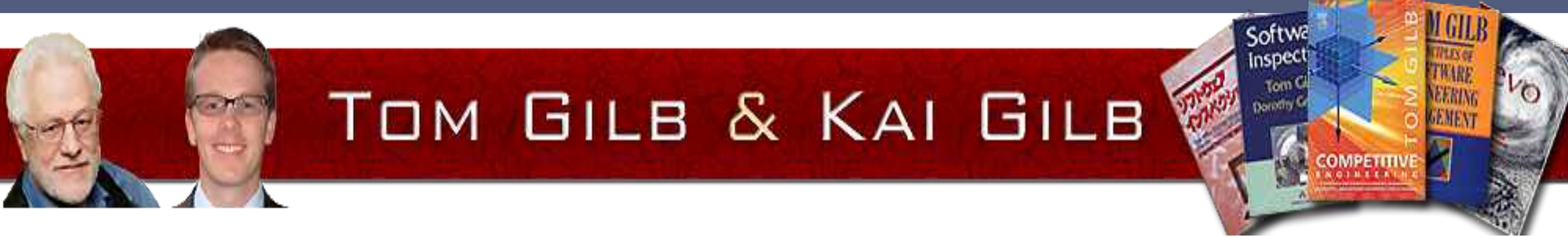


Enabling Quality, through tools and technology: ‘Lean QA’

London SPIN, 17 November
25 minutes + 5 Q&A

by Gilb

Slide owner: Kai@Gilb.com @kaigilb
Tom@Gilb.com @imtomgilb
www.gilb.com

These slides will be at:
http://www.gilb.com/tiki-download_file.php?fileId=437

Main Take-away Points

Quality Assurance is far more than ‘test’,
    and it can be far more cost-effective

‘Quality’ is far more than ‘bugs’

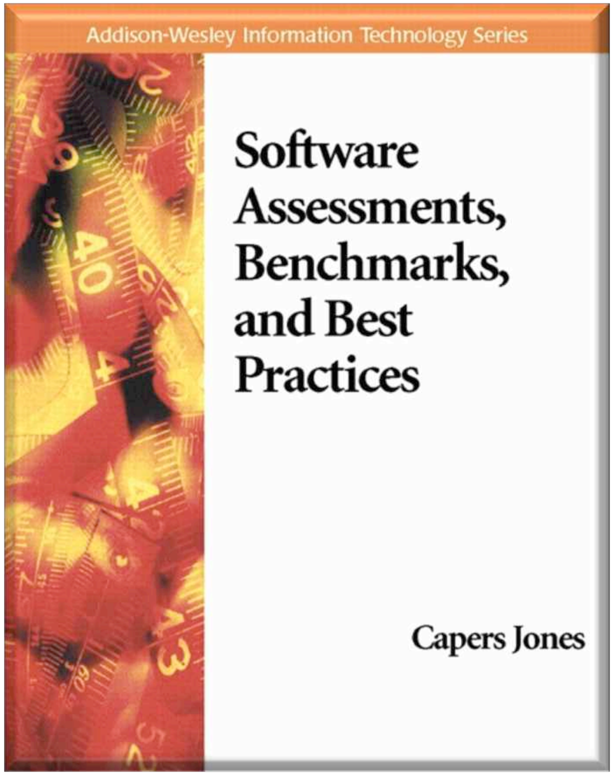
You probably have a lot to learn,
    if you want real competitive quality

Begin:
Quality Assurance is far more than ‘test’

and it can be far more cost-effective

Inspection Effectiveness

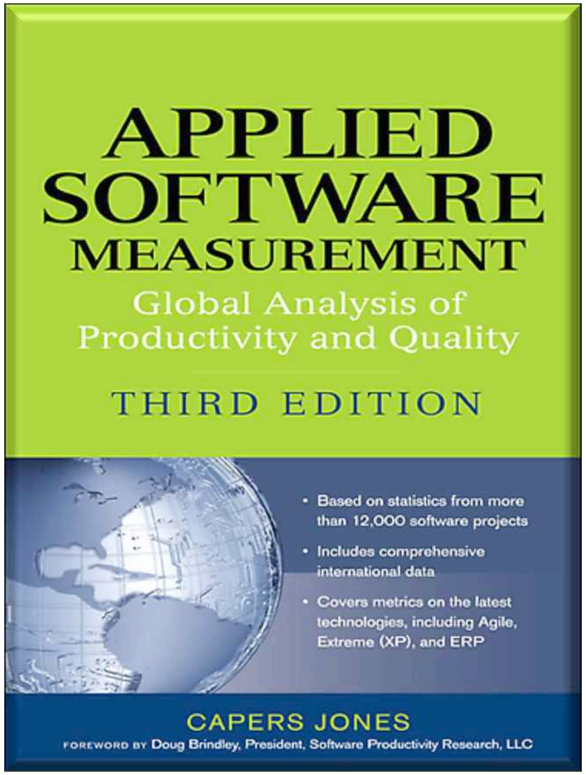
Capers Jones



Addison-Wesley Information Technology Series

Software Assessments, Benchmarks, and Best Practices

Capers Jones



APPLIED SOFTWARE MEASUREMENT

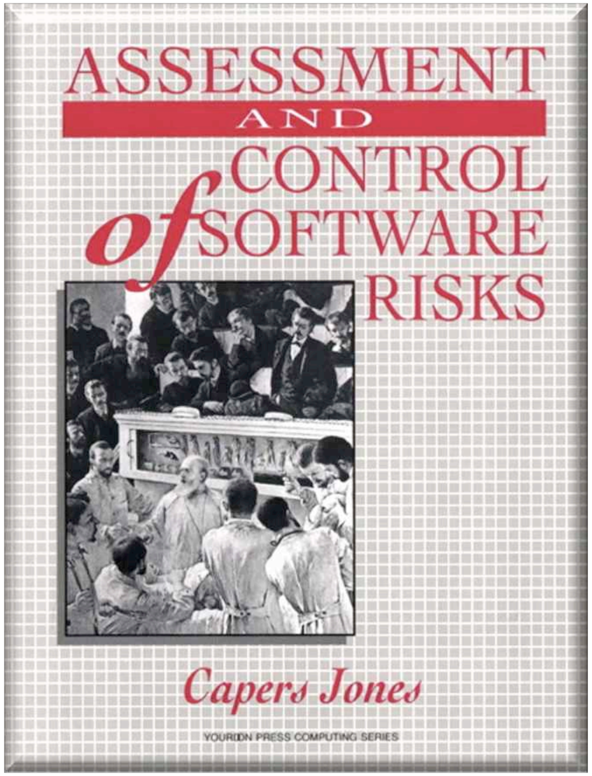
Global Analysis of Productivity and Quality

THIRD EDITION

- Based on statistics from more than 12,000 software projects
- Includes comprehensive international data
- Covers metrics on the latest technologies, including Agile, Extreme (XP), and ERP

CAPERS JONES

FOREWORD BY Doug Brindley, President, Software Productivity Research, LLC



ASSESSMENT AND CONTROL of SOFTWARE RISKS

Capers Jones

YOURDON PRESS COMPUTING SERIES

Regression test ?

15% to 30%

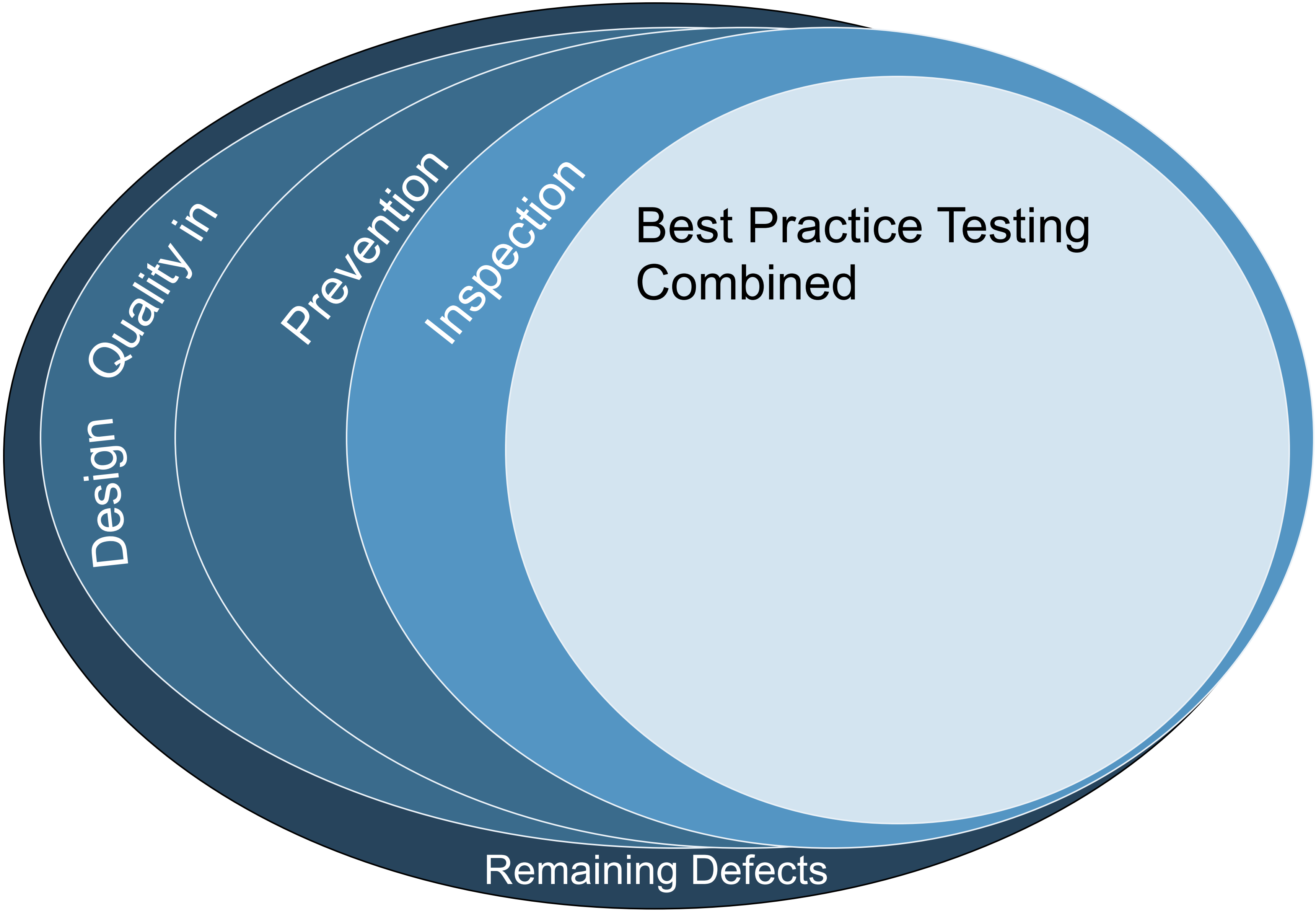
Integration test ?

25% to 40%

Unit test	15% to 50%
New function test	20% to 35%
Performance test	20% to 40%
System test	25% to 55%
Acceptance test (1 client)	25% to 35%
Low-volume Beta test (< 10 clients)	25% to 40%
High-volume Beta test (> 1000 clients)	60% to 85%

Inspections?

Informal design reviews	25% to 40%
Formal design inspections	45% to 65%
Informal code reviews	20% to 35%
Formal code inspections	45% to 70%


Design Quality in
Prevention
Inspection
Best Practice Testing Combined
Remaining Defects

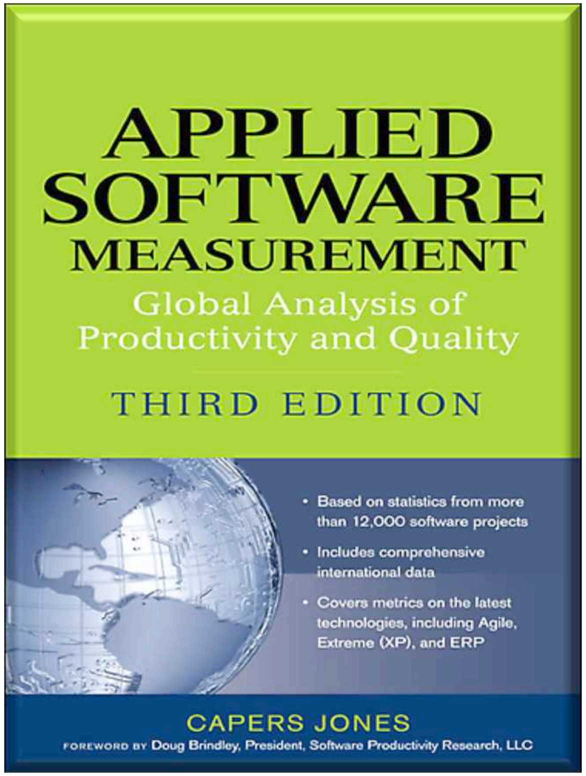
Little hope of ‘zero defects’

“Between

8 and **10**

defect removal stages required to achieve removal effectiveness of

95%”



APPLIED SOFTWARE MEASUREMENT

Global Analysis of Productivity and Quality

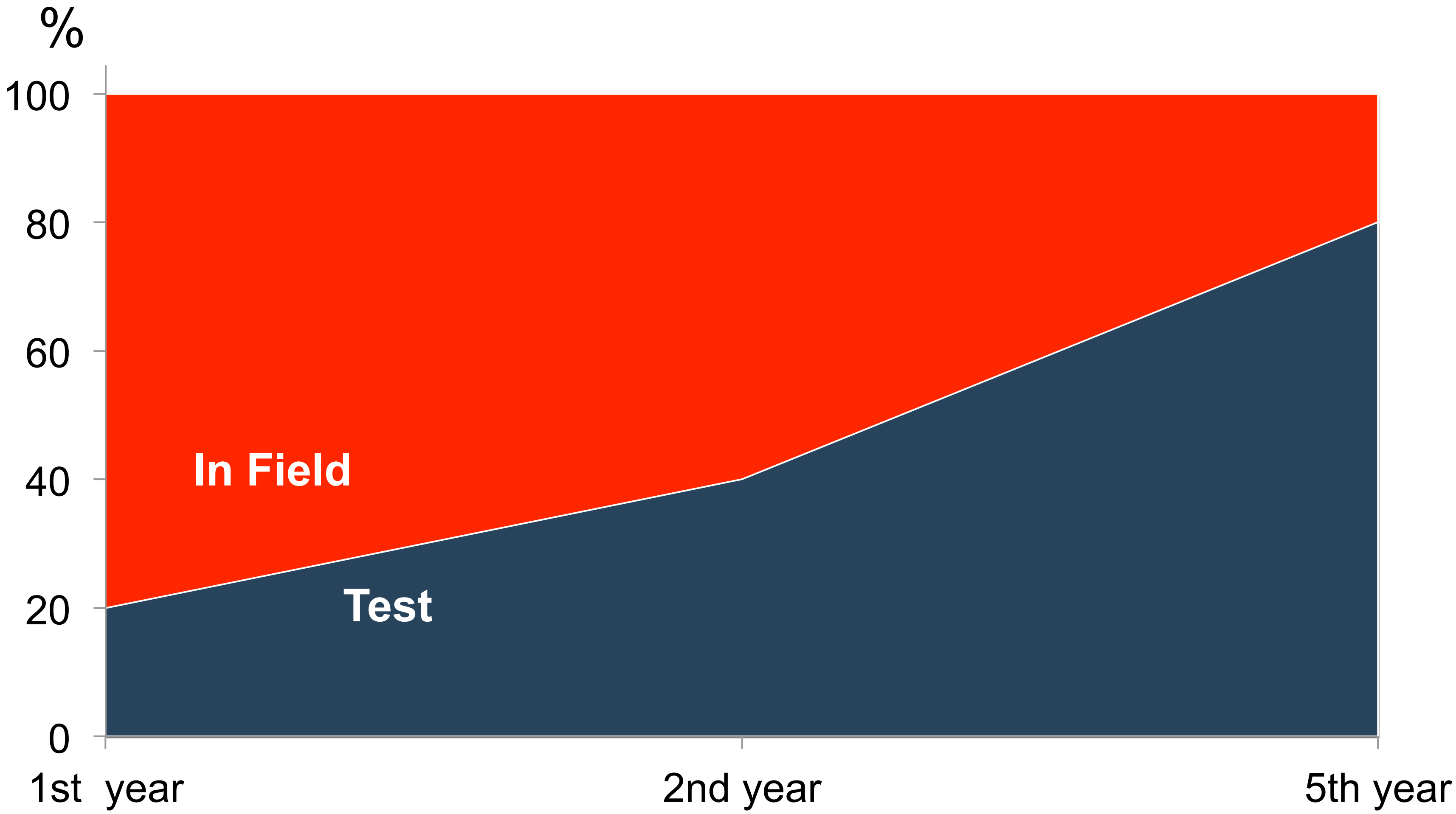
THIRD EDITION

- Based on statistics from more than 12,000 software projects
- Includes comprehensive international data
- Covers metrics on the latest technologies, including Agile, Extreme (XP), and ERP

