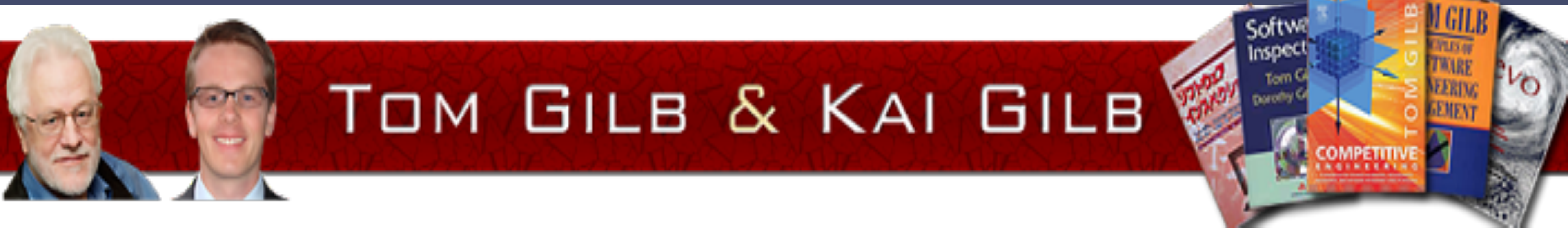


‘Lean’

Quality Assurance

January 21 2016
Quality Days, Workshop
Vienna

by Tom Gilb

Tom@Gilb.com @imtomgilb
www.gilb.com
These slides will be at:
http://www.gilb.com/

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

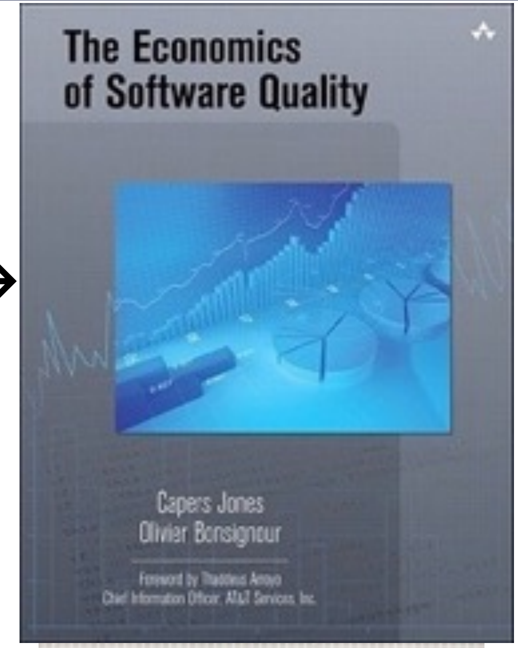
Begin:
Quality Assurance is far more than ‘test’

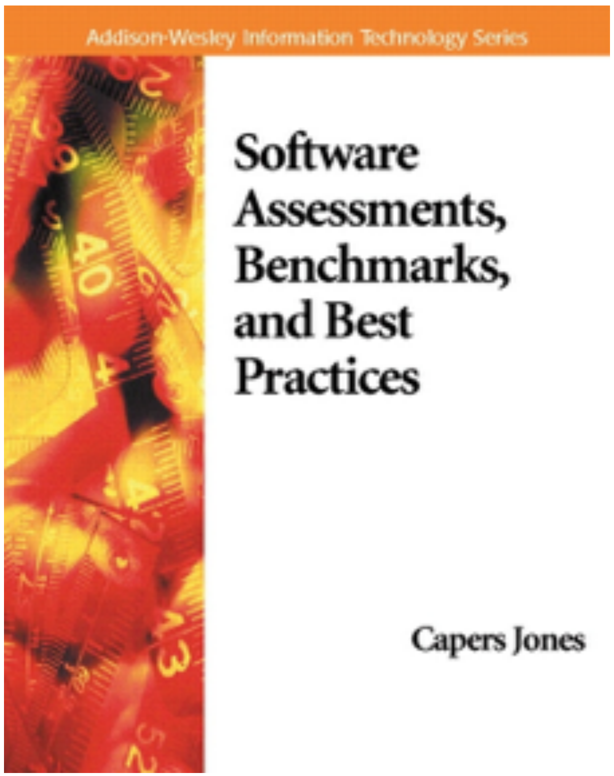
and it can be far more cost-effective

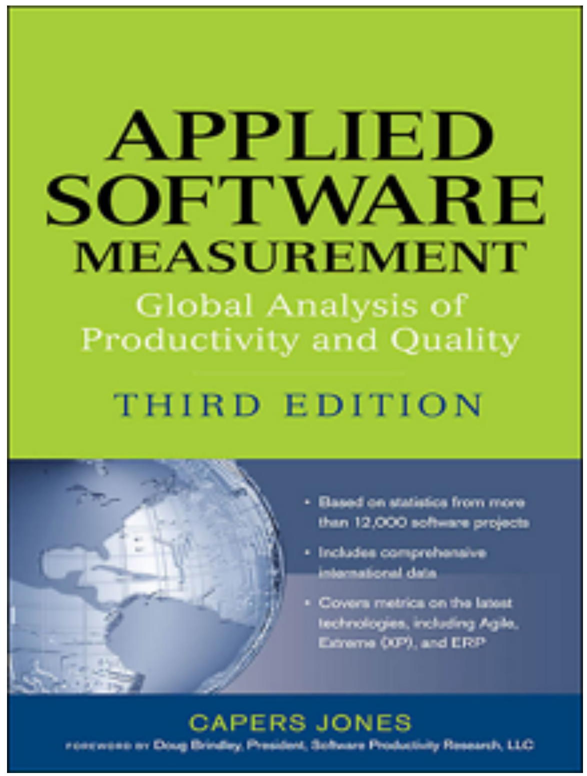
Inspection Effectiveness

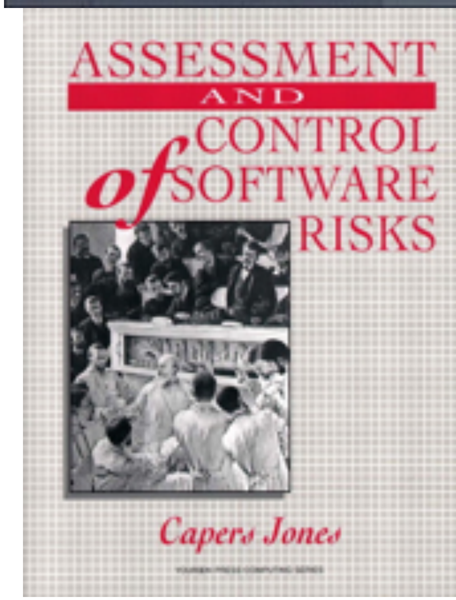
Capers Jones

Latest book -→









Regression test ?

15% to 30%

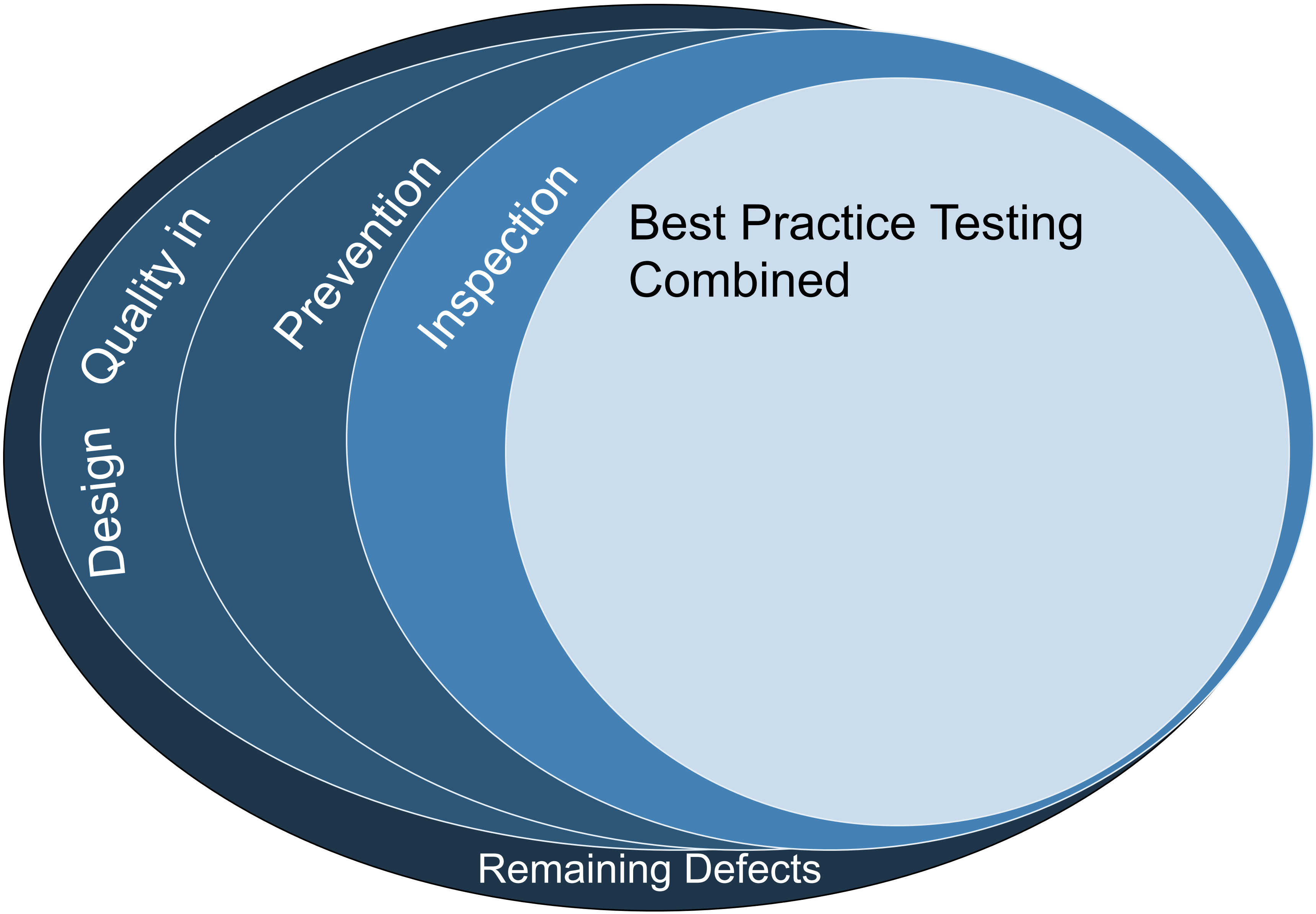
Integration test ?

