time Saving. Increase the information flow distribution	Number of sheets distributed	[<2012, HH, Send Information [Physical] location]	20 wards	Anywhere
I8	Distribution [Accessibility]	Accessibility. Elapse Time Access	System access volume	[<2012, HH, Open Time, 9am – 5pm]	9am – 12pm	Anytime
I9	Notification [Query Calls]	Notification. Elapse Change Over [Query Calls]	[Decrease the number of query calls]	[<2012, Calls Measure, 85% Volume]	40%	10%
I10	Update [Connect Content] Rules	Efficiency. Elapse Time Saving. Reduce [Time] taken to produce label	Average [time] taken	[<2012, HH, Producer, Processing, 10 minutes]	6 minutes	2 minutes
I11	Time. Costing to [Retrieve]	Cost. Cost Saving. Reduce cost in retrieval of information	Average [time] taken	[<2012, HH, User, 240 minutes searching time]	60 minutes	15 minutes
I12	Time [File]	Efficiency. Efficiency Saving. Reduce time taken to file	Average [time] taken	[<2012, HH, Administrator, 30 minutes]	15 minutes	3 minutes
I13	Time [Learn]	Learn ability. Elapse Time Learning. Reduce Time on Training	Average time taken for [request type: default=user] to learn process	[<2012, HH, Learner, 1 day]	4 hours	1 hour

2. **QUALITY REQUIREMENTS:**

- The RE is **responsible for the quality level**, *in relation to official standards*, of all requirements they transmit to others.
  - They are consequently responsible for making sure the quality of incoming raw requirements, needs, values, constraints etc. is good enough to process. No GIGO.
  - If input is not good quality,
    - they are responsible for making sure it is better quality,
    - or at least clearly annotated where there is
      - » doubt, incompleteness, ambiguity and any other potential problems, they cannot resolve yet.

Source Eric Simmons, erik.simmons@intel.com 25 Oct 2011
Personal Public Communication

A Recent Example

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

Rev.	# of Defects	# of Pages	Defects/ Page (DPP)	% Change in DPP
0.3	312	31	10.06	
0.5	209	44	4.75	-53%
0.6	247	60	4.12	-13%
0.7	114	33	3.45	-16%
0.8	45	38	1.18	-66%
1.0	10	45	0.22	-81%
Overall % change in DPP revision 0.3 to 1.0:				-98%

Downstream benefits:
- Scope delivered at the Alpha milestone increased 300%, released scope up 233%
- SW defects reduced by ~50%
- Defects that did occur were resolved in far less time on average

3. ARCHITECTURE:

- The Requirements Engineer is NOT responsible for any architecture or design process itself.
  - This will be done by professional engineers and architects.
- They are however very much responsible for a *complete* and *intelligible quality* set of requirements,
  - transmitted to the designers and architects.
- The are also responsible for transmitting quality-controlled architecture or design specifications to any relevant system builders.
  - These are the designs which are input requirements to builders. Effectively they are ‘design constraints requirements’.

Impact Estimation Elements

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

Architecture Specification Rules

from CE Book Ch. 7

7.4 Rules: Design Specification

R8: IE table:

The set of design ideas specified to meet a set of requirements should be validated at an early stage by using an Impact Estimation (IE) table.