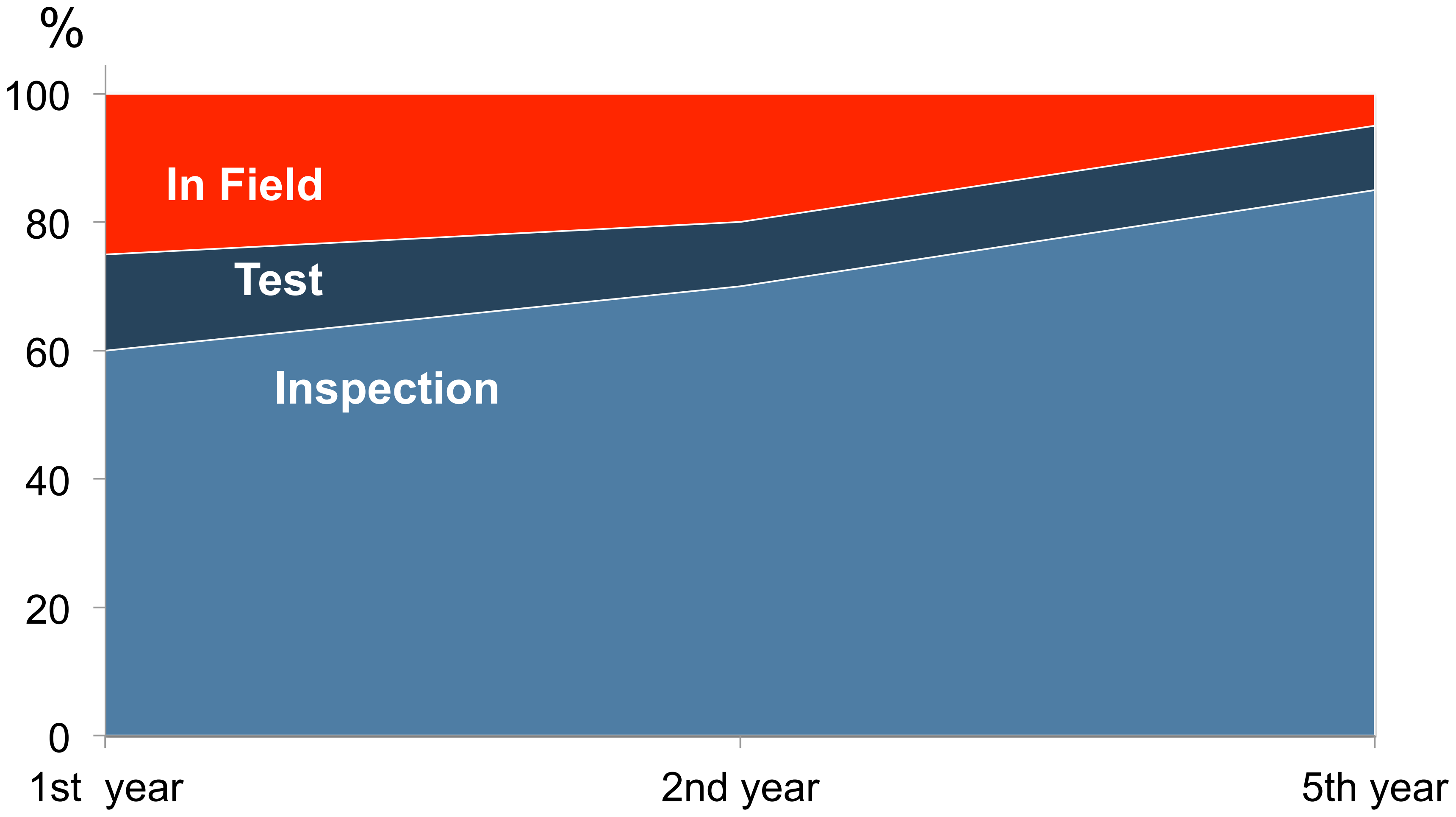
CAPERS JONES

FOREWORD BY Doug Brindley, President, Software Productivity Research, LLC

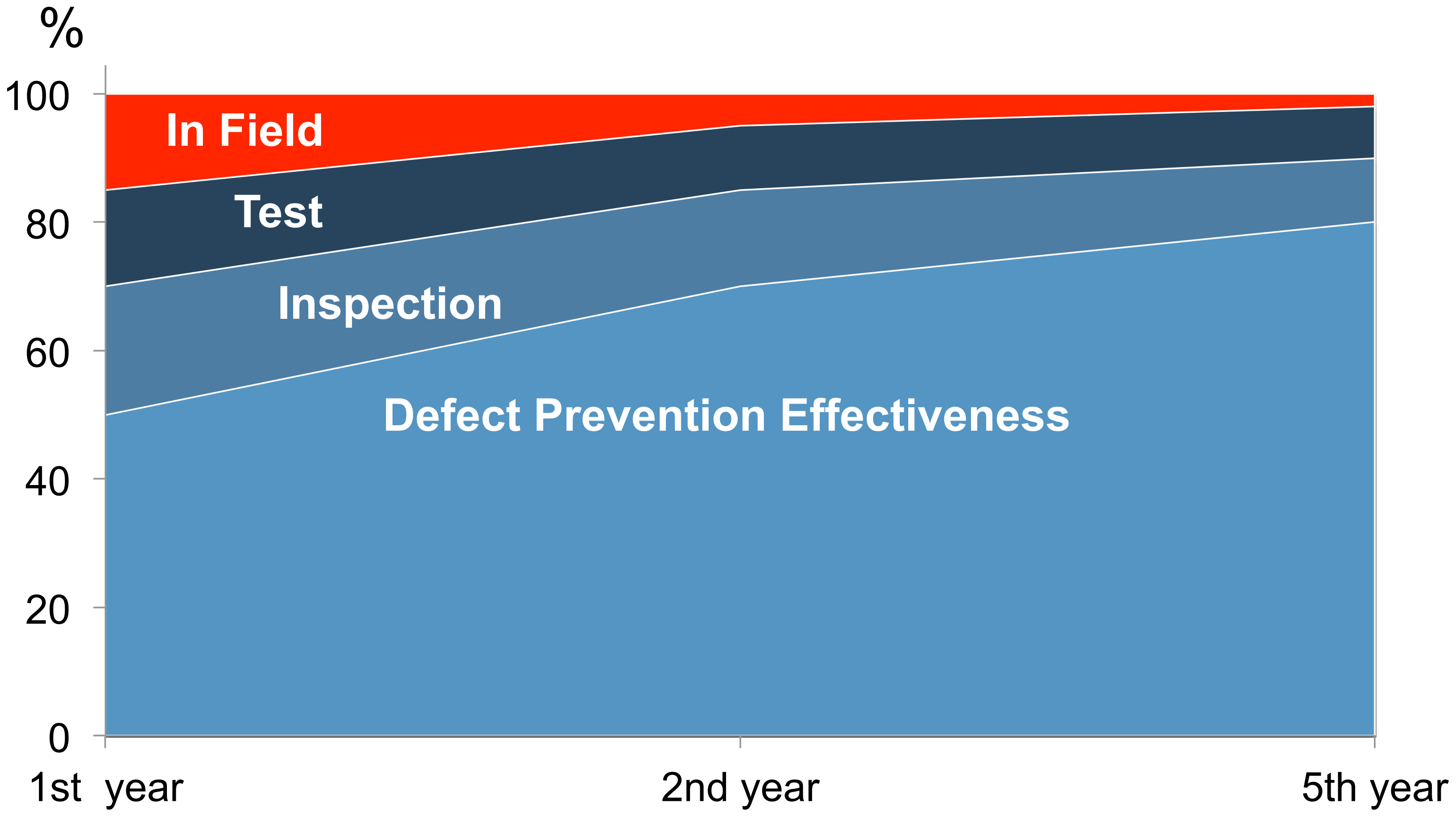
Testing Capability (C. Jones)



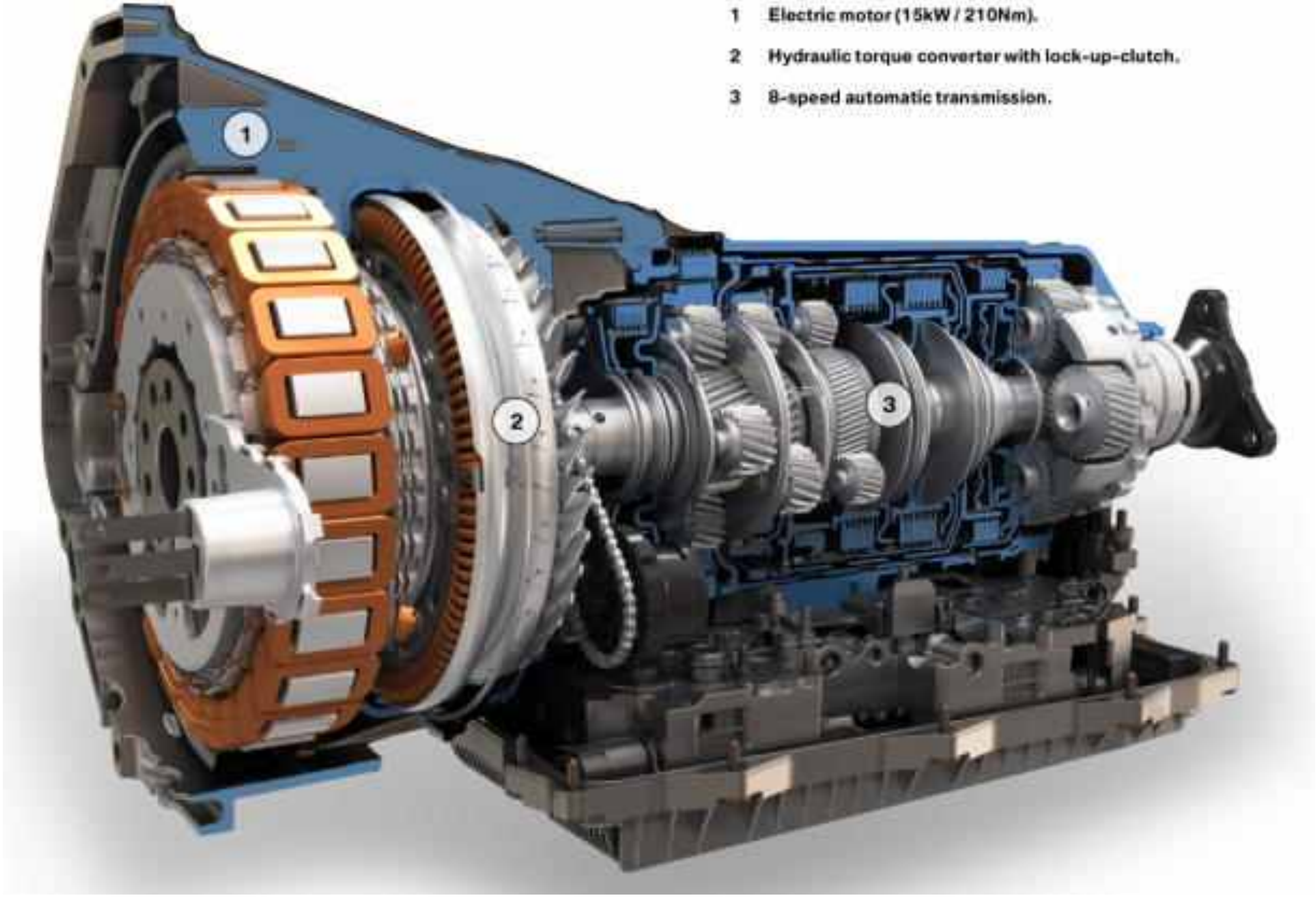
Defect Detection Capability (C. Jones)



IBM Defect Avoidance Experience



Design Quality In



1 Electric motor (15kW / 210Nm).

2 Hydraulic torque converter with lock-up-clutch.

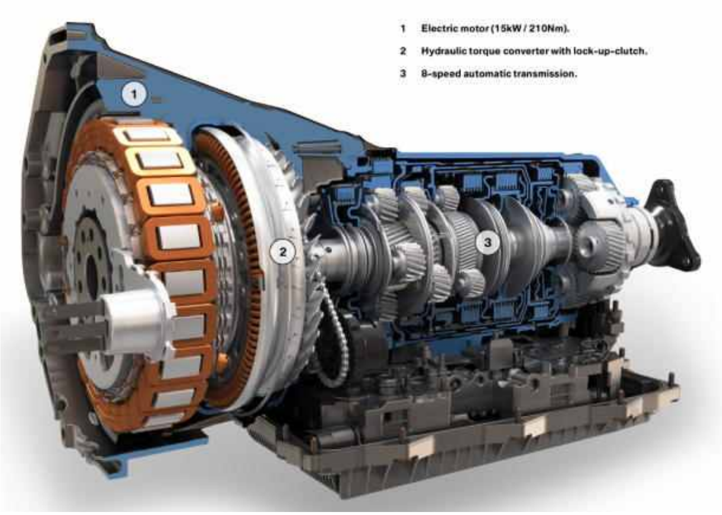
3 8-speed automatic transmission.

You don't get quality by testing it in

but by ‘Engineering’ Quality In

Work hours

$ € Kr.


1 Electric motor (15kW / 210Nm).
2 Hydraulic torque converter with lock-up-clutch.
3 8-speed automatic transmission.
1
2
3

Reliability

Performance

Security

Usability

Maintenance

Setting Quality Goals simple example

Usability.**Learn**

Scale: average time to Learn how to operate the computer, from .. to ..

Status [today] 3 hours

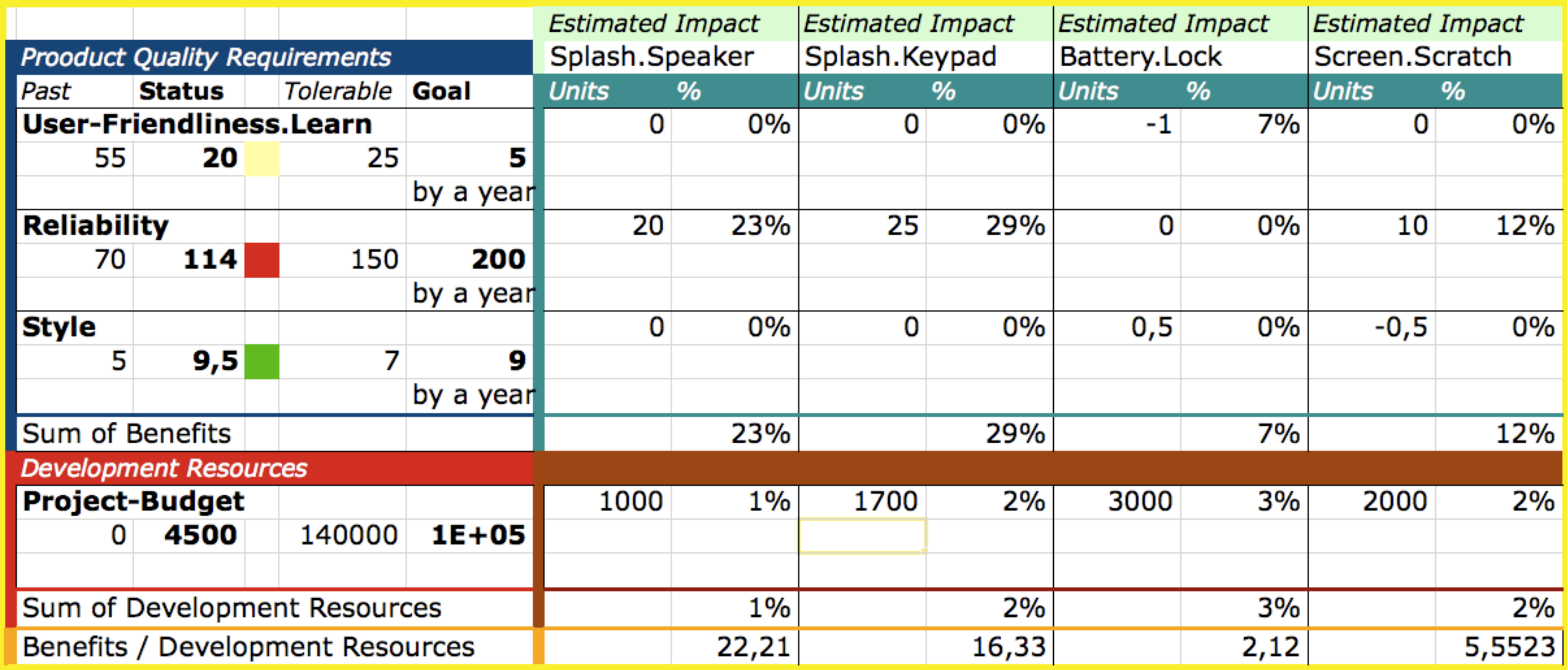
Goal [next year] 10 min.