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

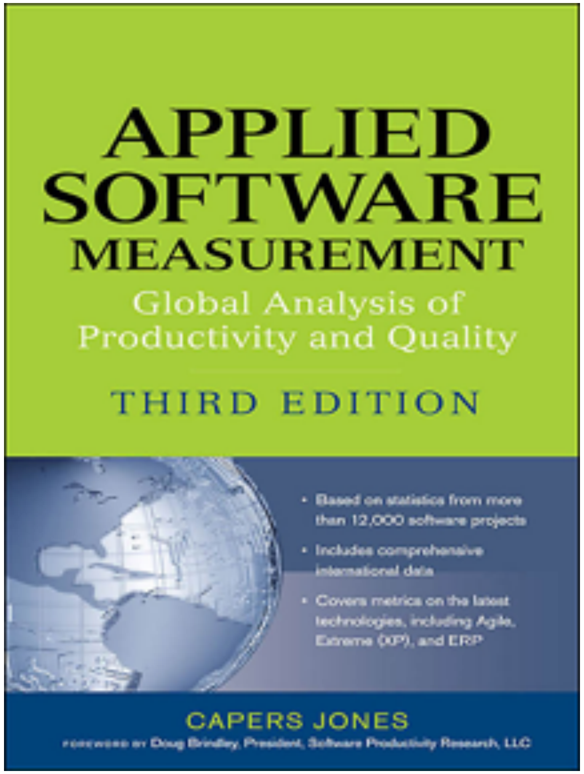
Little hope of 'zero defects'

"Between

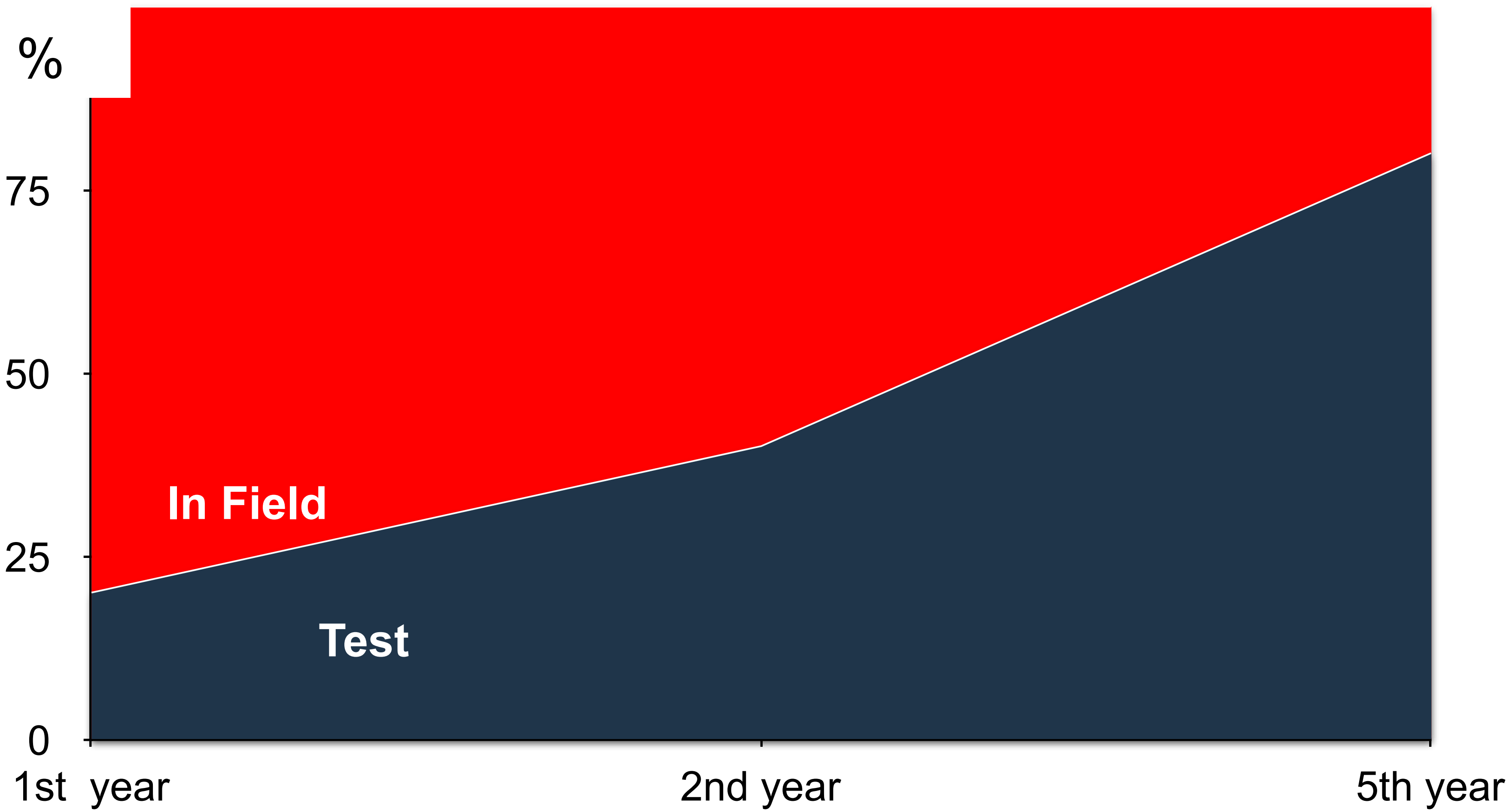
8 and 10

defect removal stages required to achieve removal effectiveness of

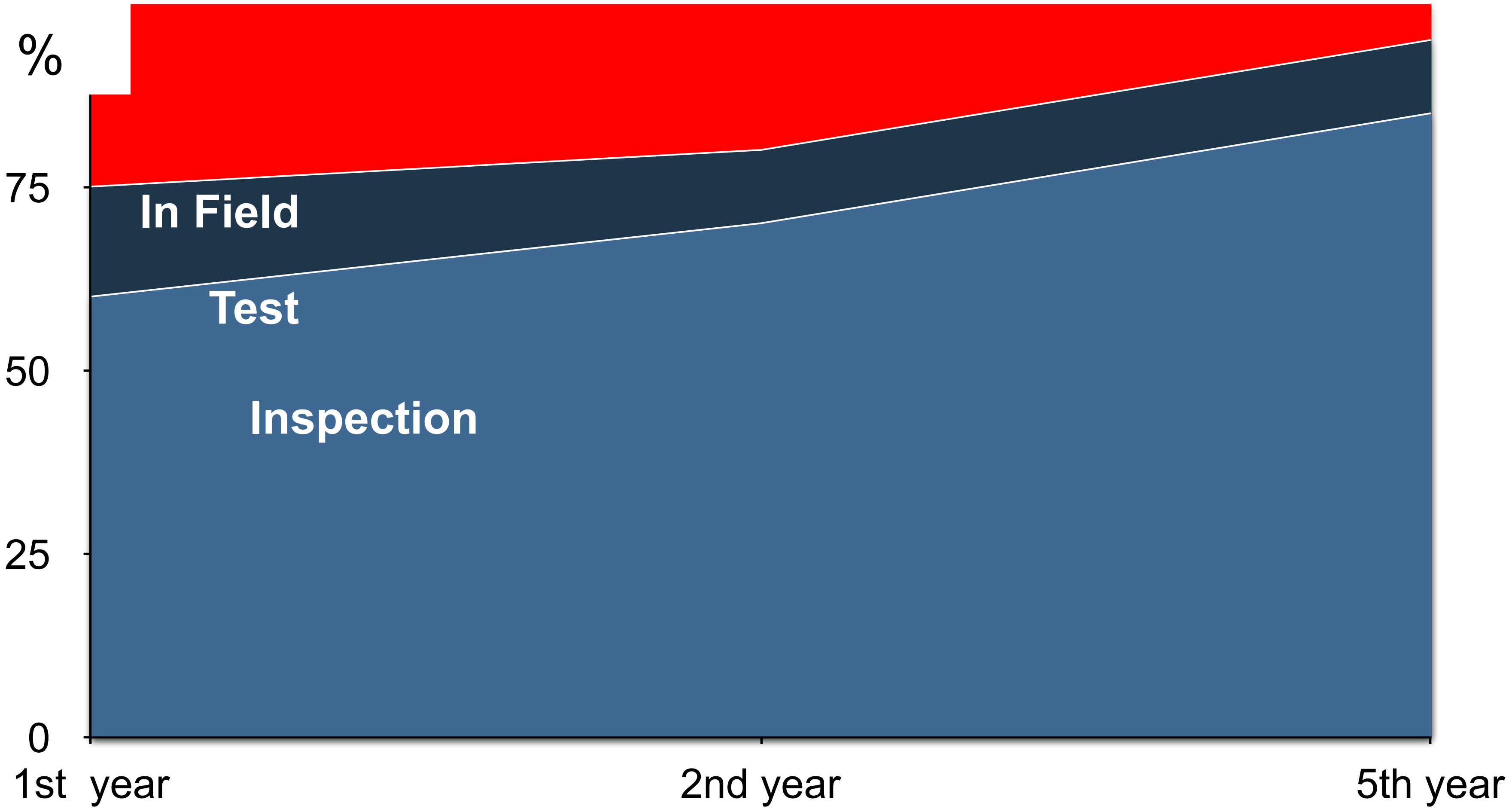
95%"