Acer Project: Impact Estimation Table

Strategies

Objectives

Impacts

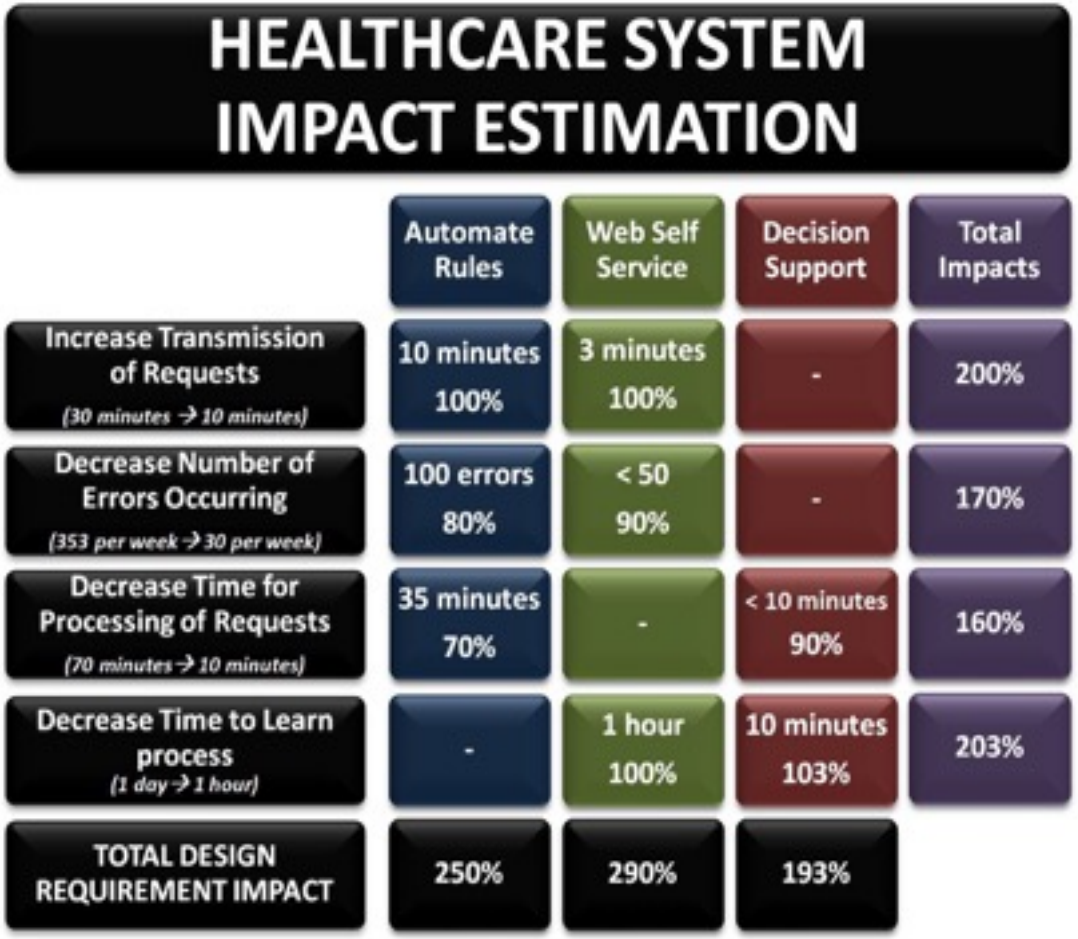
Strategies / Goals	Identify Binding Compliance Requirements Strategy	System Control Strategy	System Implementation Strategy	Find Services That Meet Our Goals Strategy	Use The Lowest Cost Provider Strategy
Security Administration Compliance 25% → 90%	100%	100%	100%	50%	0%
Security Administration Performance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

Impact Estimation: Value-for-Money Delivery Table

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**	

Healthcare Impact Estimation

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



VALUE Decision Tables: Multiple Levels

Product - Solution - VKoT			swipe payments		economic overview		Netbank ajax		Netbank server		payment.tonon		search.contexta			
				213%		208%		171%		175%		367%		194%		0%
				52%		25%		38%		31%		-37%		123%		
				123%		119%		50%		9%		59%		99%		
Value Requirements			units	% of Goal	units	% of Goal	units	% of Goal	units	% of Goal	units	% of Goal	units	% of Goal	units	% of Goal
Snappiness			**10**	**71%**	**-5**	**-36%**	**10**	**71%**	**12**	**86%**	**-1**	**-7%**	**14**	**100%**		
85	90	99	5	36%	2	14%	5	36%	3	21%	5	36%	10	71%		
5-Dec-13	5-Jun-14	5-Jun-14	0.1	7%	0.3	-11%	0.7	50%	0.1	9%	0.1	-1%	0.5	50%		
Reliability			**10**	**11%**	**30**	**33%**	**90**	**100%**	**80**	**89%**	**-1**	**-1%**	**-5**	**-6%**		
30	60	120	1	1%	7	8%	2	2%	9	10%	2	2%	1	1%		
5-Dec-13	5-Jun-14	5-Jun-14	0.4	4%	0.8	27%					0.7	-1%	0.2	-1%		
Usability.Intuitiveness			**40**	**100%**	**80**	**200%**					**30**	**75%**	**40**	**100%**		
30	40	70	10	25%	5	13%					10	25%	20	50%		
5-Dec-13	5-Jun-14	5-Jun-14	0.9	90%	0.5	100%					0.8	60%	0.5	50%		
Productivity-Task			**-3**	**30%**	**-1**	**10%**					**-30**	**300%**				
30	25	20	1	-10%	1	-10%					10	-100%				
5-Dec-13	5-Jun-14	5-Jun-14	0.7	21%	0.3	3%										
PV5																
1	2	3														