Designing to meet Quality within Costs
A systematic Quantitative Method
Using 'Impact Estimation' Tables

Design Ideas

Qualities

€ $



					Estimated Impact		Estimated Impact		Estimated Impact		Estimated Impact	
Proodduct Quality Requirements					Splash.Speaker		Splash.Keypad		Battery.Lock		Screen.Scratch	
Past	**Status**		*Tolerable*	**Goal**	*Units*	*%*	*Units*	*%*	*Units*	*%*	*Units*	*%*
User-Friendliness.Learn					0	0%	0	0%	-1	7%	0	0%
55	**20**	🟨	25	**5**								
				by a year								
Reliability					20	23%	25	29%	0	0%	10	12%
70	**114**	🟥	150	**200**								
				by a year								
Style					0	0%	0	0%	0,5	0%	-0,5	0%
5	**9,5**	🟩	7	**9**								
				by a year								
Sum of Benefits						23%		29%		7%		12%
Development Resources												
Project-Budget					1000	1%	1700	2%	3000	3%	2000	2%
0	**4500**		140000	**1E+05**								
Sum of Development Resources						1%		2%		3%		2%
Benefits / Development Resources						22,21		16,33		2,12		5,5523

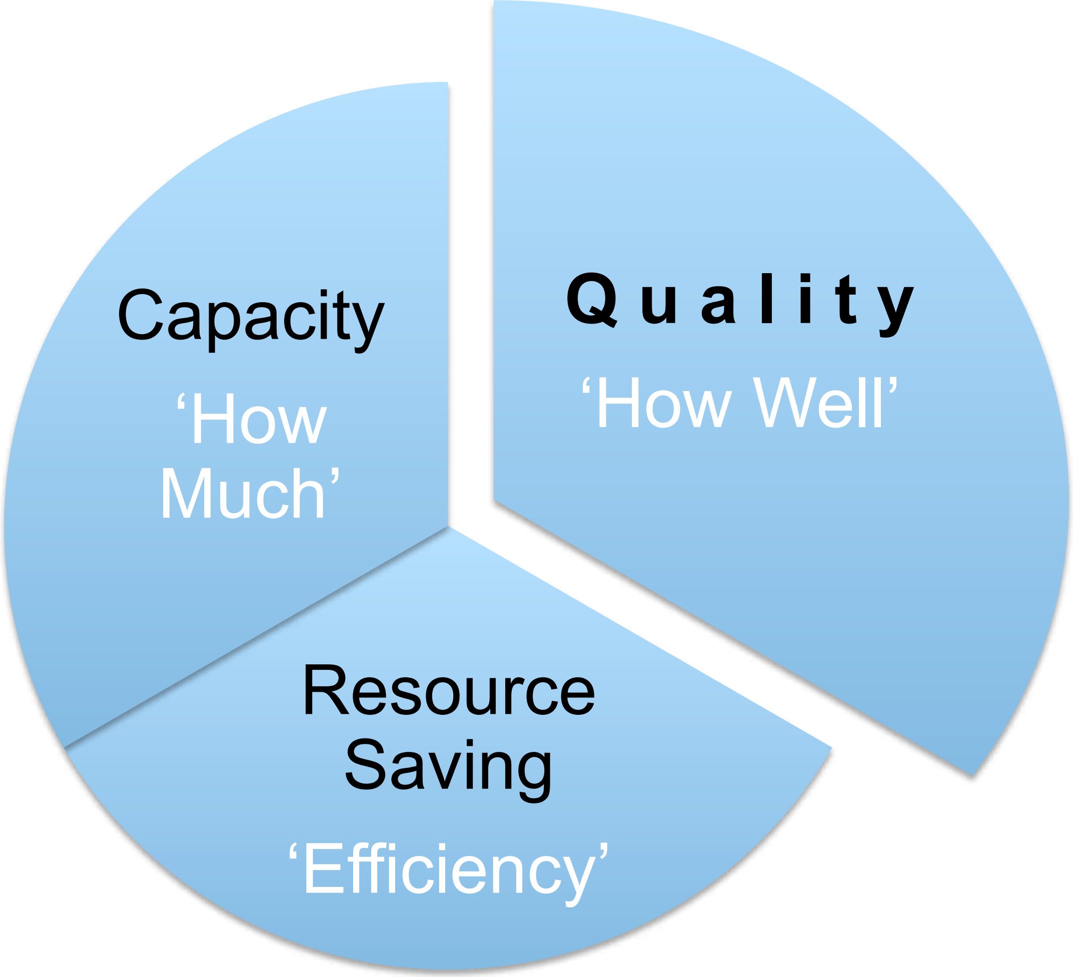
Quality Assurance
is far more than ‘test’

and, QA can be far more cost-effective
Than ‘test’ approaches

Cost-Effective = Quality Delivered / Cost

Quality is far more than ‘bugs’

System Performance



Capacity

'How Much'

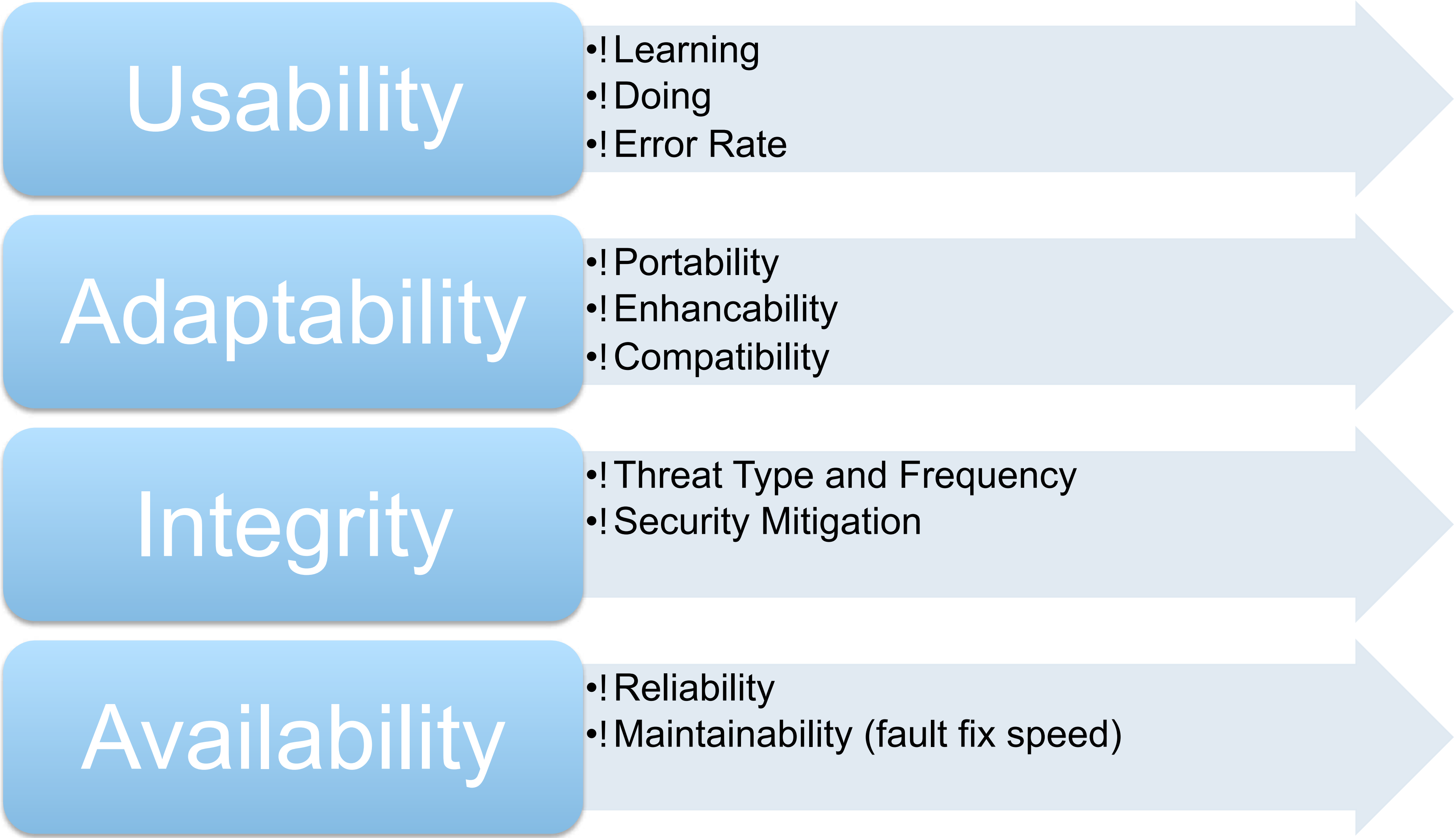
Quality

'How Well'

Resource Saving

'Efficiency'

Qualities are many and variable



Usability

•! Learning
•! Doing
•! Error Rate

Adaptability

•! Portability
•! Enhancability
•! Compatibility

Integrity

•! Threat Type and Frequency
•! Security Mitigation

Availability

•! Reliability
•! Maintainability (fault fix speed)

http://www.gilb.com/tiki-download_file.php?fileId=26

www.Gilb.com/Downloads/Esstentials/Ce Ch5

Quantify the Quality to ‘Assure’ It

“…I often say that

when you can **measure**
what you are speaking about,

and **express it in numbers**,

you know something about it;

but when you **cannot measure** it,

when you **cannot express it in numbers**,

your knowledge is of a meagre and unsatisfactory kind;…”

- Lord Kelvin, 1893

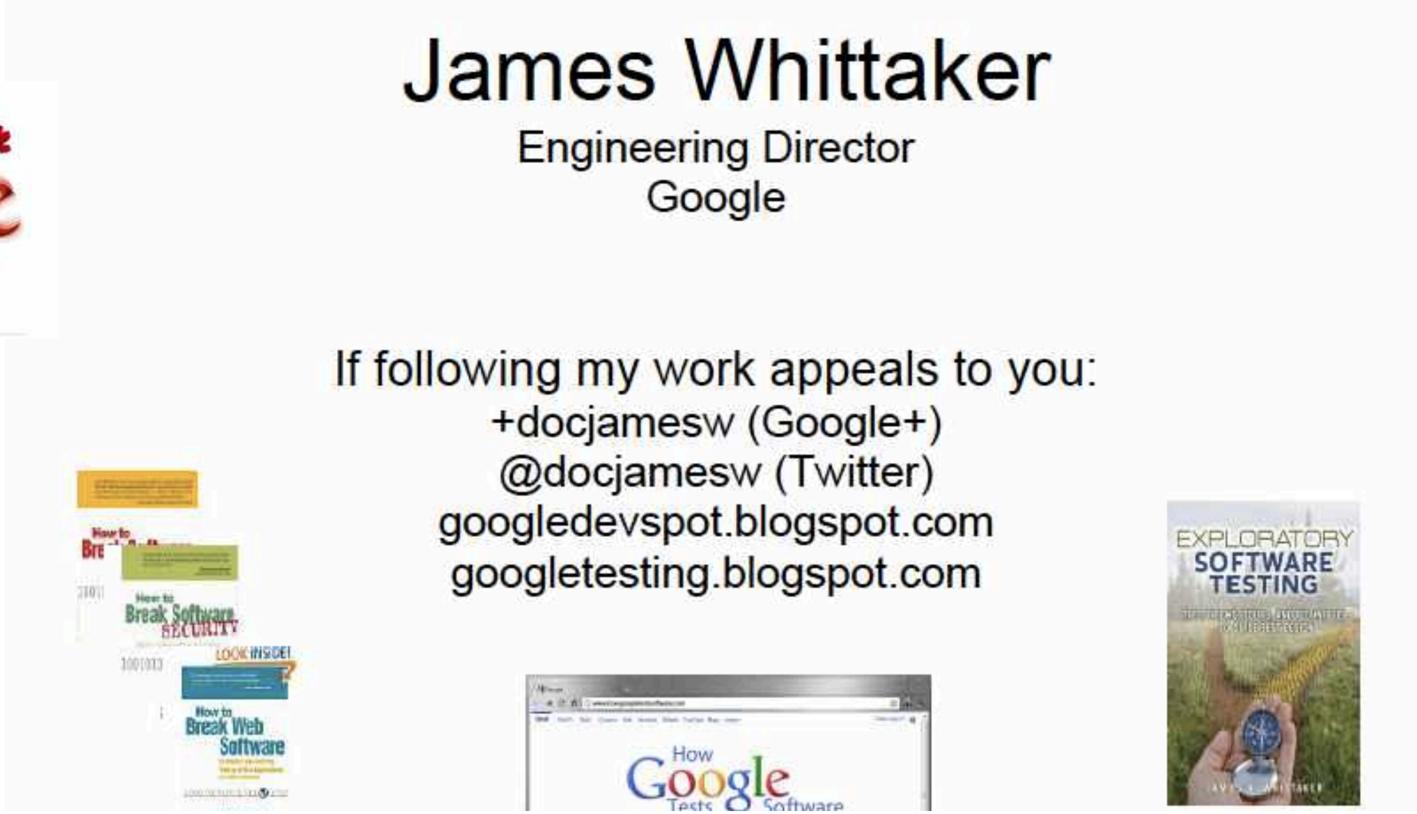
Main Idea, again

- There are many much smarter ways to get quality than ‘testing it in’

- For example, at Google ..

Google, is now experimenting in real Google projects. No Professional Testers

He has **totally eliminated** the use of **professional testers** on his team, replacing them with a set of *more cost effective means* for 'testing' the software.. (Construx Summit Talk, Oct 2011, Seattle)



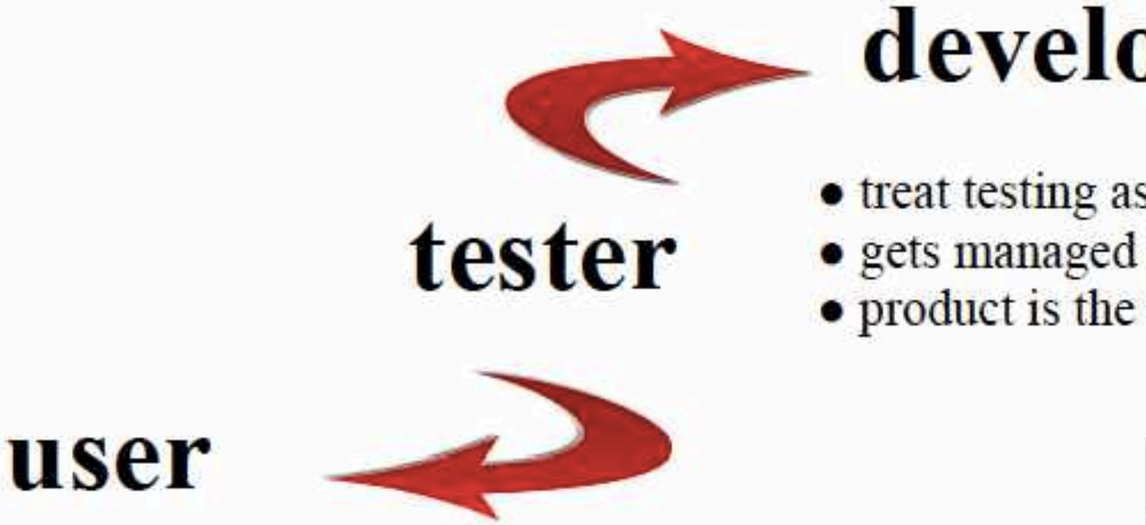
James Whittaker

Engineering Director
Google

If following my work appeals to you:
+docjamesw (Google+)
@docjamesw (Twitter)
googledevspot.blogspot.com
googletesting.blogspot.com

How to Break Software
How to Break Software Security
LOOK INSIDE!
How to Break Web Software
How Google Tests Software
Exploratory Software Testing

Google/Whittaker Summary 2011 “Where does testing fit in this world” JW



developer

- treat testing as a feature
- gets managed in dev workflow
- product is the focus, not the role

tester

user

- it doesn't matter *who* does the testing, only that it gets done
- establish test goals, measure progres toward these goals
- specialized testing is focus

However

- *Optimizing* the *testing process* is great….