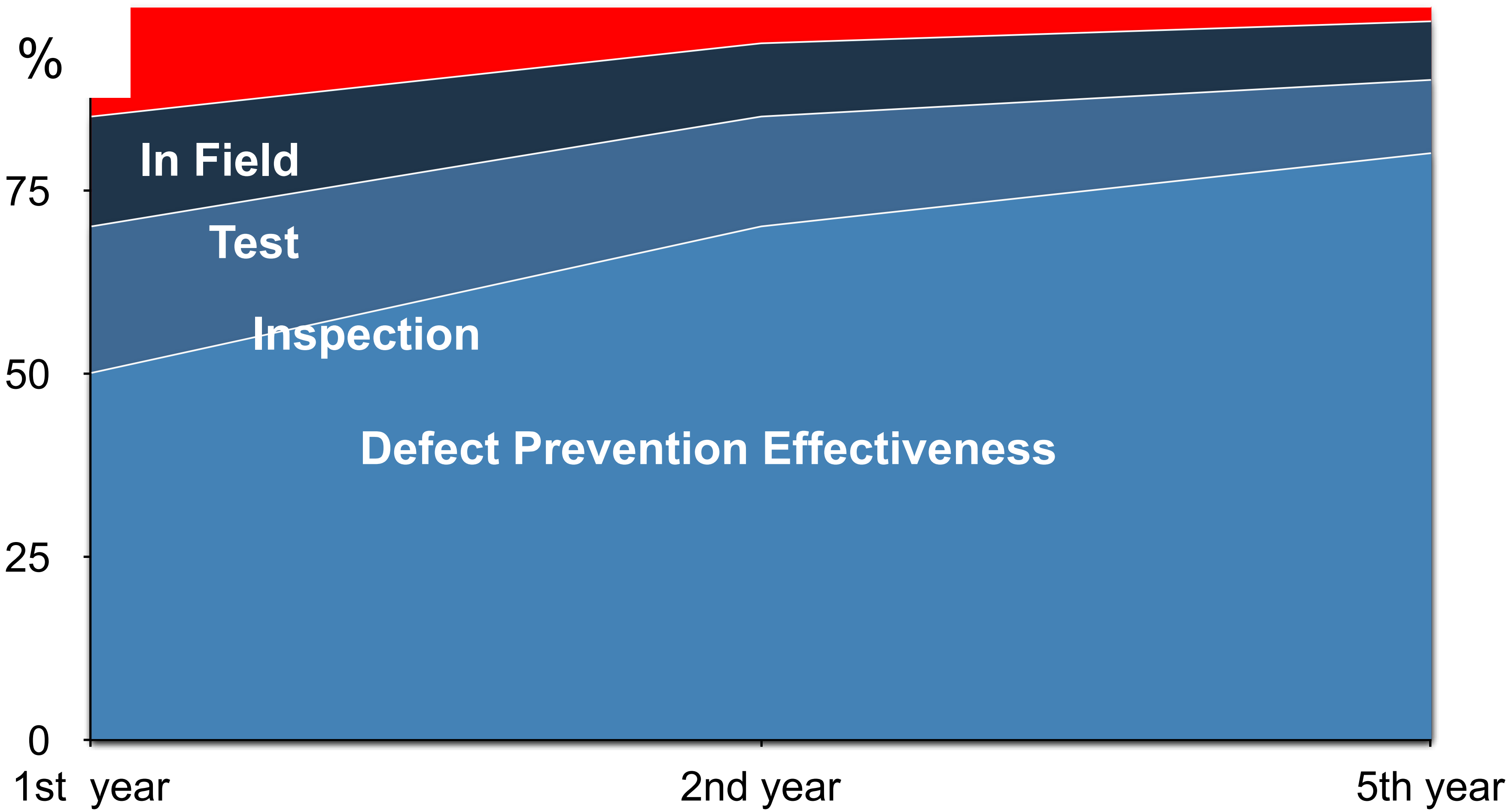
Testing Capability (C. Jones)



Defect Detection Capability (C. Jones)



IBM Defect Avoidance Experience



End of Introduction: Quality Assurance is far more than ‘test’

and, QA can be far more cost-effective.

Now for some more practical detail about what and how to do QA

Quality is far more than 'bugs'

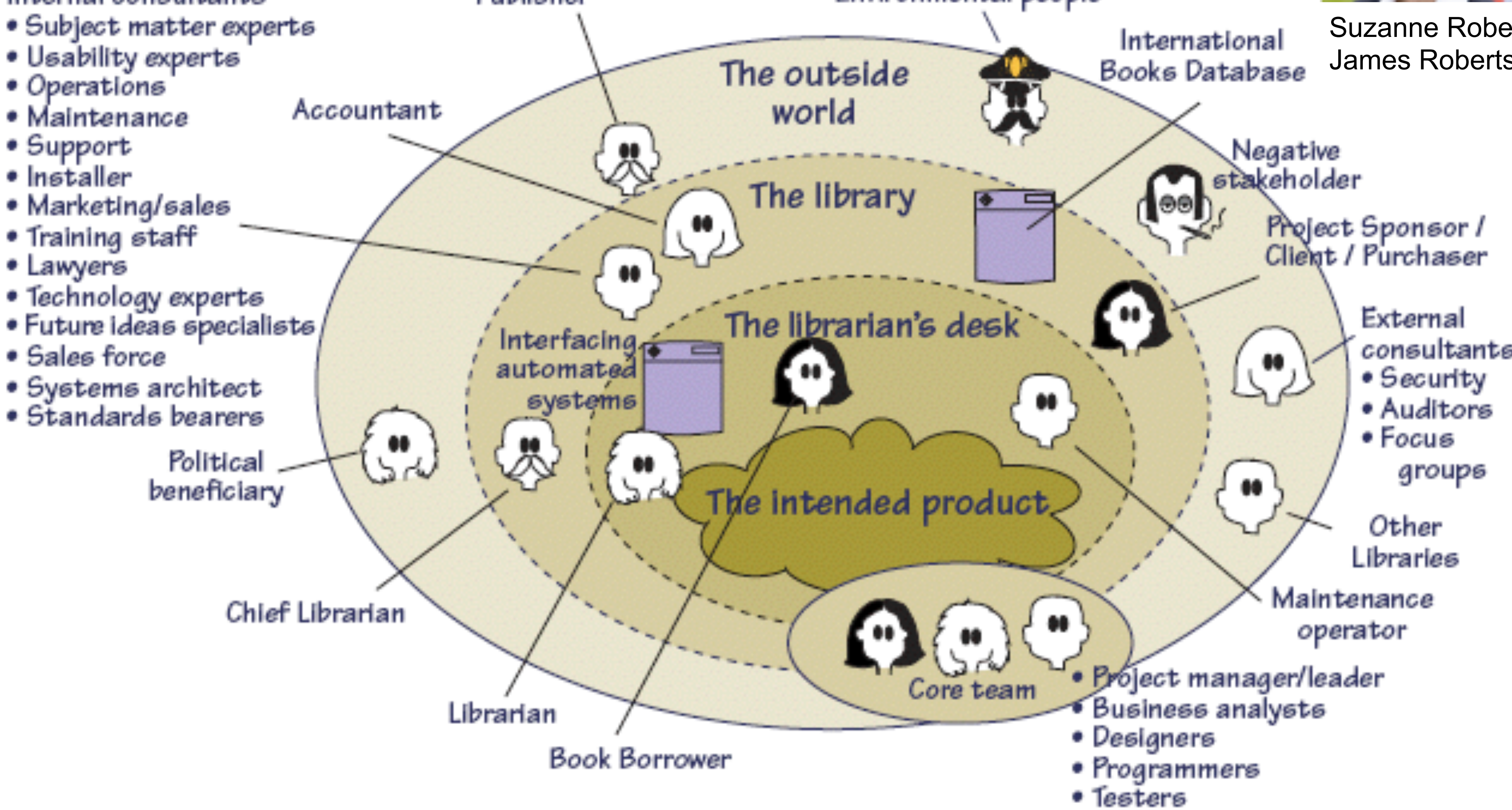
The Lean Quality Assurance Methods

- Everything 'not adding value to the Customer' is considered to be <u>waste</u>.
  - 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 (Power to the Programmers gilb.com/dl821)
  - 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"

7

Competitive Lean QA methods to Learn

1. Stakeholders Decide Qualities



Suzanne Robertson & James Robertson

Stakeholder: Concept *233 .

'Stakeholders' are:

Any person, group or thing

that can determine our systems degree of success or failure,

by having an opinion about

system performance characteristics and

system lifecycle constraints

Brodie's Stakeholder Map 2014 PhD

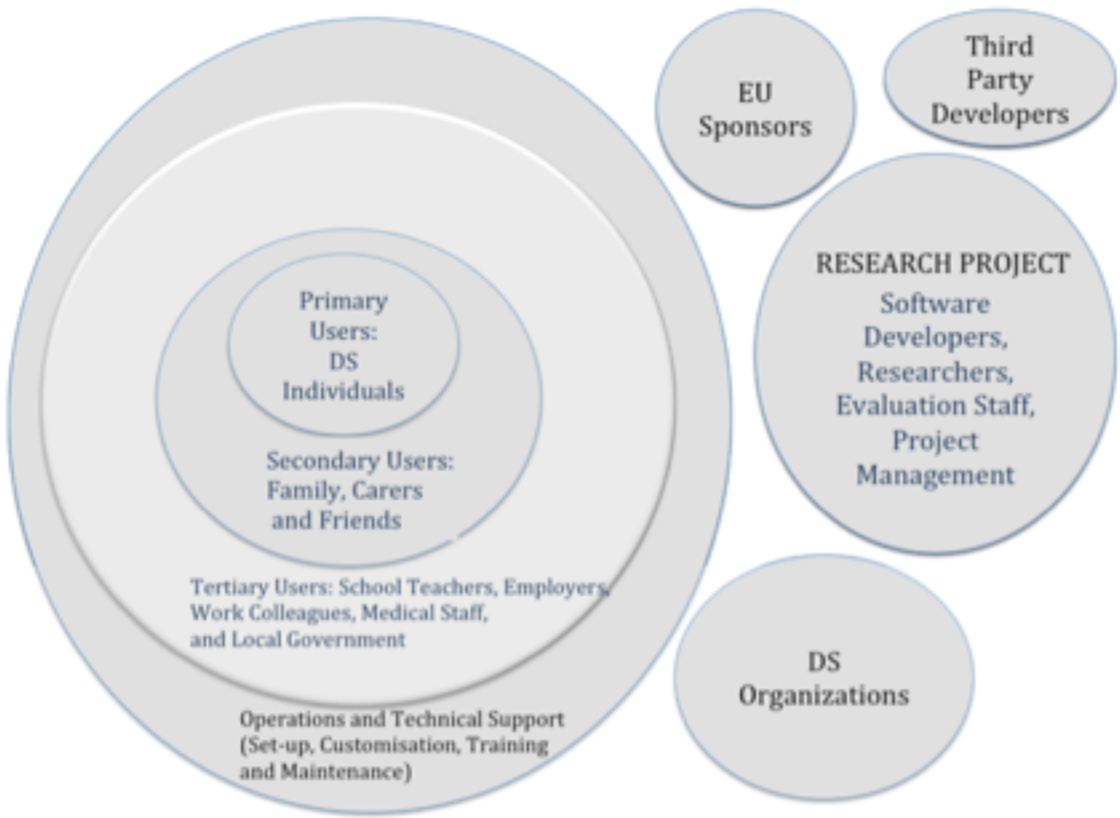


Figure 5.y: Various stakeholders

Brodie's Stakeholder Map 2014 PhD

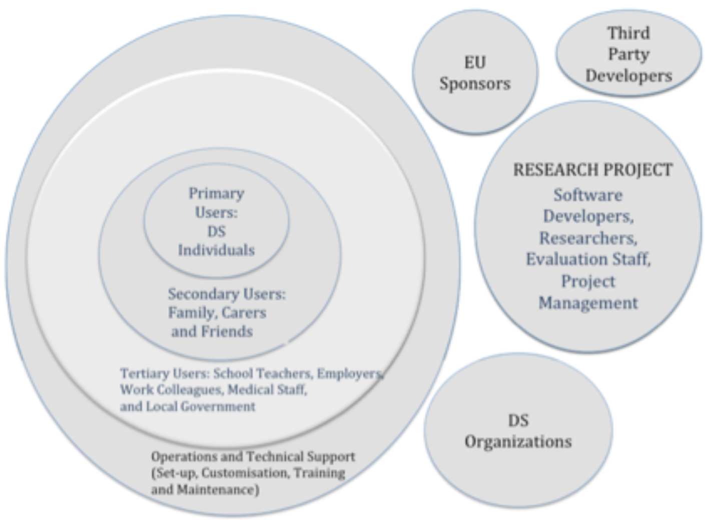


Figure 5.y: Various stakeholders

- **2. The stakeholders**
- The stakeholders identified to date include:
- Primary users (PU) - Down's Syndrome individuals
  - children
  - teenagers
  - adults (19% work and 23% attend a day centre)
- Secondary users (SU) - carers
  - Family or care home (85% + 3%)
  - Monitoring (as opposed to living alongside) (12%)
- Tertiary users (TU) – friends (Note: in their own right some could additionally be primary users)
- Tertiary users (TU) - teachers (including day centre staff) (23% attend a day centre + x% at school)
- Tertiary users (TU) - employers (19% work)
- Tertiary users (TU) - health-related staff (doctors, nurses, dentists, nutritionists, etc.)
- Down's Syndrome organizations
- Project system developers
- Technical support
- Operations
- Researchers
- EU project sponsors
- Legislation
- Third party developers
- Project management
- Research organizations
- Industrial partners.

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

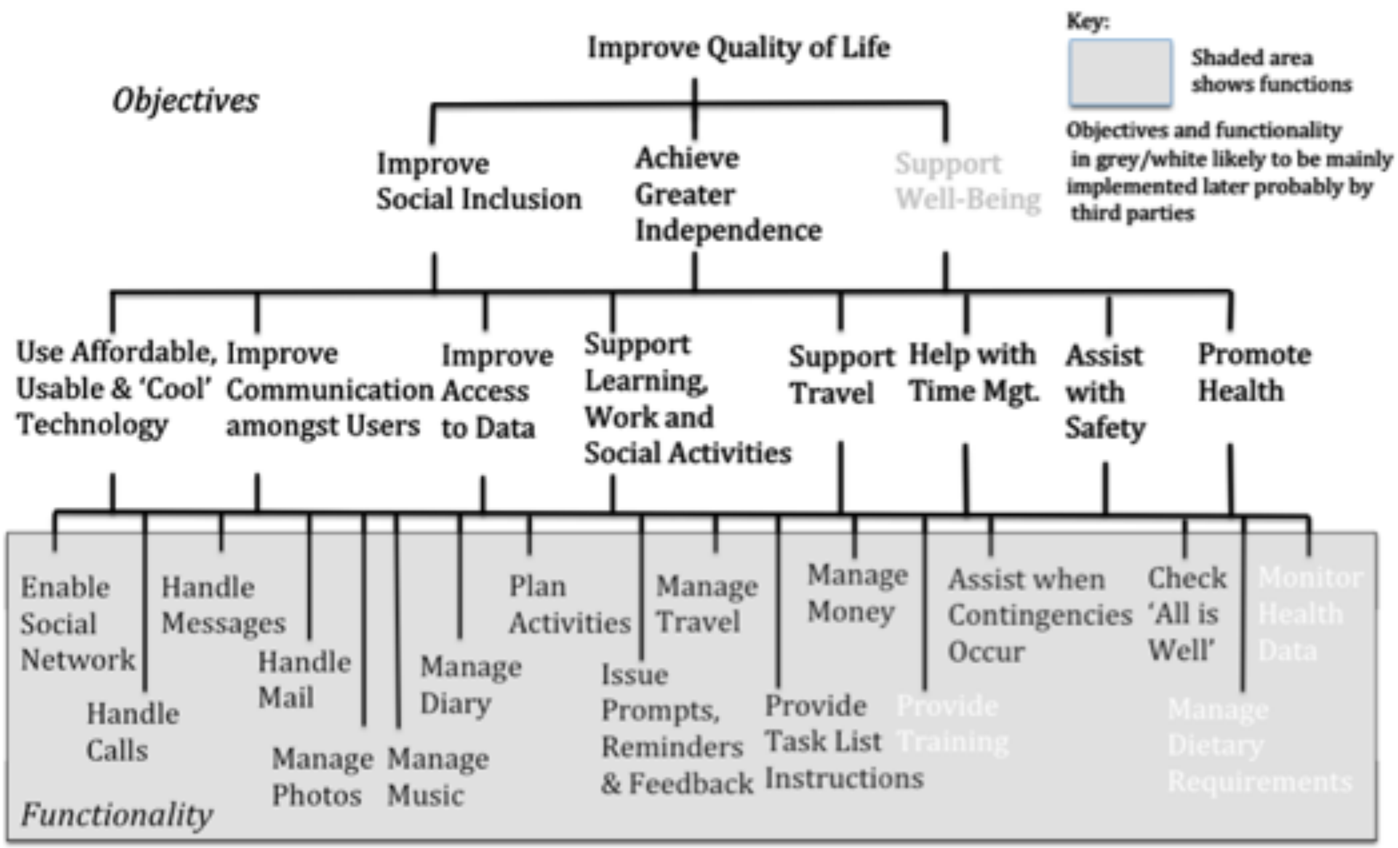


Figure 5.X: Primary user objectives and functionality

Stakeholder Interests

For example they might have an interest in

1. *Setting the objectives for a process.*
2. *Evaluating the quality of the product*
3. *Using the product or system, even indirectly*
4. *Avoiding problems for themselves as a result of our product or system.*
    - *.Being compatible with another machine or software component.*
    - *.Determining constraints on development, operation or retirement of the system.*

Project failures due to poor stakeholder engagement in US

- In contrast to these successful projects, the GAO has **regularly reported on instances of project failures due to poor stakeholder engagement**.
- Examples include:
- The Federal Emergency Management Agency (FEMA),
  - where end users were not sufficiently involved in defining requirements for the National Flood Insurance Program's insurance policy and claims management system.
  - **The program was canceled** in final end-user testing after seven years of development and a budget of $40m, forcing the agency to continue to rely on an outdated 30 year-old system.
- The Department of Homeland Security (DHS)
  - which did not allow sufficient time for stakeholder involvement in its planning and had no consistent method for identifying stakeholder roles and incorporating their feedback.
- The 2010 US Census
  - where lack of local user involvement in software testing hindered local governments' ability to accurately update address lists and maps.
- Sources:
  - {U.S. Government Accountability Office June 2011 #38}
  - {U.S. Government Accountability Office 15/09/2011 #209: 28}
  - {U.S. Government Accountability Office 14/06/2007 #210}