4. Priority Information:

- The Requirements Engineer is NOT responsible for *prioritization* of requirements.
- Prioritization is done dynamically
  - at the project management (PM) level,
  - based on prioritization signals in the requirements,
  - and on current feedback and experience in the value delivery cycles (Sprints).
- The primary responsibility of the Requirements Engineer,
  - is to *systematically and thoroughly collect and disseminate* all relevant priority signals, into the requirement specification;
  - so that intelligent prioritization can be done at any relevant level, and at any time.

Risk Management

- **the Requirements Engineer is NOT responsible for Risk Management**
  - But **is** responsible for
    - making sure that all specifications follow guidelines
      - (Rules, Quality Levels) that demand information specified about, or related to, risks and their mitigations.

Design Spec Enlarged 2 of 2

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

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

A1: **FCCP is assumed to** not currently exist and i Requirements Spec. <- discussions AH MA JH EC.

Consequence: FCx impact estimation

A2: **Costs**, the developm different. All will base o and 3 years. The ops cos mm for hardware. MA AH

A3:Boss X will continue t

A4: the schedule, 3 year we can in fact deliver, O budget. If not "I would h

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

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

Dependencies: *<State a*

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

ASSUMPTIONS:
• broadcasts critical factors for present and future re-examination
• helps risk analysis
• are an integral part of the design specifiction

DEPENDENCIES:

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

R1. FCxx is delaye tsg 2.12

R2: the technical thought & we mus

R3: the and or sca allow us to meet t

R4: **scalability** of year especially <-

R5: re Cross Desk on technical desig no solution allowi

Risks specification:
• shares group risk knowhow
• permits redesign to mitigate the risk
• allows relistic estimates of cost and impacts

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

I1: Do we need to put t the objectives (Owners a huge differentiator. D

I2: what are the time sc now BB

I3: what will the succes what we are actually be

I4: for the business othe a lack of clarity as to w how they might differ f

I5: the degree to which useful without Intra Day. BB 2 dec

Issues:
• when answered can turn into a risk
• shares group knowledge
• makes sure we don't forget to analyze later

Product:

- The system that delivers the primary critical values to stakeholders. (Tsg 7 dec 2013)

Product Owner:

- The instance (person or team) responsible for Effective Communication between all stakeholders, and any technical project, both development and maintenance. (Tsg 7 Dec 2013)