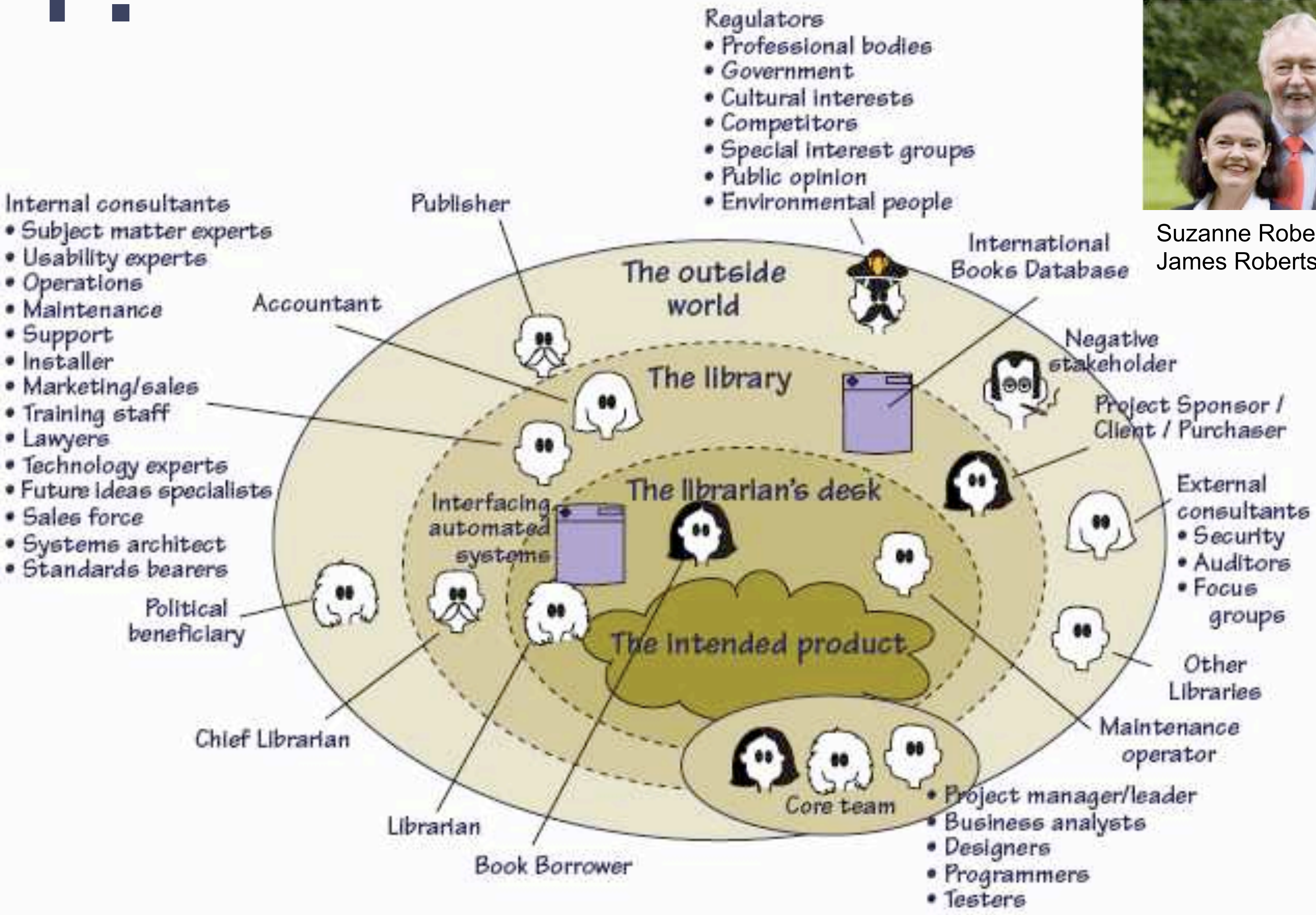
- But,
  - a **lean, upstream, proactive** approach is even far more powerful

- (for getting critical qualities, cost-effectively)

7

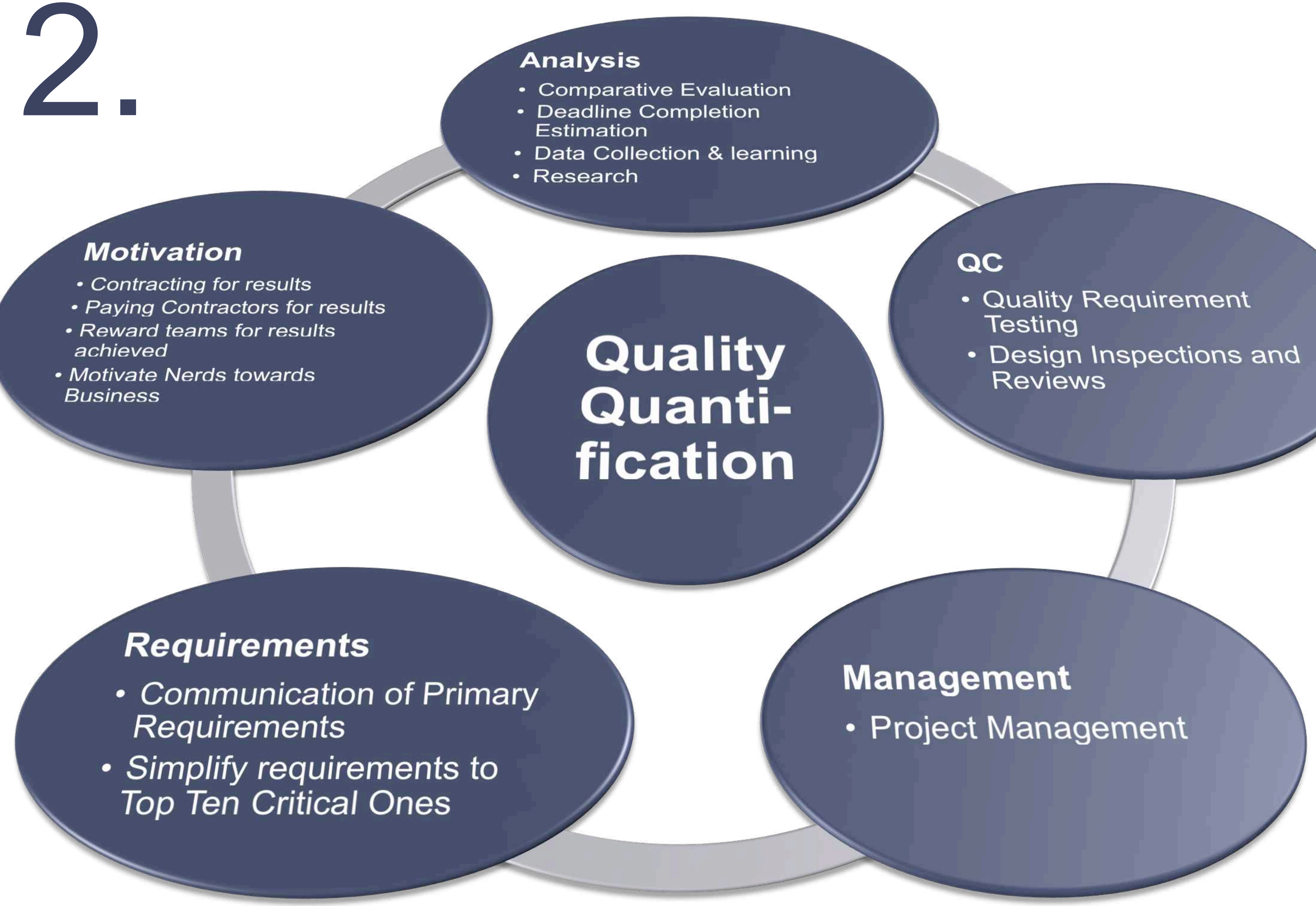
Competitive Lean QA methods to Learn

1. Stakeholders Decide Qualities



Suzanne Robertson & James Robertson

2.



Quality Quanti-fication

Analysis

- Comparative Evaluation
- Deadline Completion Estimation
- Data Collection & learning
- Research

QC

- Quality Requirement Testing
- Design Inspections and Reviews

Management

- Project Management

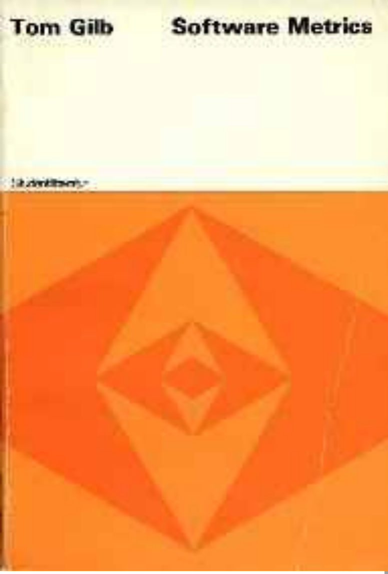
Requirements

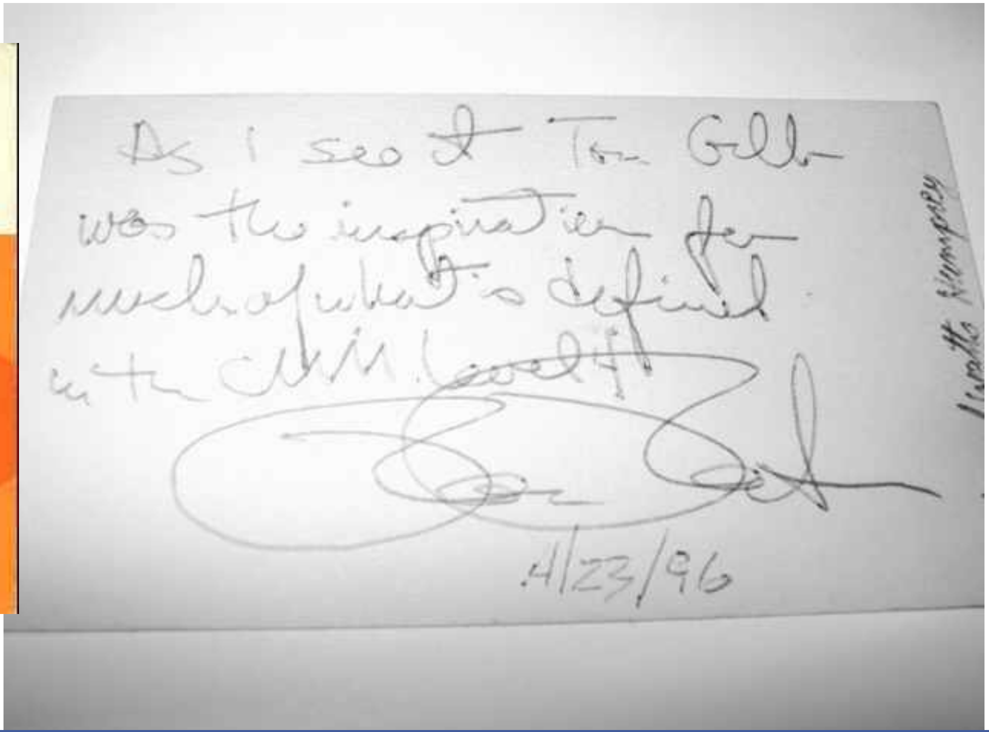
- *Communication of Primary Requirements*
- *Simplify requirements to Top Ten Critical Ones*

Motivation

- *Contracting for results*
- *Paying Contractors for results*
- *Reward teams for results achieved*
- *Motivate Nerds towards Business*

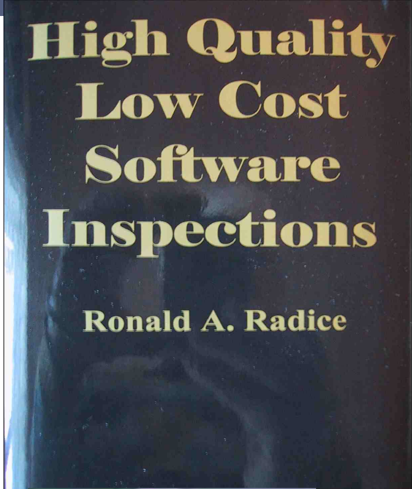
CMM Level 4 Basis



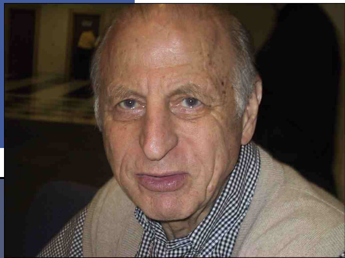

As I see it Tom Gilb was the inspiration for much of what is defined in the CMM level 4

4/23/96

Watts Humphrey



- **"As I see it Tom Gilb was the inspiration for much of what is defined in CMM Level 4."**
- *Ron Radice (CMM Inventor at IBM) 1996 Salt lake City (agreed orally by Watts Humpreys - his IBM Director)*
- *stt@stt.com, www.stt.com*



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

Bad Objectives, for 8 years

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

*2. Will provide a much more efficient **user** experience*

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

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

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

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

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

*8. Major improvements in **data quality** over current practice*

Quantified Objectives (in Planguage),

Robustness.Testability:

Type: Software Quality Requirement.

Version: 20 Oct 2006-10-20

Status: Demo draft,

Stakeholder: {Operator, Tester}.

Ambition: Rapid-duration automatic testing of <critical complex tests>, with extreme operator setup and initiation.