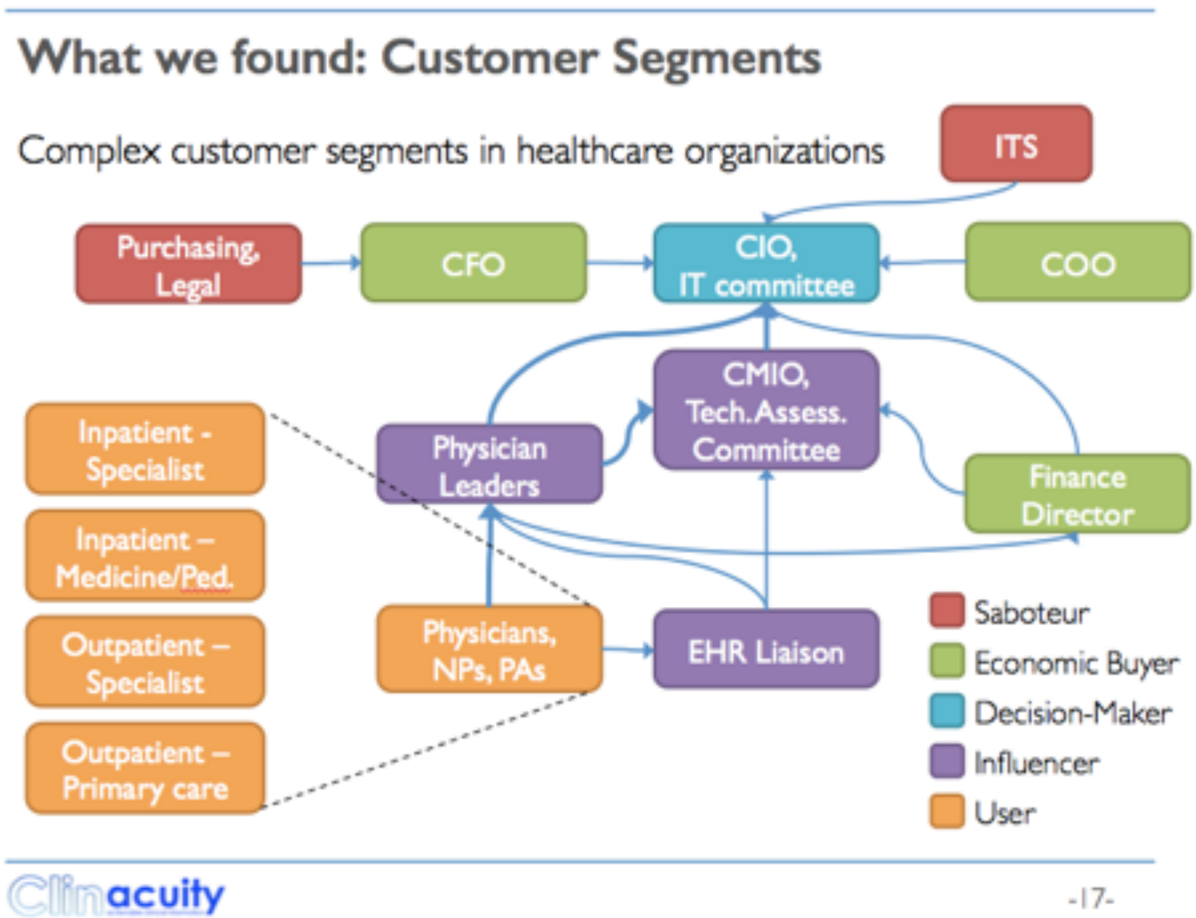
- Kilde: Wernham Agile Project Management for Government, 2012

UK Revenue and Customs 2007-2011

- In contrast, a major project by the UK Revenue and Customs had delivered 4% uptake of salaried employee tax returns over the period 2007-11
- with effective stakeholder engagement applied during a phased implementation of online services.
- Each stakeholder group **was identified and assigned a ‘champion’ to** act as a single point of contact,
- and consultative groups were set up to liaise with tax agents and industry representatives.
- Customer concerns were researched and face-to-face events were held to help small businesses and individuals understand the new processes.
- Requirements for the new services were prioritized according to stakeholder concerns.
  - For example, as a response to these concerns mandatory filing was delayed, which gave rise to the opportunity to reduce the overall budget of £373m by about 10%.
  - New requirements were proposed and implemented.
  - Example of these were free entry-level software for small businesses, and soft landings of non-mandatory solutions that allowed customers to familiarize them-selves with online filing without fear of penalties.
  - Third-party tax and accounting software developers were also identified as important stakeholders and targeted technical information was sent to them to assist them in developing compatible systems.

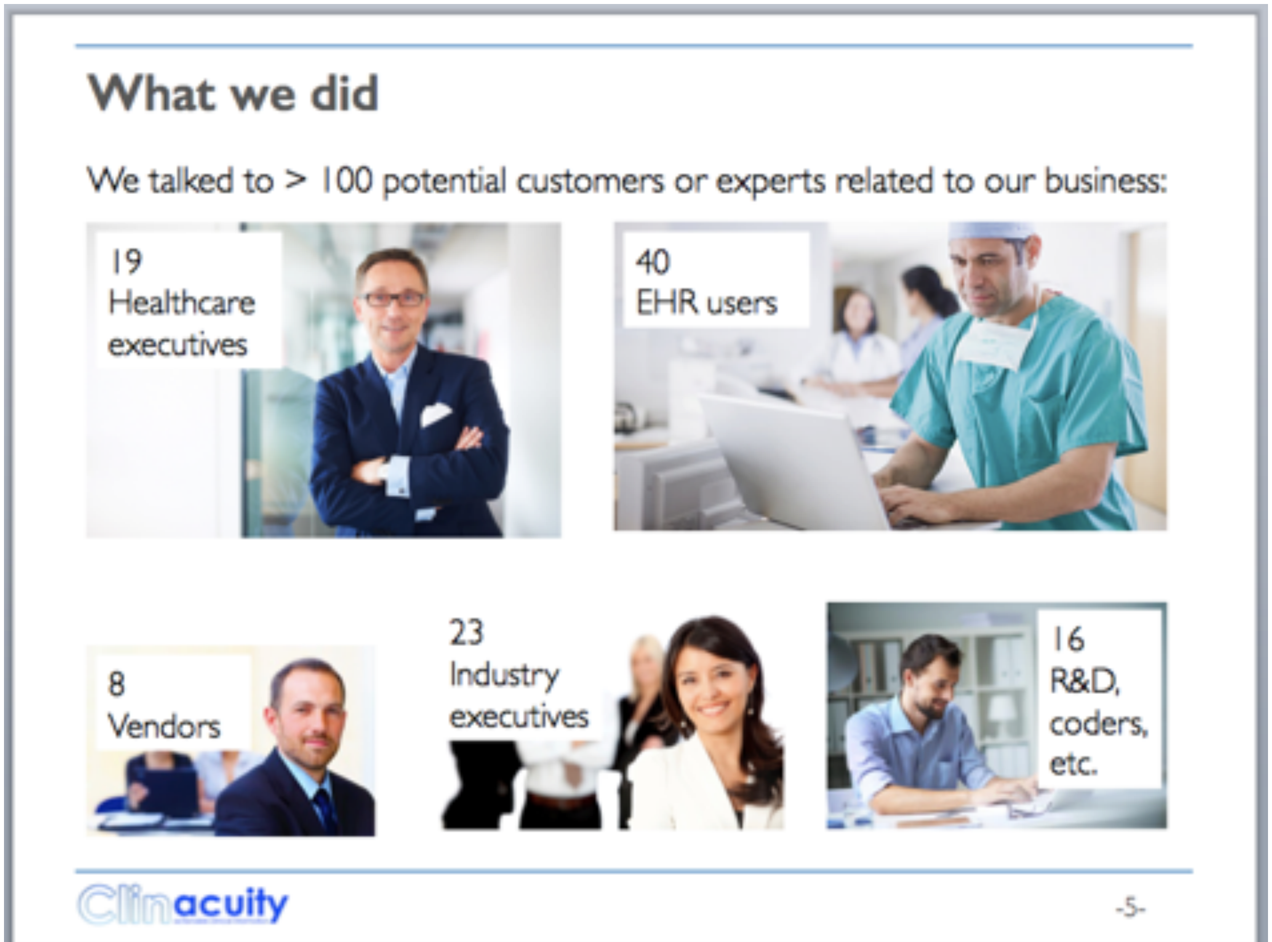
- Source: {UK NAO 09/11/2011 #207} in Wernham Agile Project Management for Government, 2012

Complex Stakeholders



http://www.slideshare.net/tomgilb1/savedfiles?s_title=clinacuity-icorpsnih-121014&user_login=sblank

Interviewing 100 Stakeholders



http://www.slideshare.net/tomgilb1/savedfiles?s_title=clinacuity-icorpsnih-121014&user_login=sblank

Modeling Multi-level Stakeholder Relations Quantitatively using IE Tables

In order to save a large IT Scrum project that failed initially, (the new system drastically killed sales!). Kai modelled the (<u>obviously</u>, ‘it failed’) ‘wicked system’. He built one Impact Estimation Table (aka Value Decision Table) for the top level of the Bring (Norwegian Post Office essentially) organization. This succeeded to resurrect the system, because it mapped the connection between technology and the higher levels of organizational objectives. The IT Development team was then instructed to focus on developing things that led to business (sales!) success.

<u>Business Goals</u>: The top management stakeholder level has problems, like *Increase Profit* and *Market Share*. Solutions have been identified (reduce *Training Costs*, and improve *User Productivity*). The expected, estimated, impact of these solutions on the (elsewhere, see Figure W4 for ‘how it looks’) *quantified* Problems, is given by the numbers estimated (later ‘measured as a result) at their intersection. For example Training Costs reduction, if the solution works as expected, promised to move us 50% of the way towards our Market Share objective (the Problem,

<u>Stakeholder Value</u>: These solutions become the the Problem at the next level. The Stakeholder level. Think of these as the 30 or so individual transport companies that had been bought and merged to form Bring. It looks like the Solution named ‘Intuitiveness’ is estimated to contribute 10% of the progress we need towards the User Productivity problem objective. All objectives are of course quantified, elsewhere.

<u>Product Val.</u>:At the third level (Product Values), ‘Find.Fast’ (one of the Stakeholder solutions, is considered an IT System objective (a problem statement).

It looks like ‘Service Guide’ is a solution that is expected to contribute 40% towards the ‘Find.Fast’ Problem solution. And ‘Service Guide’ *also* is expected to contribute 80% towards a Performance problem.

<u>Scrum Level</u>: The Service Guide solution will be developed and implemented by the Scrum Team. Hopefully its impact will be approximately as expected, and will impact several levels up towards the Business Goals.