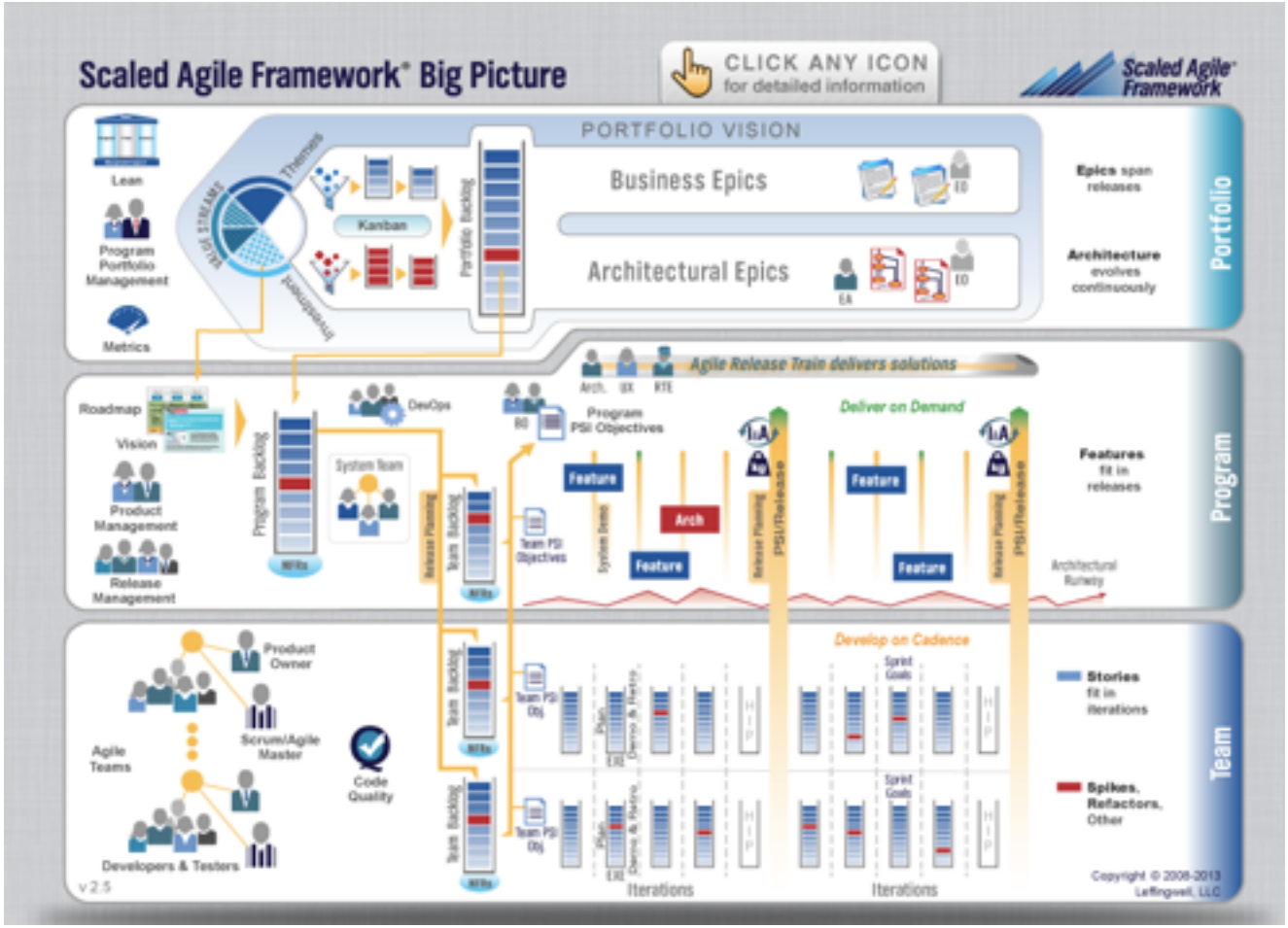
Effective Communication:

- Two-way communication, between *all* related instances in technical projects, is effective when:
  - 1. Communication is rapid: first try
  - 2. Communication meets relevant standards (Rules,) including these basic rules.
    - Clear enough to *test*
    - Unambiguous to *intended readership*
    - Critical variables (esp. qualities) *quantified*
    - Clear *distinction* between ends and means
  - 3. Communication is 'relevant'.
    - What stakeholders *really* want
      - » NOT perceived means to their true ends
    - What developers really need to know

Priority Signals

- When Due
- Higher level requirements
- Stakeholders
- Under which conditions
- Constraints
- Residual resources (running out of time, money etc)

What About scaledagileframework.com ?