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

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

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

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

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

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

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

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

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

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

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

Managing Risk – Accurate – Consolidated – Real Time

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

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

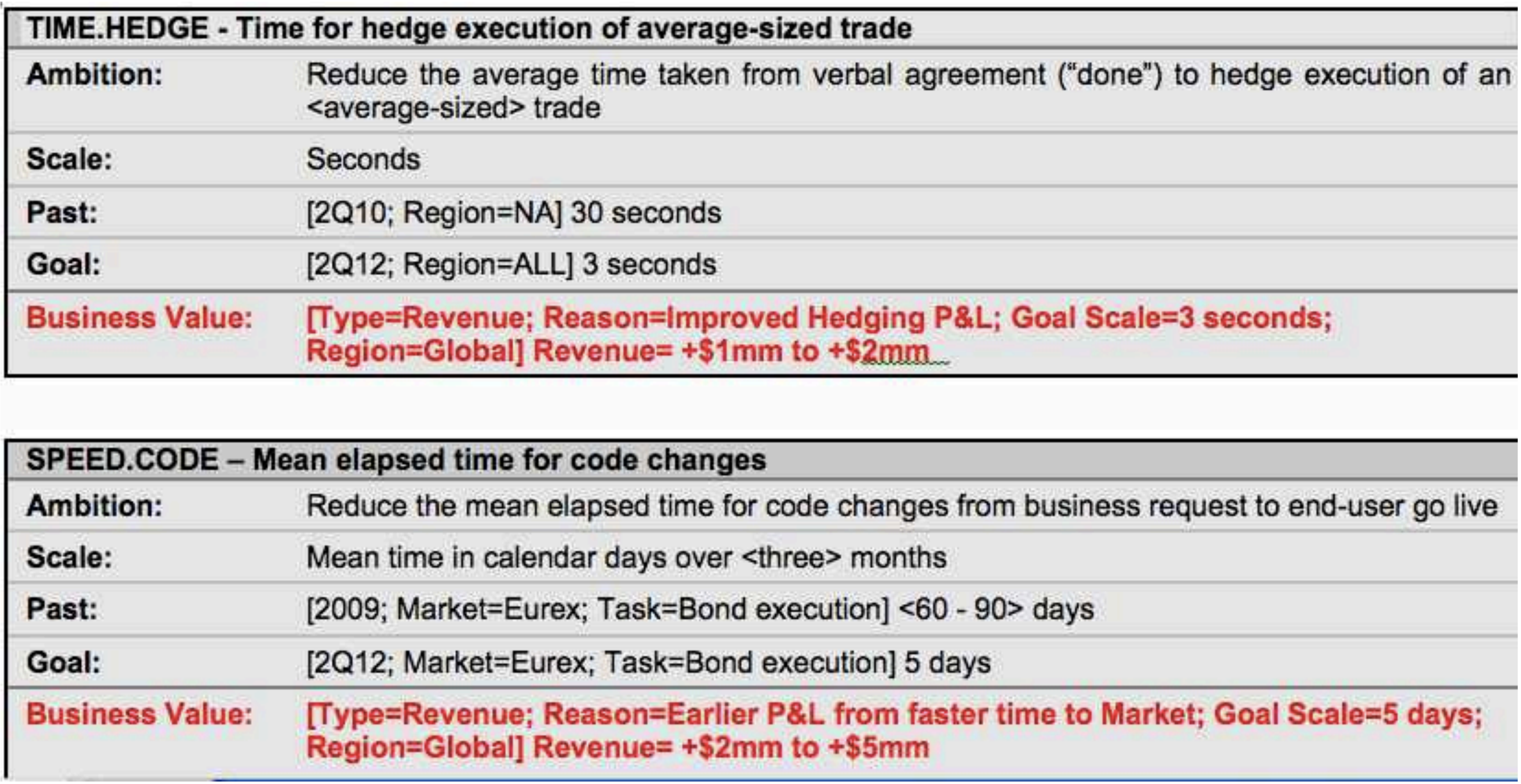
Risk.Accuracy

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

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

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

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

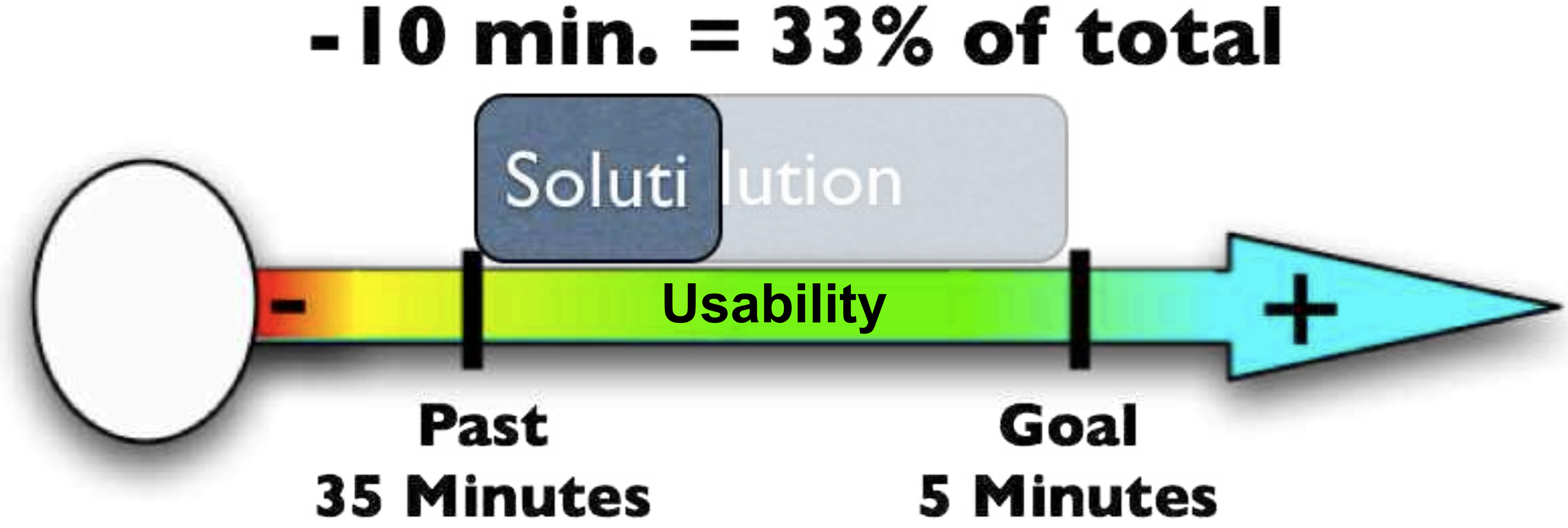


TIME.HEDGE - Time for hedge execution of average-sized trade	
Ambition:	Reduce the average time taken from verbal agreement ("done") to hedge execution of an <average-sized> trade
Scale:	Seconds
Past:	[2Q10; Region=NA] 30 seconds
Goal:	[2Q12; Region=ALL] 3 seconds
Business Value:	[Type=Revenue; Reason=Improved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

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

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

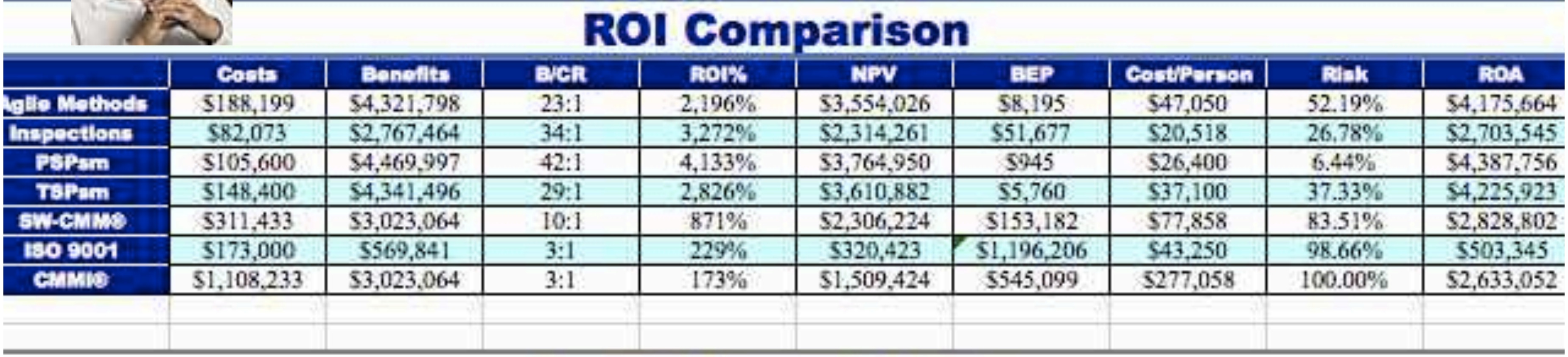
3. Assuring that Designs give Qualities



4. Measure Quality Levels in Specifications with Inspection

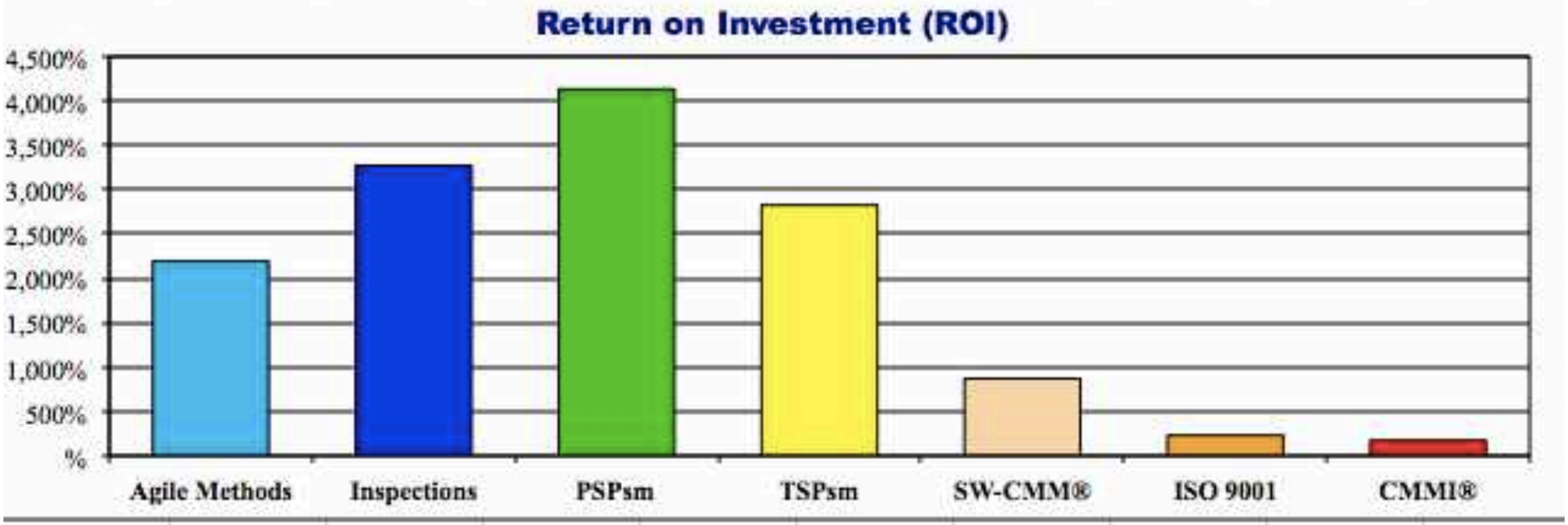
Value for Money Inspection and CMMI

David Rico, http://davidfrico.com



ROI Comparison

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



Return on Investment (ROI)

A Recent Example

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

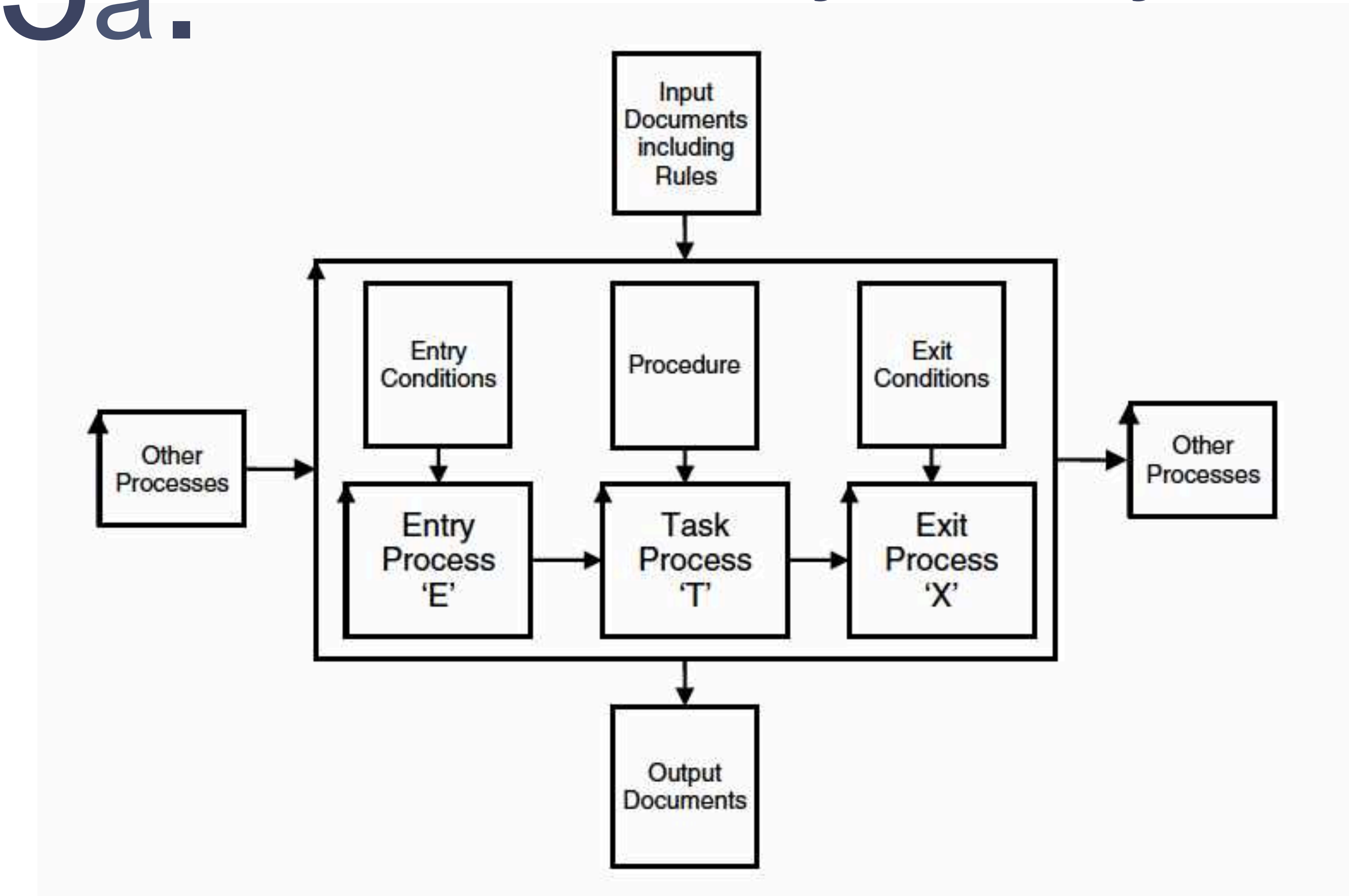
Application of Specification Quality Control (Gilb Inspections) by a SW team resulted in the following defect density reduction in requirements over several months:

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%

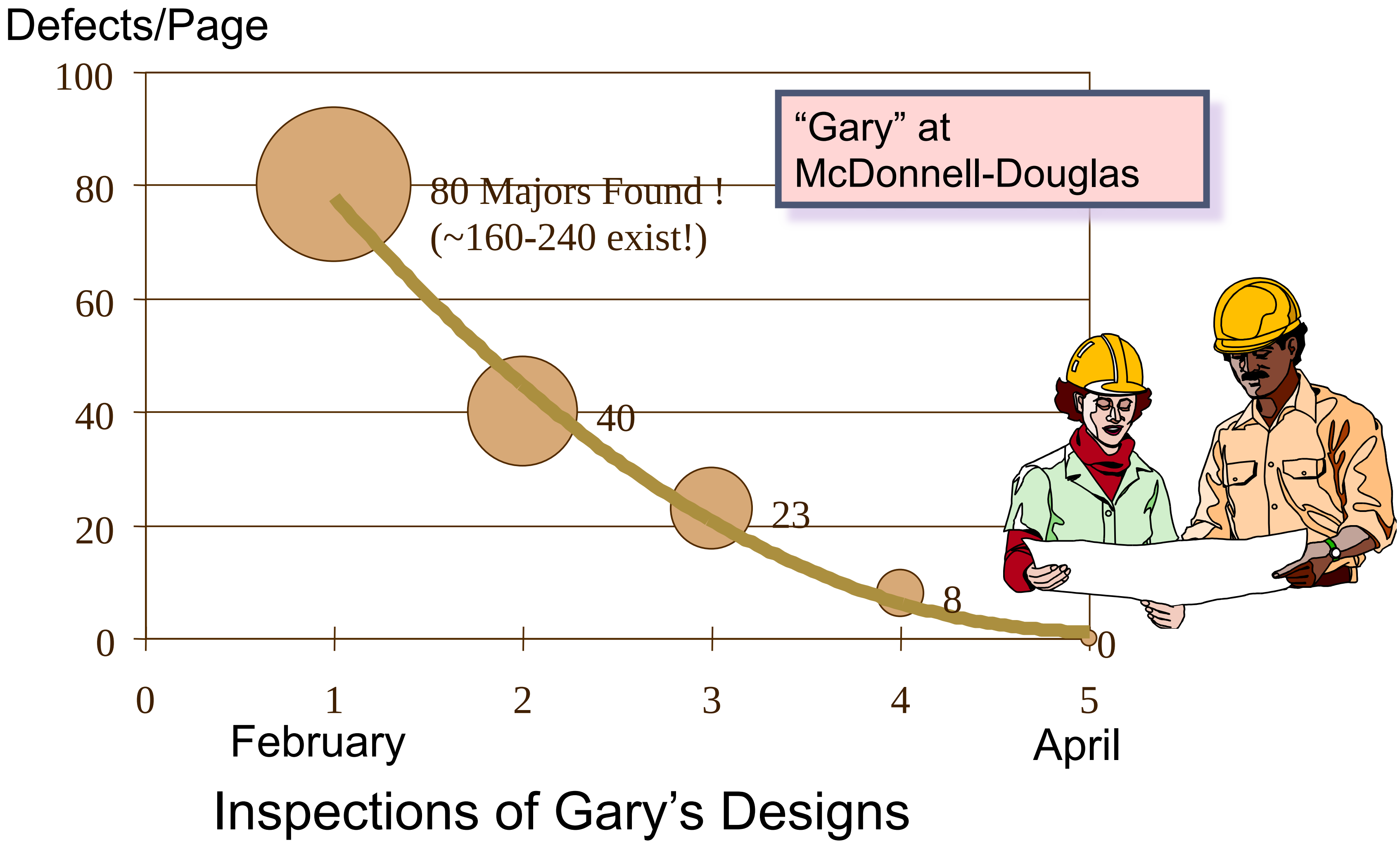
Downstream benefits:

- 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

5a. Numeric Quality Gateways

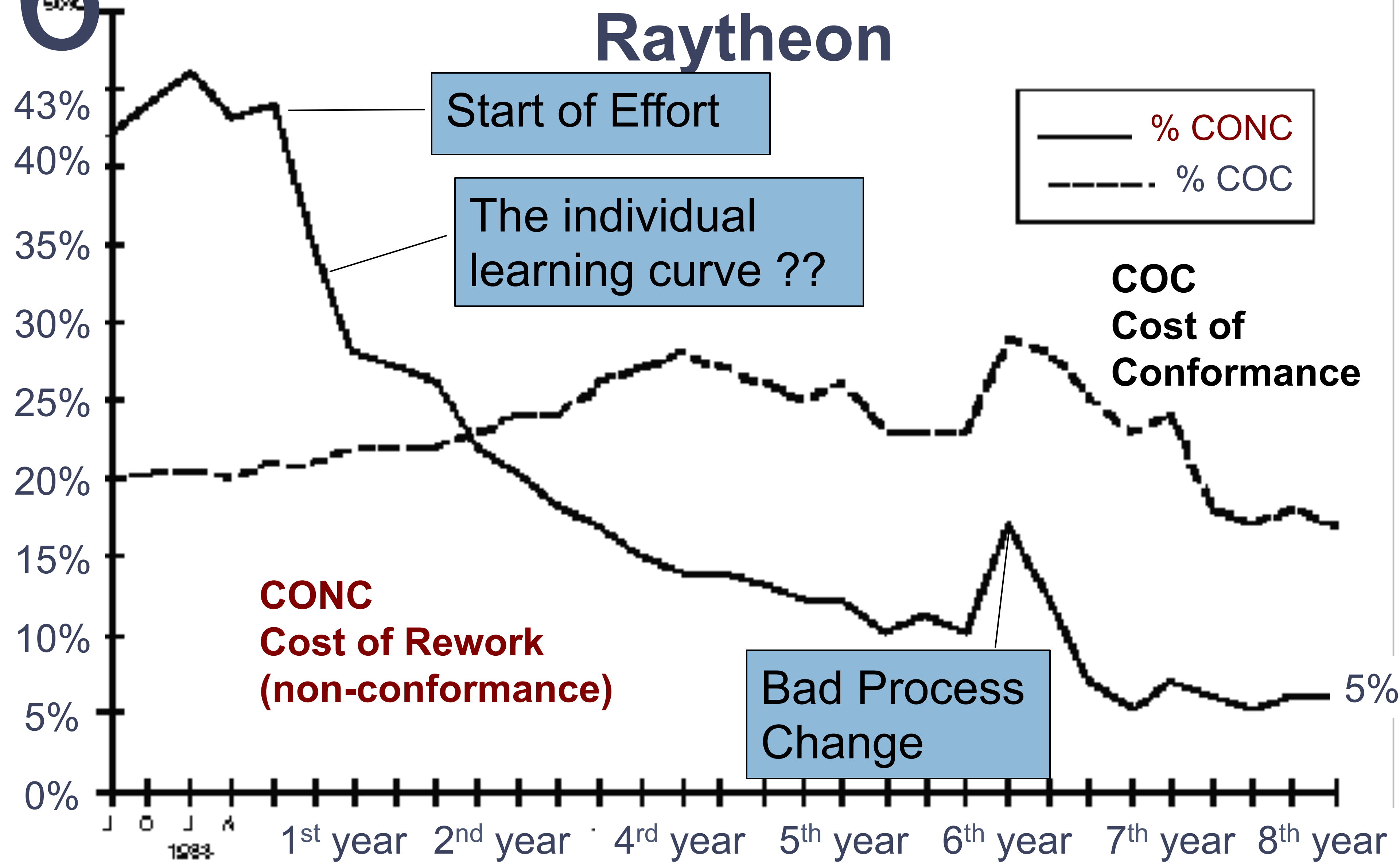


5a. Numeric Quality Gateways Improve Quality of work



“Gary” at McDonnell-Douglas

6 DPP (=CMM 5) Improves Quality by 10x: Raytheon



Start of Effort

The individual learning curve ??

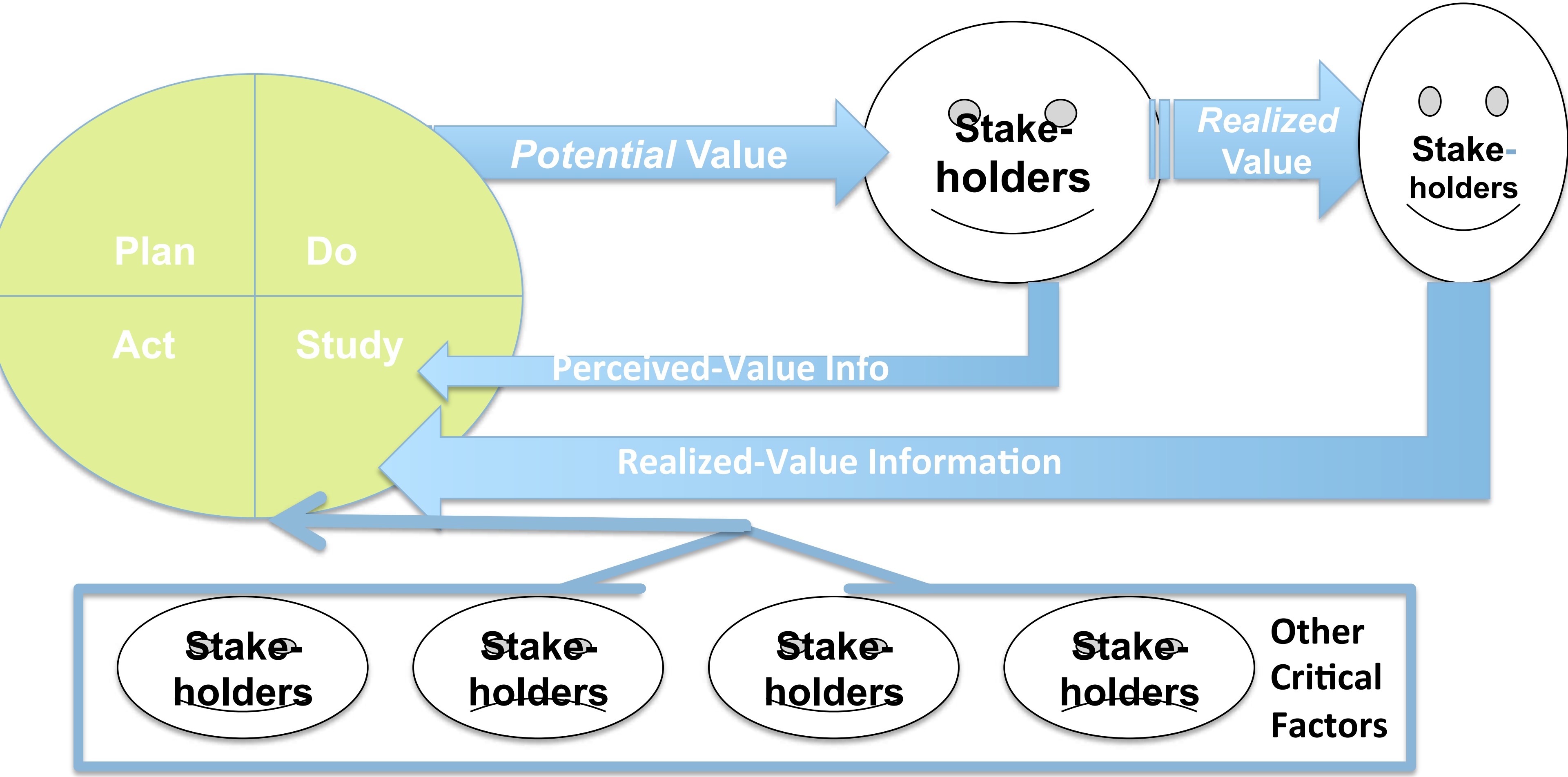
COC
Cost of Conformance

CONC
Cost of Rework (non-conformance)

Bad Process Change

www.sei.cmu.edu/publications/documents/95.reports/95.tr.017.html

7 a Frequent feedback and improvement assure quality



- 2 Kinds of Feedback from Stakeholders, when value increment is *really* exploited in practice after delivery.
- Combined with other information from the relevant environment. Like budget, deadline, technology, politics, laws, marketing changes.

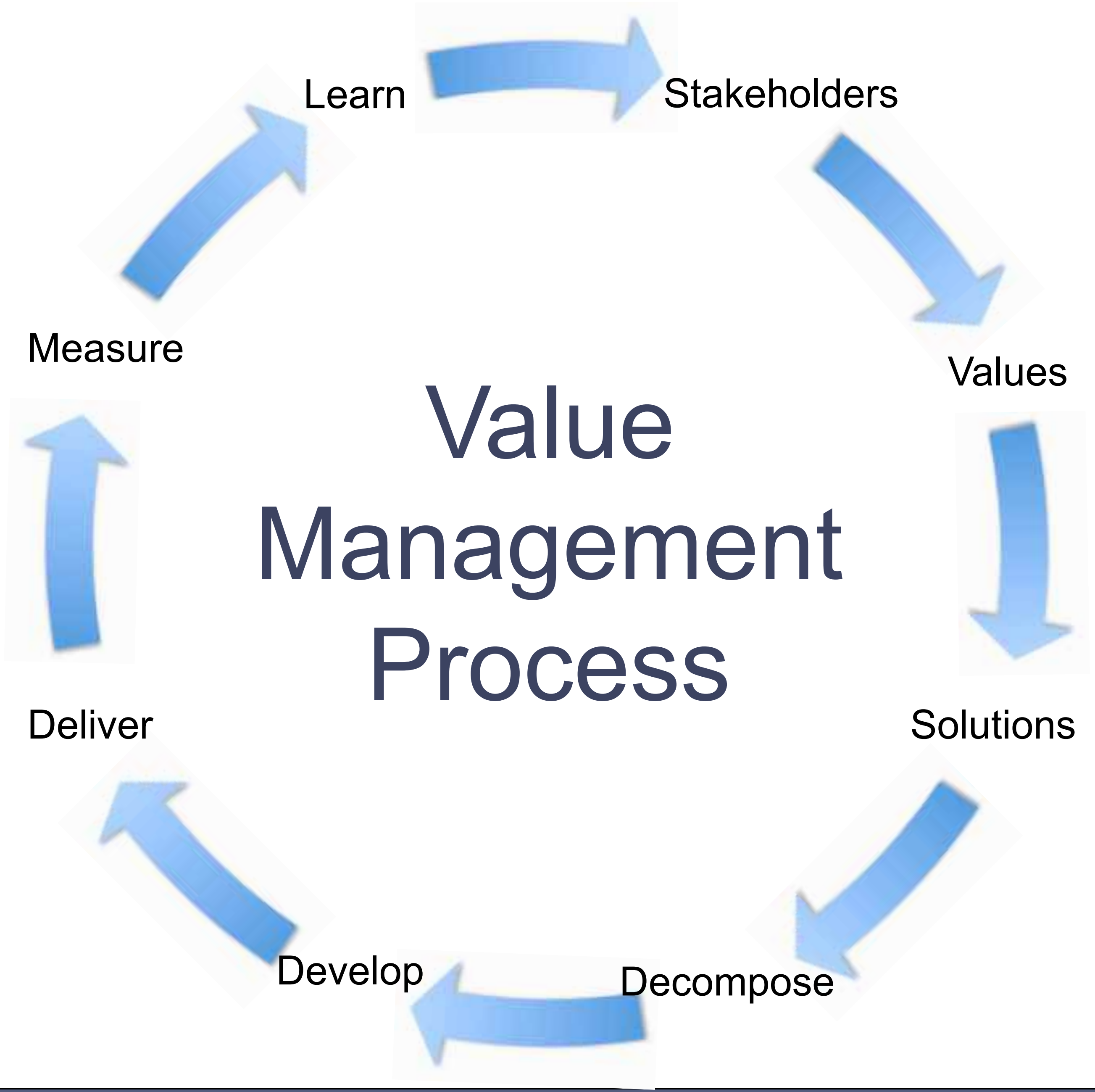
Recent (20 Sept, 2011) Report on Gilb Evo method (Richard Smith, Citigroup)

- 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 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"