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

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

Product Values	GUI Style Rex	Service Guide
Find.Fast	-10%	40%
Performance	50%	80 %
Resources	1 %	2 %

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

Scrum Develop
We measure improvements
Learn and Repeat

PPG's Framework for Responding to Wicked Issues

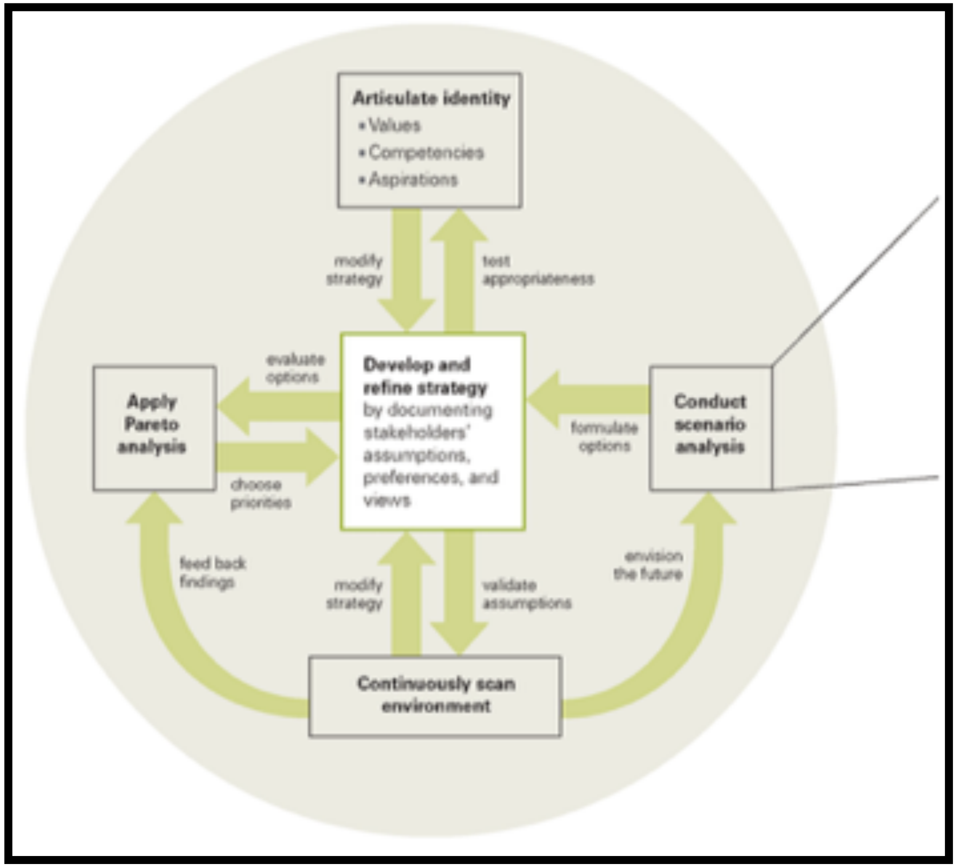
PPG Industries develops strategies

after seeking and documenting **stakeholders**' *assumptions, preferences, and alternate views*.

It evaluates the appropriateness of the strategies it draws up against its statement of identity and continually scans the environment and tests assumptions to see if it needs to change course.

The assessment of possible scenarios helps PPG formulate new options,

and its managers apply Pareto analysis to identify **a small number of actions that are likely to have a large impact.**



No Stakeholder?

- No Stakeholder: no requirements
- No requirements: nothing to do
- No requirements: nothing to test
- If you find a requirement without a Stakeholder:
  - Either the requirement isn't a requirement
  - Or, you haven't determined the Stakeholder yet
- If you don't know the Stakeholder:
  - Who's going to pay you for your work?
  - How do you know that you are doing the right thing?
  - When are you ready?

2. Quality and Value Quantification

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;

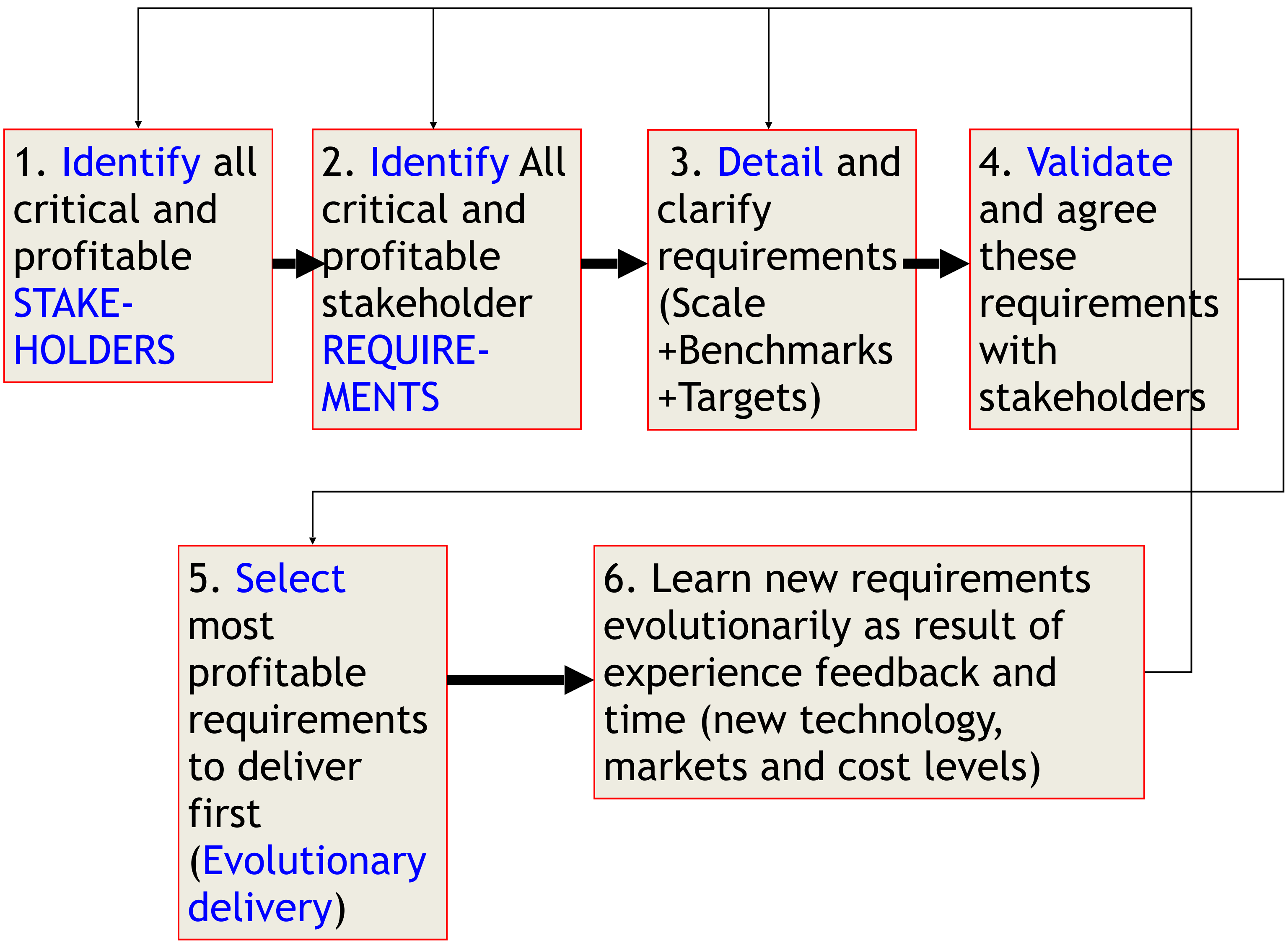
but when you **cannot measure** it,

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

your knowledge is of a meagre and unsatisfactory kind;

- Lord Kelvin, 1893

Stakeholders:
How to find out about, and confirm, their requirements



Setting Quality Goals

Usability.Learn

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

Status [today] 3 hours

Goal [next year] 10 min.

PLANGUAGE SAMPLE

EFFORT — Motivation — Design Skill — Control — Scale & Meter — Expectations [The desired rewards — Target & Benchmark → PERFORMANCE

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	[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	[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

Some potentially quantifiable Quality dimensions of Music

Examples in Planguage

Brainstormed by Steve F. and Rachel D. At lunch

- In tune
- Applause
- Moving
- Encores
- Repeat Gigs
- Busking Hat Collection
- MRI Brain Scan
- Downloads
- Utube Reviews
- Royalties
- … (many more!!)

- **Music.Moving:**
- **Type: primary music quality attribute**
- **Ambition Level: the majority of listeners feel moved to tears or strong physical emotional reactions.**
- **Scale: the % of defined [Listeners] hearing defined [Music] under defined [Environments] who reports a defined [Emotion] at a defined [Strength]**
- **Goal [1st UK Release, Music = Hip Hop, Environment = Itunes, Emotion = {Tears, Sadness}, Strength = Powerful] 50% ± 20% ?**

How to Quantify any Qualitative Requirement

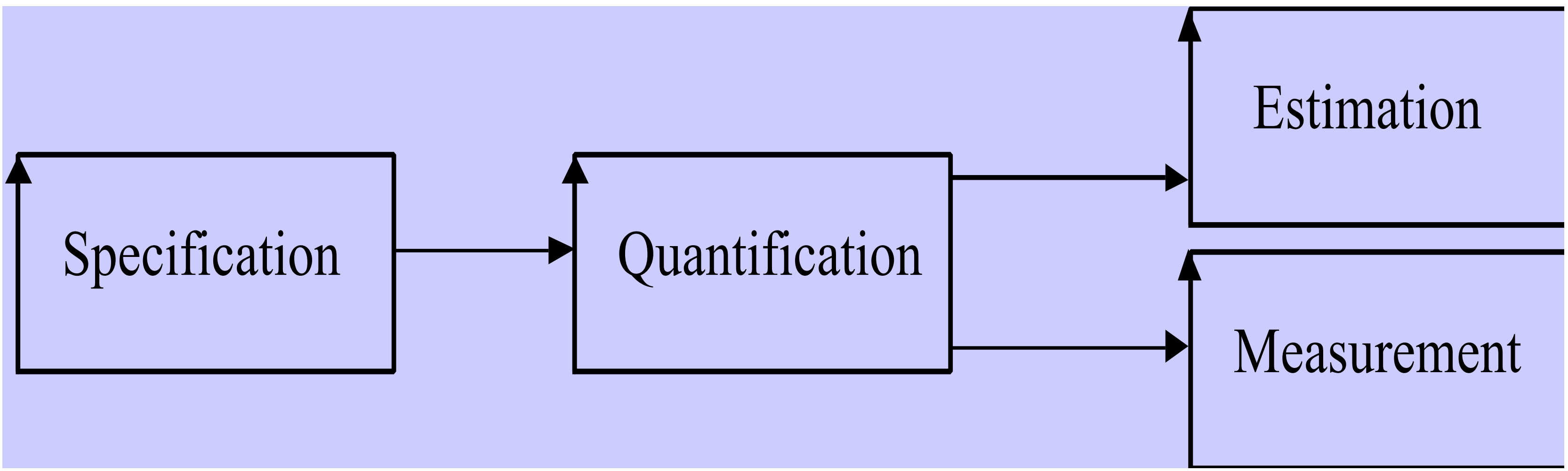


Diagram from 'Competitive Engineering.' book.