Epic value Statement Format

Forward-Looking Position Statement	
For	<customers>
who	<do something>
the	<solution>
is a	<something – the "how">
that	<provides this value>
Unlike	<competitor, current solution, or non-existing solution>
our solution	<does something better – the "why">
Scope	
Success Criteria:	▸ ▸
In Scope:	▸ ▸
Out of Scope:	▸ ▸
NFRs:	▸ ▸

Figure 1. Epic value statement template format

Epic Lightweight Business Case

Epic Name	Go or NO Go Recommendation:	Date entered Backlog:	Analyst Epic Owner:
Version		Changes	
Description of the Epic			
Success Criteria	• •		
Stakeholders sponsors	(Identifie		
Users and markets affecte			
Products, programs, servic affected			
Impact on sales, distributio deployment			

Estimated investment	Story points:		Cost:
Weighted rating	(WSJF)	Type of return	(Nature of potential return. Revenue, market share, new markets served)
In house or outsource development	(describes recommendations for where the epic is to be developed)		
Estimated development timeline	Start Date:		Completion date: (Estimated calendar date or number of PSIs)
Incremental Implementation Strategy	(Breaks initiative down into preliminary epics or sub-epics that fit the companies PSI cadence)		
Reevaluation checkpoints	(If the epic is large, identifies potential milestones or checkpoints for reevaluation)		
Analysis summary	(Brief summary of the analysis that has been formed to create the business case. Pointers to other data, feasibility studies, models, market analysis, etc. that was used on the creation of the business case)		
Attachments	Project Stakeholder Needs Assessment (see Chapter 7) System Stakeholder Needs Assessment		
Other notes and comments			

Figure 1. Epic Lightweight Business Case

1/2

Epic Name	**Go or NO Go Recommendation:**	**Date entered Backlog:**	**Analyst Epic Owner:**
Version		Changes	
Description of the Epic			
Success Criteria	• •		
Stakeholders sponsors	(Identifies key business sponsors who will be supporting the initiative)		
Users and markets affected	(Describe the user community of the solution and any markets affected)		
Products, programs, services affected	(Identifies products, programs, services, teams, departments, etc. that will be impacted by this epics)		
Impact on sales, distribution, deployment	(Describes any impact on how the product is sold, distributed, or deployed)		
Estimated investment	**Story points:**	**Cost:**	
Weighted rating	(WSJF)	**Type of return**	(Nature of potential return. Revenue, market share, new markets served)
In house or outsource development	(describes recommendations for where the epic is to be developed)		
Estimated development timeline	**Start Date:**	**Completion date:** (Estimated calendar date or number of PSIs)	
Incremental implementation Strategy	(Breaks initiative down into preliminary epics or sub-epics that fit the companies PSI cadence)		
Reevaluation checkpoints	(If the epic is large, identifies potential milestones or checkpoints for reevaluation)		
Analysis summary	(Brief summary of the analysis that has been formed to create the business case. Pointers to other data, feasibility studies, models, market analysis, etc. that was used on the creation of the business case)		

1/2

Epic Name	Go or NO Go Recommendation:	Date entered Backlog:	Analyst Epic Owner:
Version		Changes	
Description of the Epic			
Success Criteria	• •		
Stakeholders sponsors	(Identifies key business sponsors who will be supporting the initiative)		
Users and markets affected	(Describe the user community of the solution and any markets affected)		
Products, programs, services affected	(Identifies products, programs, services, teams, departments, etc. that will be impacted by this epics)		
Impact on sales, distribution, deployment	(Describes any impact on how the product is sold, distributed, or deployed)		

Estimated development timeline	**Start Date:**	**Completion date:** (Estimated calendar date or number of PSIs)
Incremental Implementation Strategy	(Breaks initiative down into preliminary epics or sub-epics that fit the companies PSI cadence)	
Reevaluation checkpoints	(If the epic is large, identifies potential milestones or checkpoints for reevaluation)	
Analysis summary	(Brief summary of the analysis that has been formed to create the business case. Pointers to other data, feasibility studies, models, market analysis, etc. that was used on the creation of the business case)	

Attachments	Project Stakeholder Needs Assessment (see Chapter 7) System Stakeholder Needs Assessment
Other notes and comments	

Initial Take

- Is moving in the direction of Planguage for specification
- But, does not go near the concepts of managing value by means of quantified value and quality directly
- Does not understand dynamic prioritization via values and costs (see the weighting scheme)