7b

Value Management Process



Learn

Stakeholders

Values

Solutions

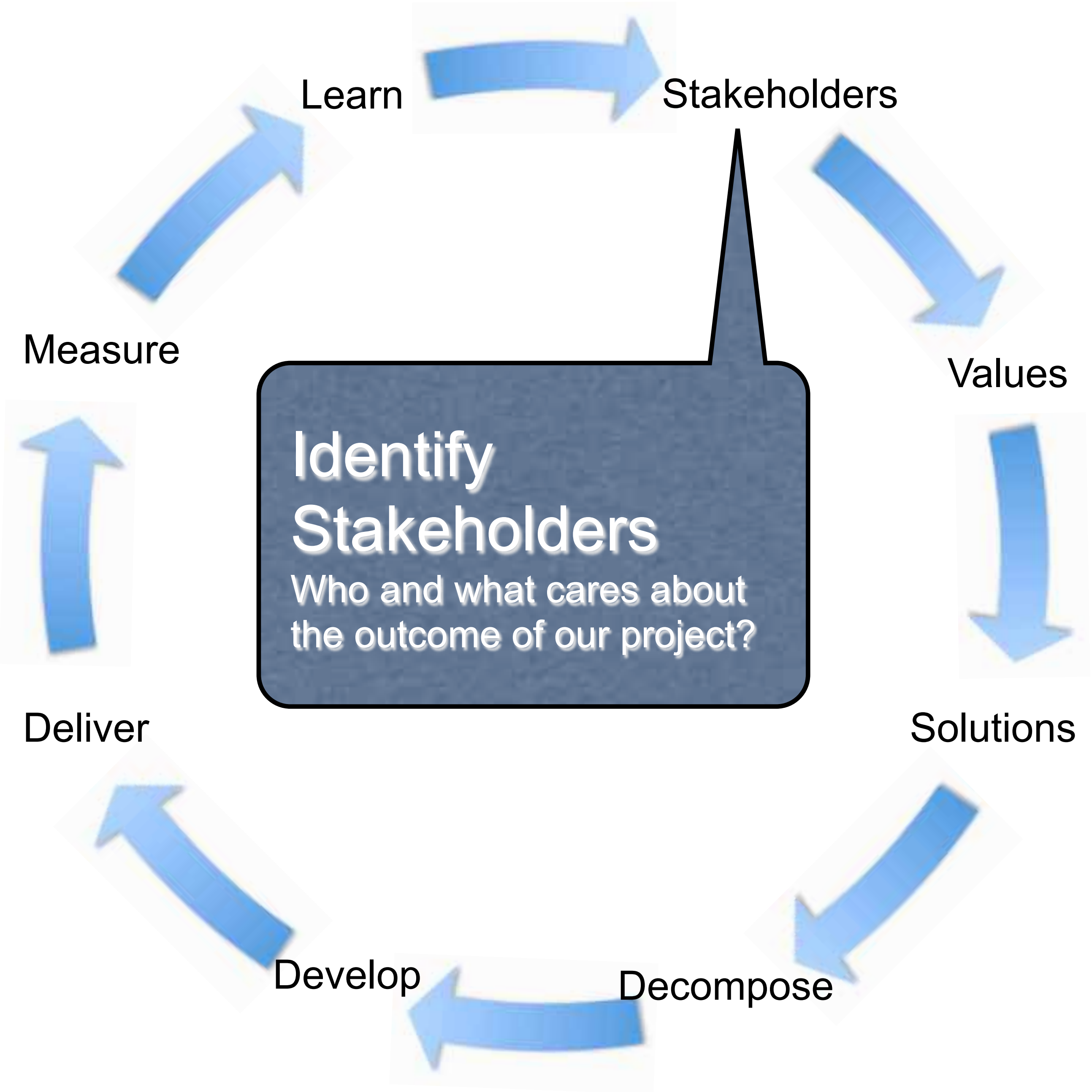
Decompose

Develop

Deliver

Measure

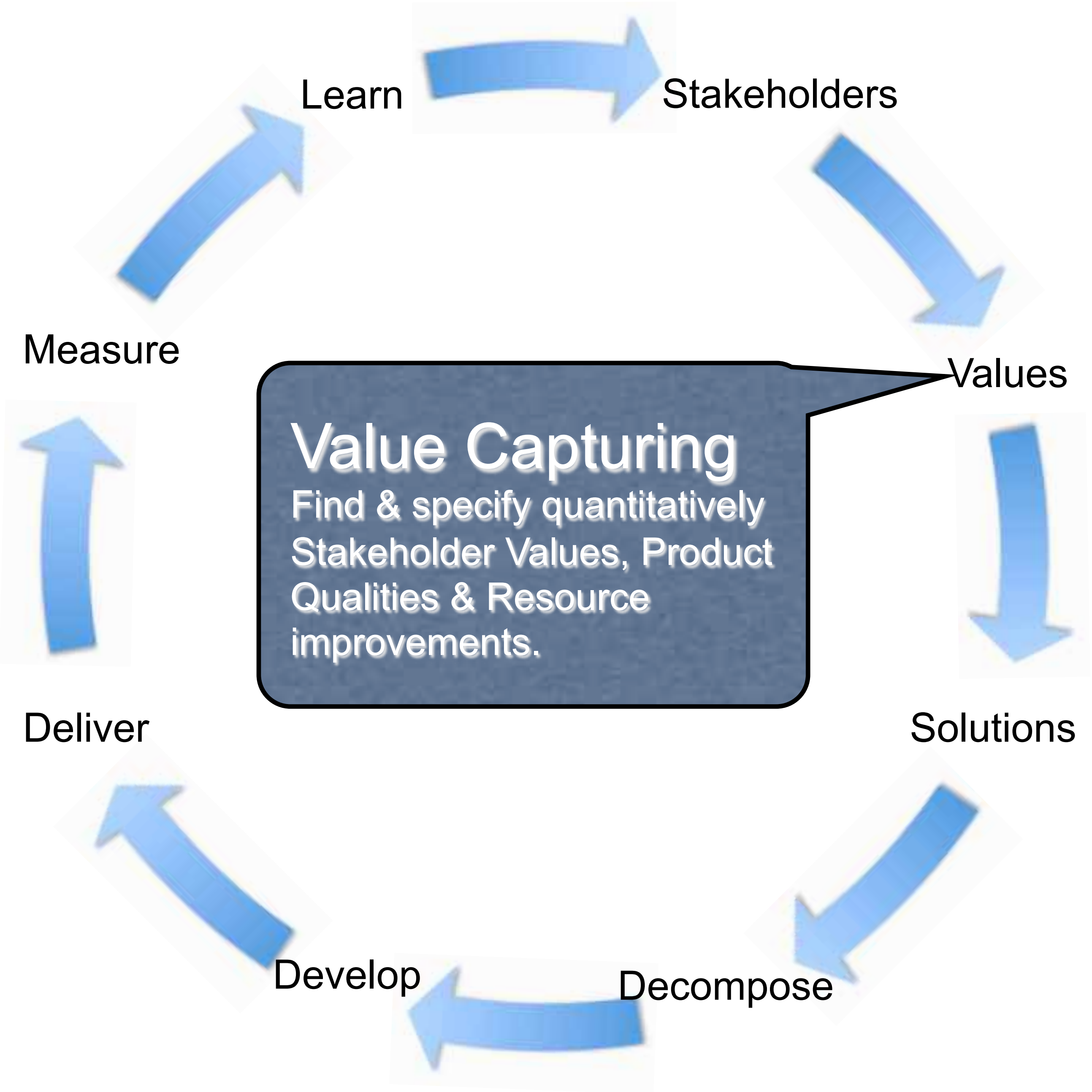
7b



Identify Stakeholders

Who and what cares about the outcome of our project?

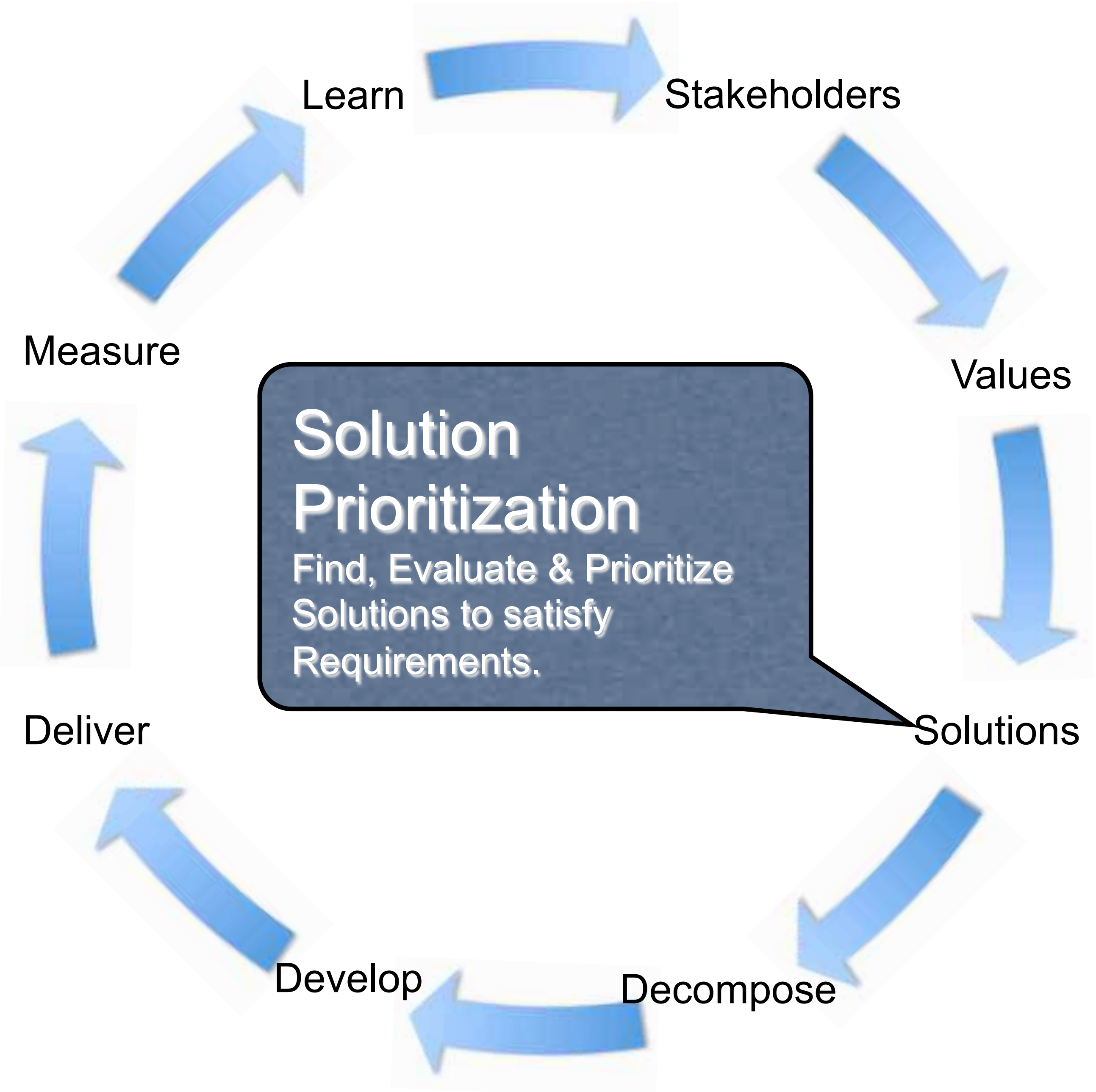
7b



Value Capturing

Find & specify quantitatively Stakeholder Values, Product Qualities & Resource improvements.

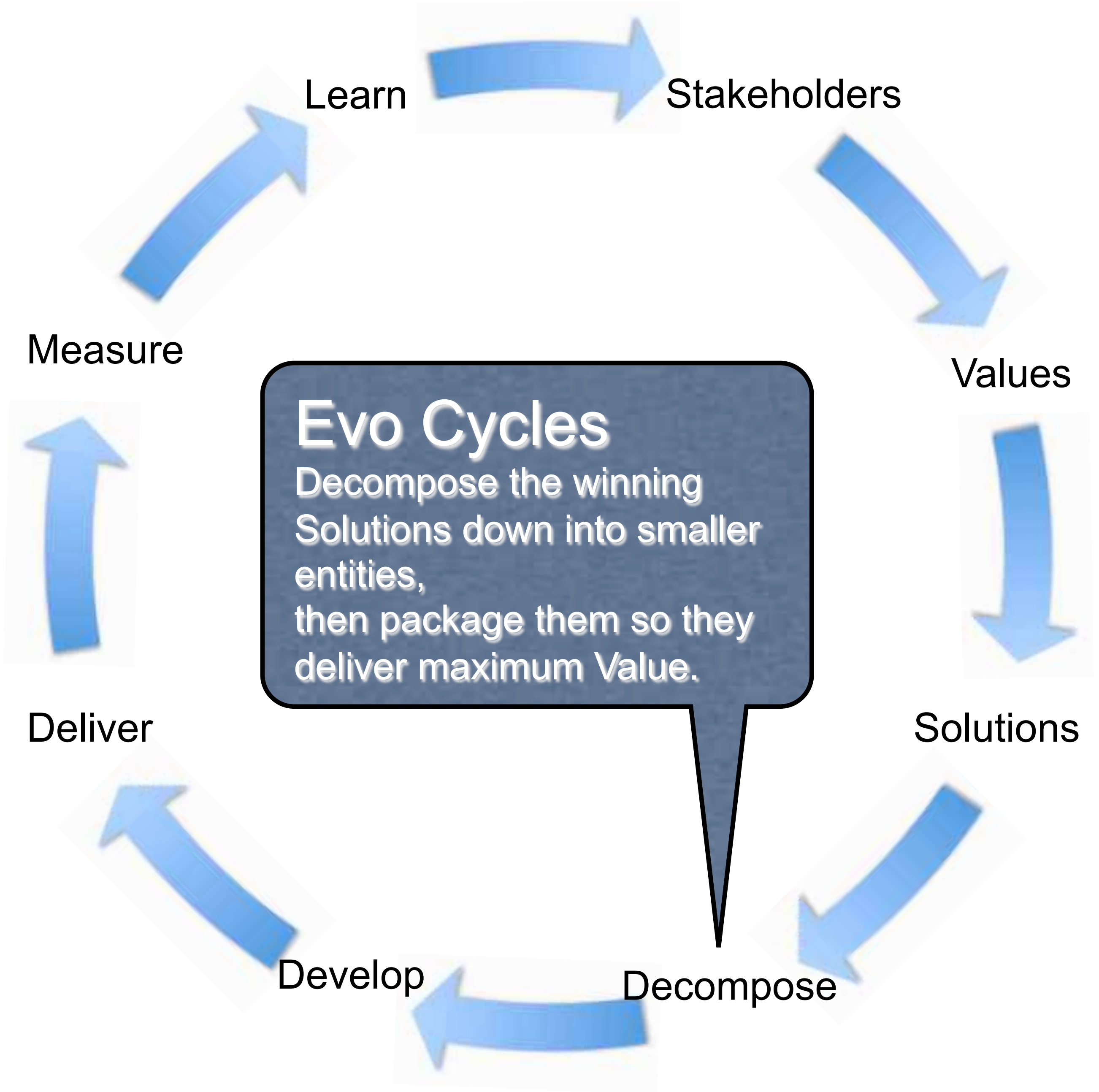
7b



Solution Prioritization

Find, Evaluate & Prioritize Solutions to satisfy Requirements.

7b



Learn

Stakeholders

Values

Solutions

Decompose

Develop

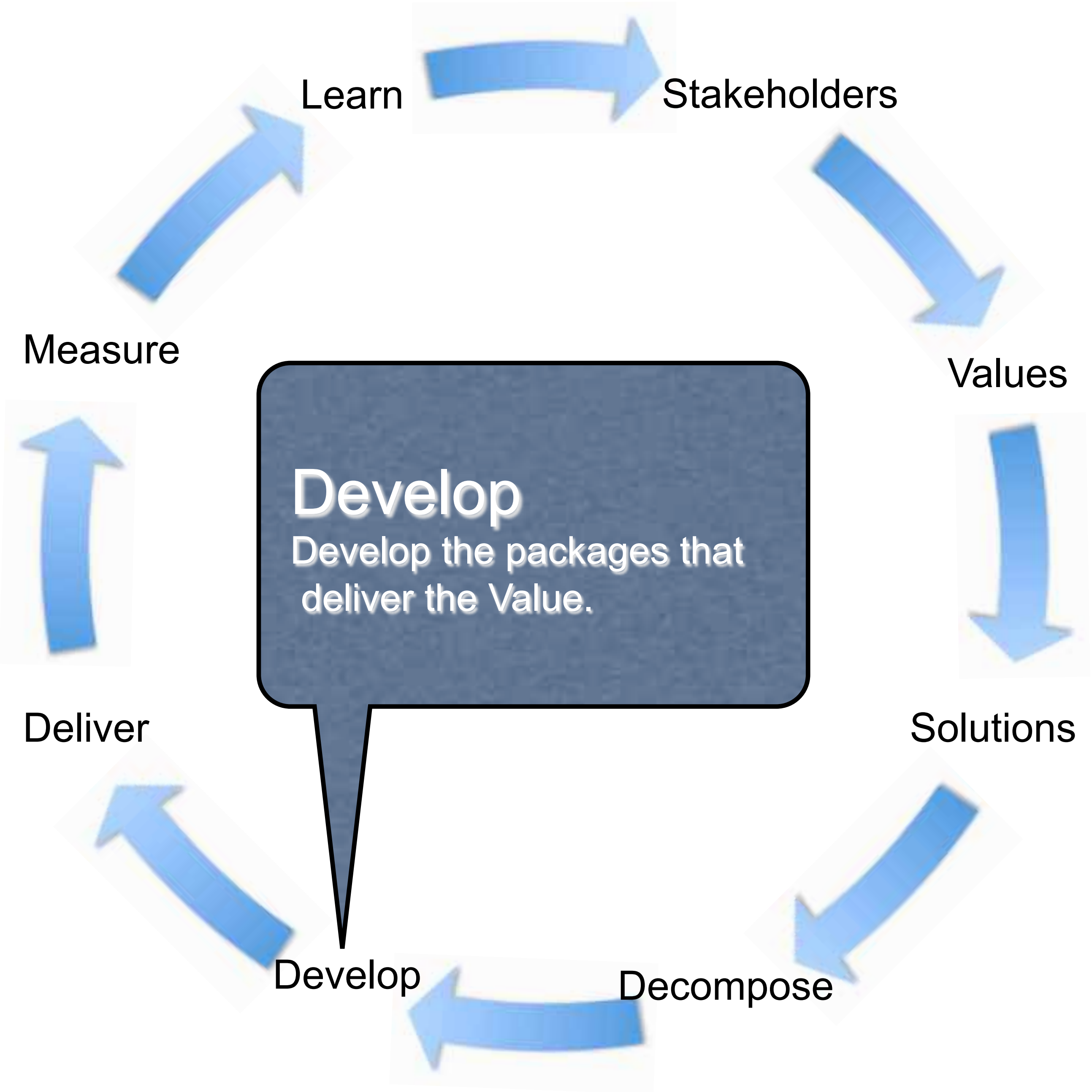
Deliver

Measure

Evo Cycles

Decompose the winning Solutions down into smaller entities,
then package them so they deliver maximum Value.