Quality Quantification Methods #1

- Common Sense, Domain Knowledge
  - Decompose "until quantification becomes obvious".
  - Then use Planguage specification:
    - **Scale**: define a measurement scale
    - **Meter**: define a test or process for measuring on the scale
    - **Past**: define benchmarks, old system, competitors on the scale
    - **Goal**: define a committed level of future stakeholder quality, on your scale.

Quality Quantification Methods #2, **Look it up in a book**

Maintainability:
Type: Complex Quality Requirement.
Includes: {Problem Recognition, Administrative Delay, Tool Collection, Problem Analysis, Change Specification, Quality Control, Modification Implementation, Modification Testing {Unit Testing, Integration Testing, Beta Testing, System Testing}, Recovery}.

Problem Recognition:
Scale: Clock hours from defined [Fault Occurrence: Default: Bug occurs in any use or test of system] until fault officially recognized by defined [Recognition Act: Default: Fault is logged electronically].
Administrative Delay:
Scale: Clock hours from defined [Recognition Act] until defined [Correction Action] initiated and assigned to a defined [Maintenance Instance].
Tool Collection:
Scale: Clock hours for defined [Maintenance Instance: Default: Whoever is assigned] to acquire all defined [Tools: Default: all systems and information necessary to analyze, correct and quality control the correction].
Problem Analysis:
Scale: Clock time for the assigned defined [Maintenance Instance] to analyze the fault symptoms and be able to begin to formulate a correction hypothesis.
Change Specification:
Scale: Clock hours needed by defined [Maintenance Instance] to fully and correctly describe the necessary correction actions, according to current applicable standards for this.
Note: This includes any additional time for corrections after quality control and tests.
Quality Control:
Scale: Clock hours for quality control of the correction hypothesis (against relevant standards).
Modification Implementation:
Scale: Clock hours to carry out the correction activity as planned. "Includes any necessary corrections as a result of quality control or testing."
Modification Testing:
- **Unit Testing**:
  Scale: Clock hours to carry out defined [Unit Test] for the fault correction.
- **Integration Testing**:
  Scale: Clock hours to carry out defined [Integration Test] for the fault correction.
- **Beta Testing**:
  Scale: Clock hours to carry out defined [Beta Test] for the fault correction before official release of the correction is permitted.
- **System Testing**:
  Scale: Clock hours to carry out defined [System Test] for the fault correction.

Recovery:
Scale: Clock hours for defined [User Type] to return system to the state it was in prior to the fault and, to a state ready to continue with work.

Source: The above is an extension of some basic ideas from Ireson, Editor, Reliability Handbook, *McGraw Hill, 1966* (Ireson 1966).

Chapter
5

SCALES OF MEASURE

How to Quantify

Quality Quantification Methods #2, Look it up in a book

Maintainability:
Type: Complex Quality Requirement.
Includes: {Problem Recognition, Administrative Delay, Tool Collection, Problem Analysis, Change Specification, Quality Control, Modification Implementation, Modification Testing {Unit Testing, Integration Testing, Beta Testing, System Testing}, Recovery}.

Problem Recognition:
Scale: C
system]
electron
Adminis
Scale: C
assigned
Tool Co
Scale: C
acquire
and qua
Problem
Scale: C
toms an
Change
Scale: C
the nece
Note: Th
Quality
Scale: C
Modific
Scale: C
correctio
Modific
Unit T
Scale
Integr
Scale
Beta
Scale
releas
Syste
Scale
Recove
Scale: C
fault and, to a state ready to continue with work.

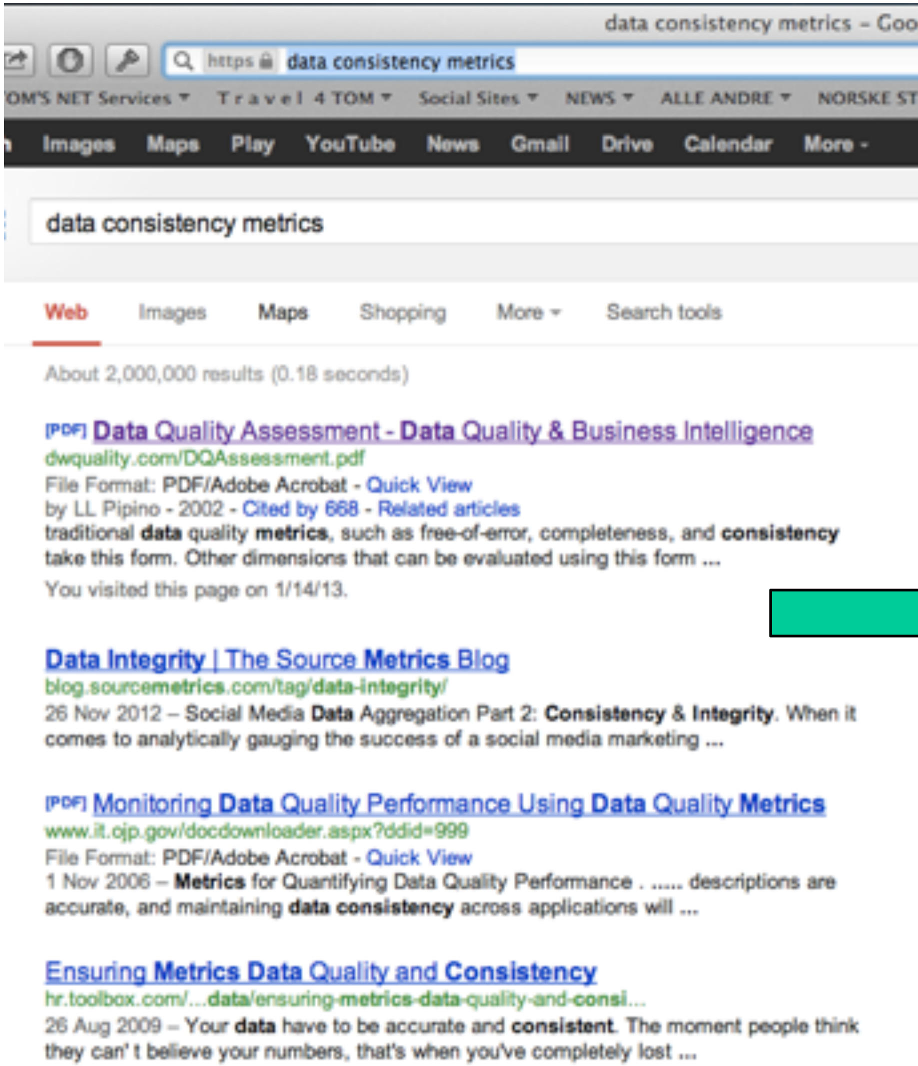
Tool Collection:

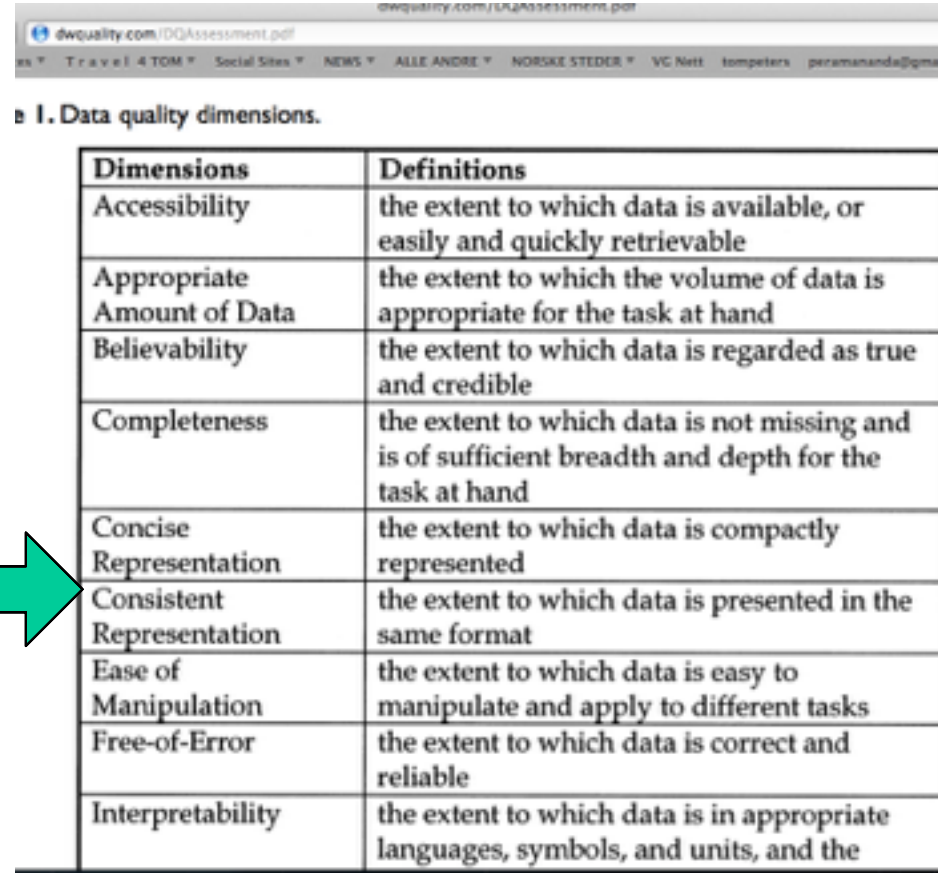
Scale: Clock hours for defined [Maintenance Instance: Default: Whoever is assigned] to acquire all defined [Tools: Default: all systems and information necessary to analyze, correct and quality control the correction].