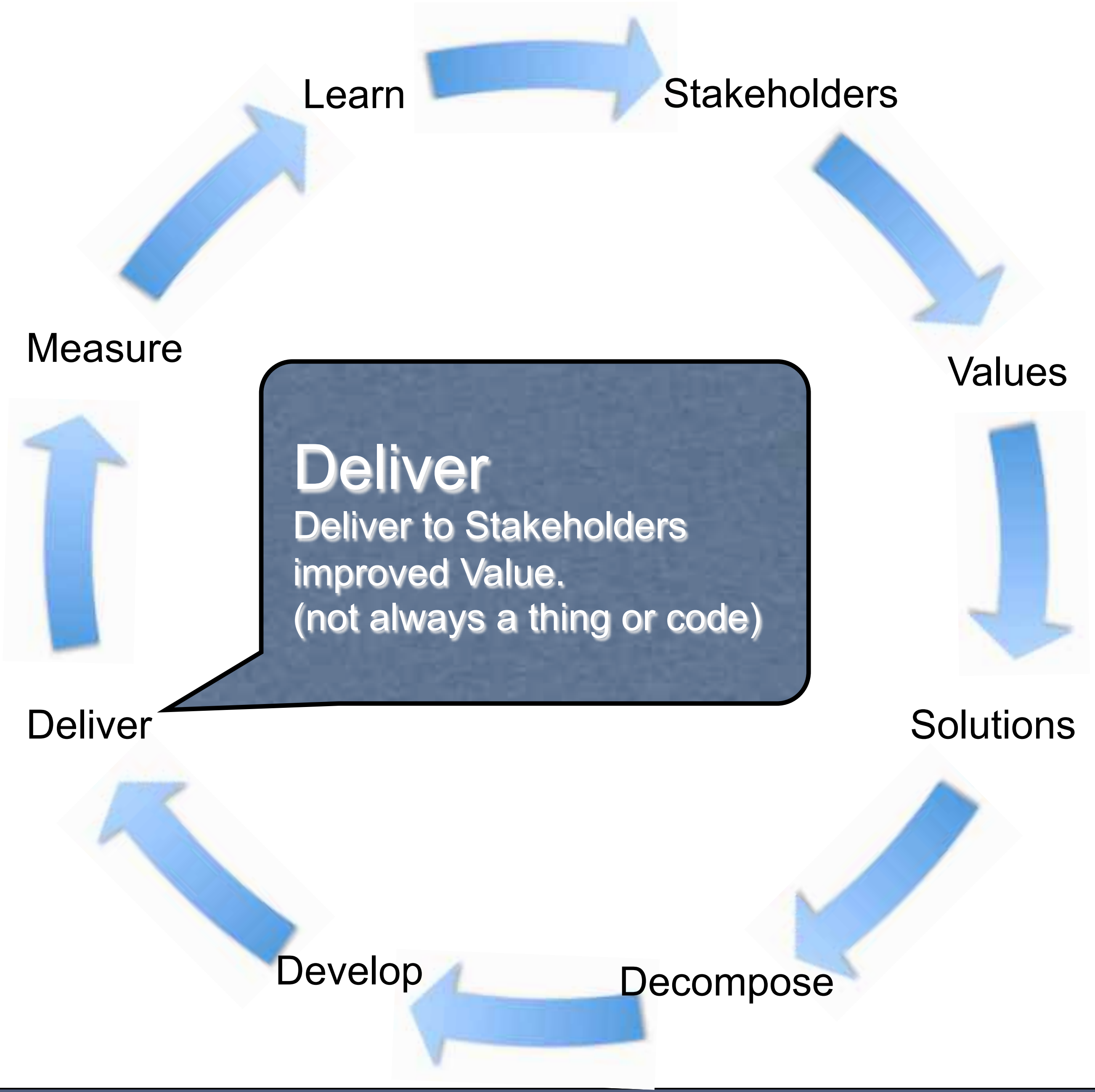
7b



Develop

Develop the packages that deliver the Value.

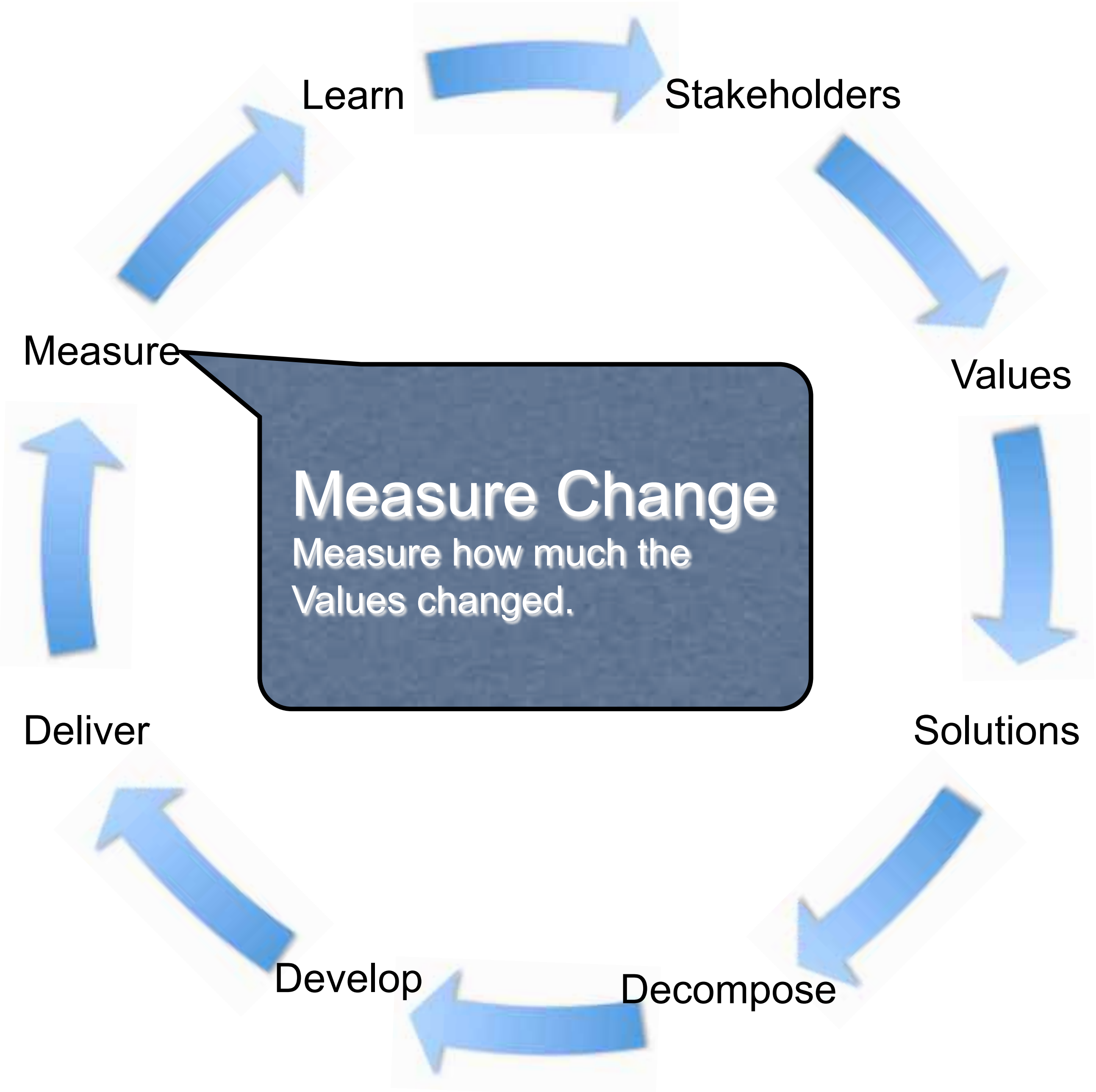
7b



Deliver

Deliver to Stakeholders improved Value.
(not always a thing or code)

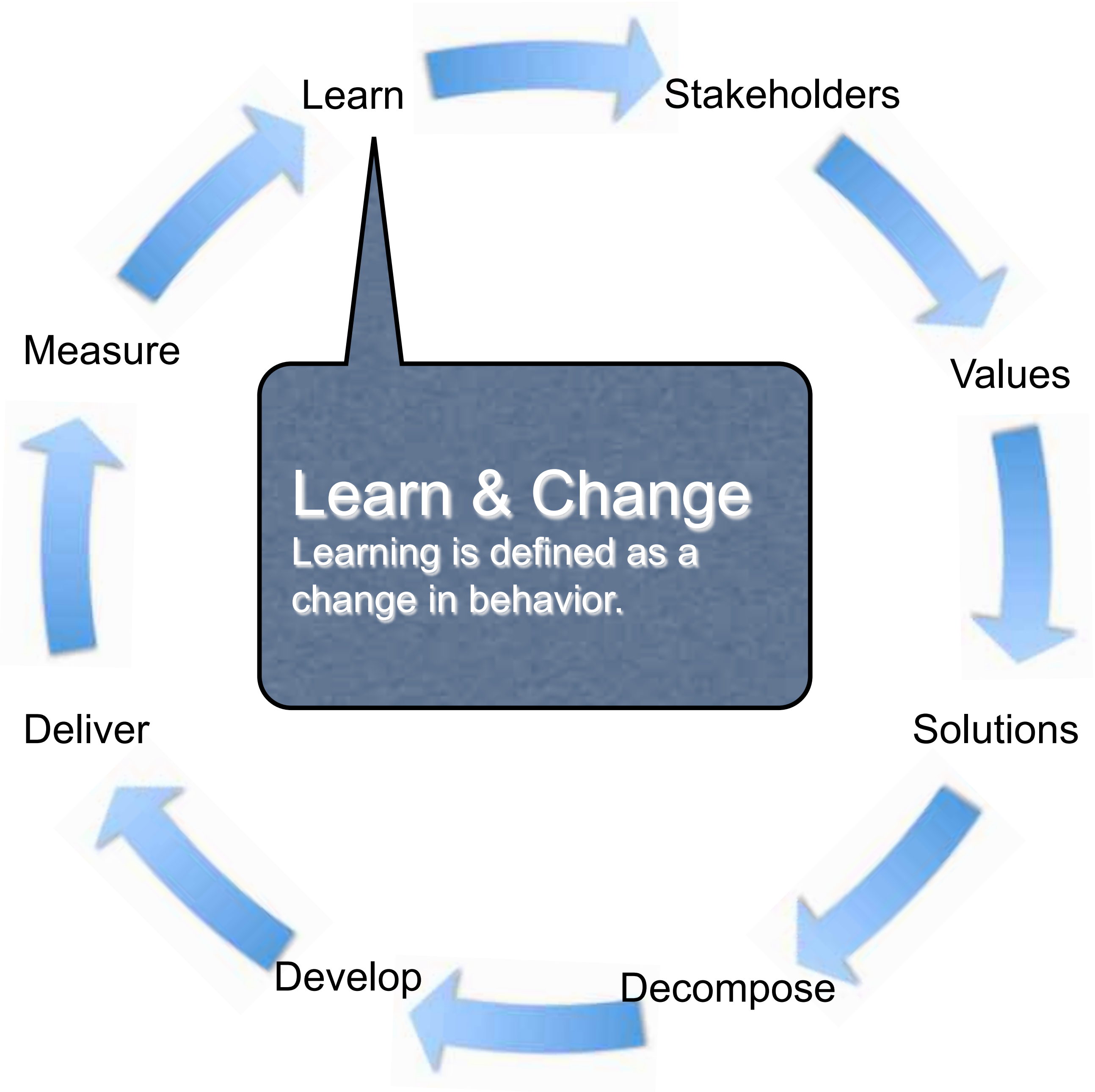
7b



Measure Change

Measure how much the Values changed.

7b

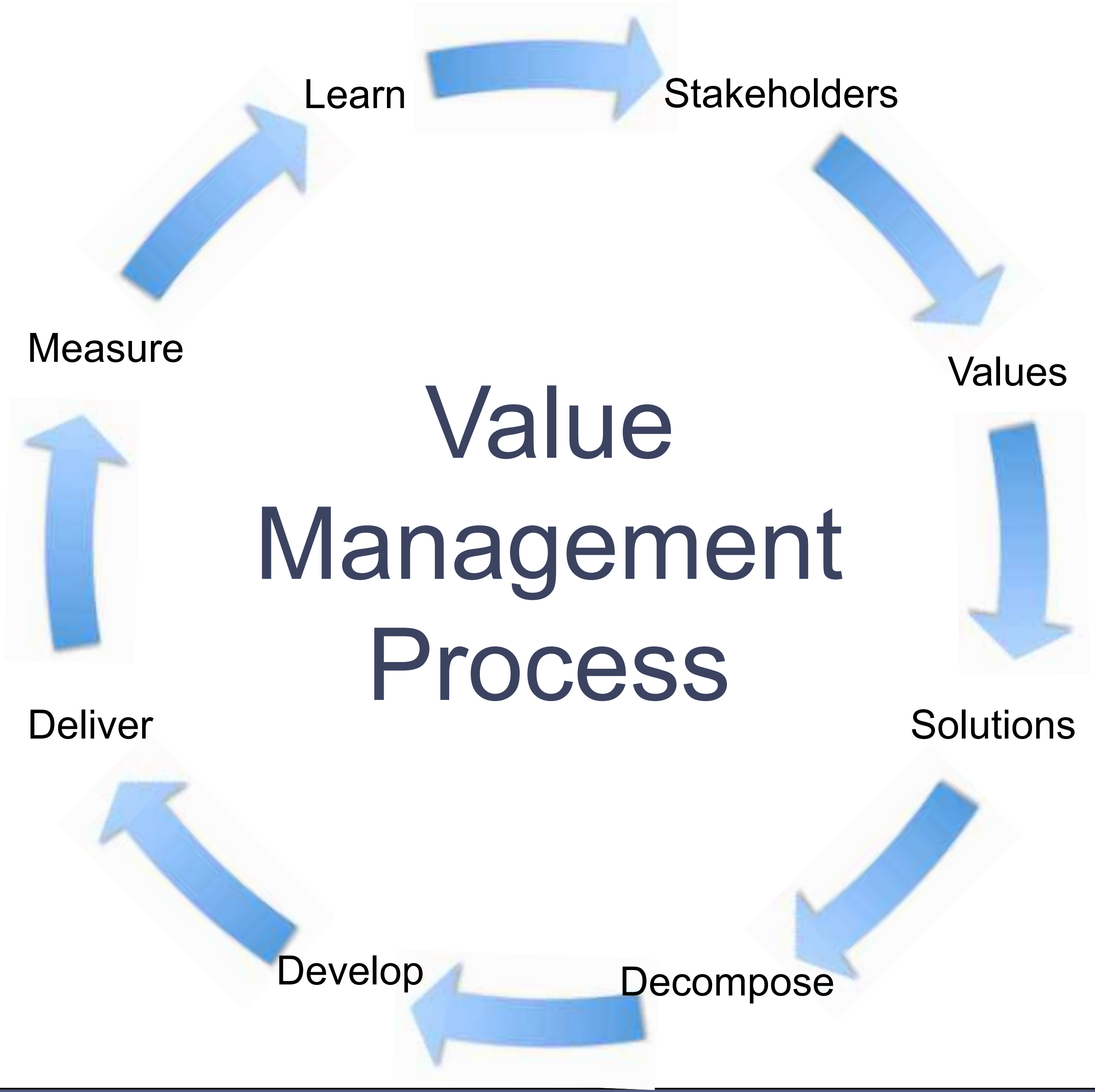


Learn & Change

Learning is defined as a change in behavior.

7b

Value Management Process



Learn

Stakeholders

Values

Solutions

Decompose

Develop

Deliver

Measure

End

7

Competitive Lean QA methods to Learn

What you can do immediately

①!Identify the 5 most critical qualities of your system.

②!Quantify the 5 qualities.

③!For each quality,

①! set a Current level

②! and a Goal level

Main Take-away Points

Quality Assurance is far more than ‘test’,

and it can be far more cost-effective

‘Quality’ is far more than ‘bugs’

You probably have a lot to learn,

if you want real competitive quality

Thanks!

Discussion After lecture, all during the conference.

Tom@Gilb.com
Mobile: +47 920 66 705
www.Gilb.com

Older Copy of these slides will be in Gilb.com Downloads/ Slides:

http://gilb.com/tiki-list_file_gallery.php?galleryId=14

The Lean Quality Assurance Methods

- !Everything ‘not adding value to the Customer’ is considered to be waste.
  - !This includes:
    - !unnecessary code and functionality
    - !Delay in the software development process
    - !Unclear requirements
    - !Bureaucracy
    - !Slow internal communication
  - !Amplify Learning
    - !The learning process is sped up by usage of short iteration cycles – each one coupled with refactoring and integration testing. Increasing feedback via short feedback sessions with Customers helps when determining the current phase of development and adjusting efforts for future improvements.
  - !Decide as late as possible
  - !Deliver as fast as possible
  - !Empower the team
  - !Build integrity in
    - !separate components work well together as a whole with balance between flexibility, maintainability, efficiency, and responsiveness.
  - !See the whole
    - !“Think big, act small, fail fast; learn rapidly”