Source: The above is an extension of some basic ideas from Ireson, Editor, Reliability Handbook, McGraw Hill, 1966 (Ireson 1966).

E N G I N E E R I N G
A HANDBOOK FOR SYSTEMS, REQUIREMENTS AND

Quality Quantification Methods #**3**, Google It



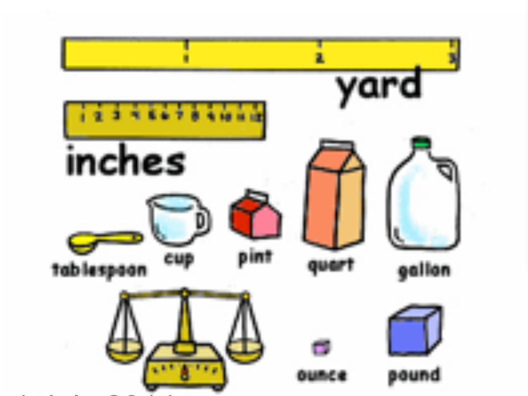


e 1. Data quality dimensions.

Dimensions	Definitions
Accessibility	the extent to which data is available, or easily and quickly retrievable
Appropriate Amount of Data	the extent to which the volume of data is appropriate for the task at hand
Believability	the extent to which data is regarded as true and credible
Completeness	the extent to which data is not missing and is of sufficient breadth and depth for the task at hand
Concise Representation	the extent to which data is compactly represented
Consistent Representation	the extent to which data is presented in the same format
Ease of Manipulation	the extent to which data is easy to manipulate and apply to different tasks
Free-of-Error	the extent to which data is correct and reliable
Interpretability	the extent to which data is in appropriate languages, symbols, and units, and the

Summary of Top ‘8’ Project Objectives

- **Defined** Scales of Measure:
  - Demands ***comparative* thinking**.
  - Leads to requirements that are unambiguously **clear**
  - Helps Team be **Aligned** with the Business



Real Example of ***Lack*** of Scales

*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 rewrite in example below)*

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

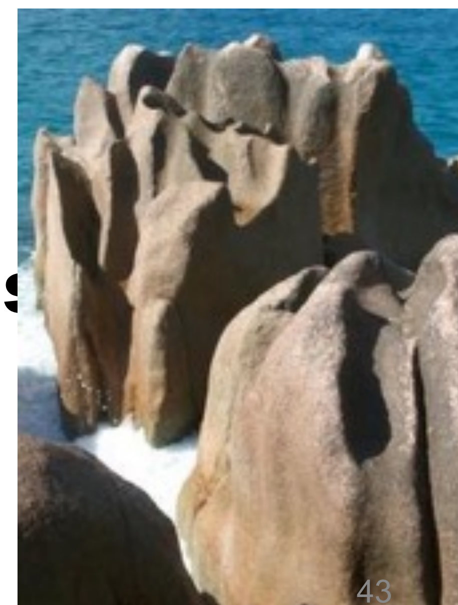
This lack of clarity cost them $100,000, 000

"Rock Solid Robustness" Defined Clearly in Planguage over a beer

<u>Rock Solid Robustness</u>:

Type: Complex Product Quality Requirement.

Includes: { Software Downtime, Restore Speed, Testability, Fault Prevention Capability, Fault Isolation Capability, Fault Analysis Capability, Hardware Debugging Capability}.



<u>Software Downtime</u>:

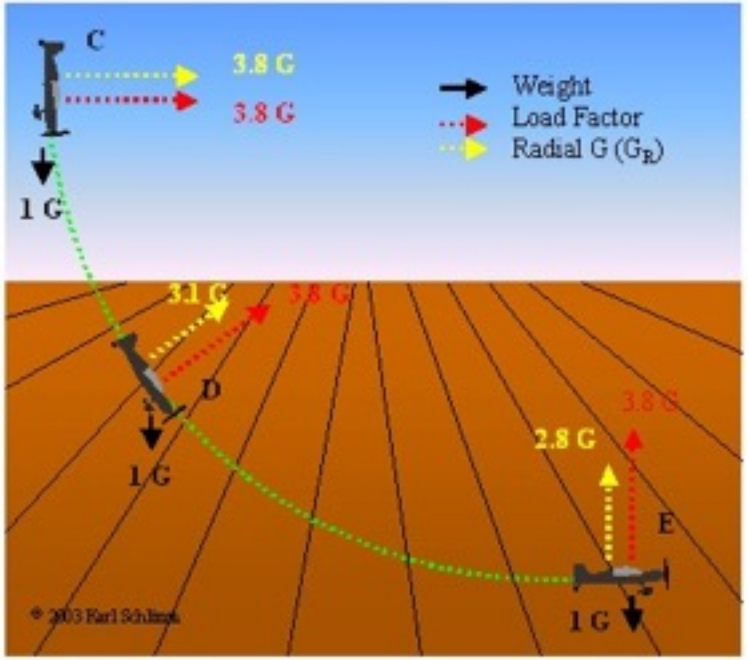
<u>Software Downtime</u>:
Type: Software Quality Requirement.
Ambition: *to have minimal downtime*
due to software failures <- HFA 6.1
Issue*: does this not imply that there is a system wide downtime requirement?*

Scale: <mean time between forced restarts for defined [Activity], for a defined [Intensity].>

Fail [Any Release or Evo Step, Activity = Recompute, Intensity = Peak Level] **14 days** <- HFA 6.1.1

Goal [By 2008?, Activity = Data Acquisition, Intensity = Lowest level] **: 300 days** ??
Stretch: **600 days**

Restore Speed:



Restore Speed:
Type: Software Quality Requirement.

Ambition: Should an error occur (or the user otherwise desire to do so), Horizon shall be abl to restore the system to a
previously saved state in less than 10 minutes. <-6.1.2 HFA.

Scale: Duration from Initiation of Restore to Complete and verified state of a defined [Previous: Default = Immediately Previous]] saved state.

Initiation: defined as {Operator Initiation, System Initiation, ?}. Default = Any.

Goal [Initial and all subsequent released and Evo steps] 1 minute?

Fail [Initial and all subsequent released and Evo steps] 10 minutes. <- 6.1.2 HFA

Catastrophe: 100 minutes.

<u>Testability</u>:

Type: Software Quality Requirement.

Version: 20 Oct 2006-10-20

Status: Demo draft,

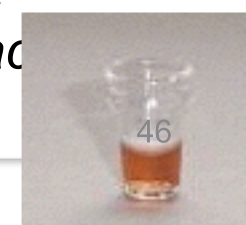
Stakeholder: {Operator, Tester}.

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

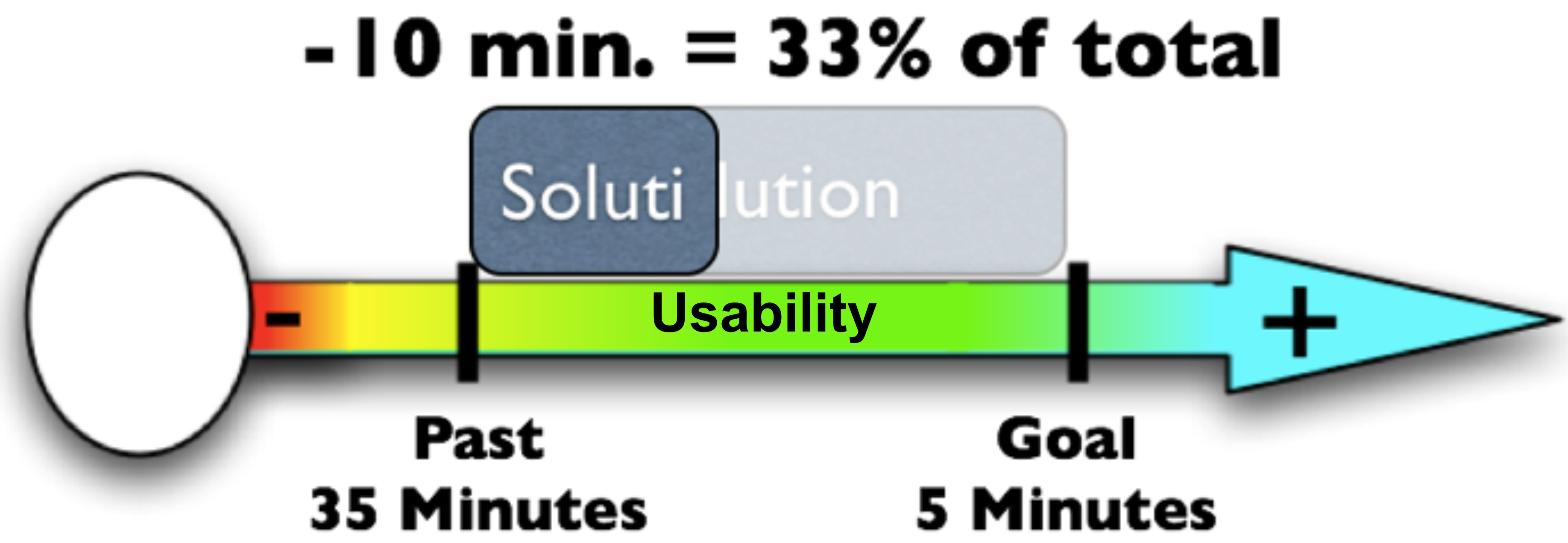
Scale: the duration of a defined [Volume] of testing, or a defined [Type], by a defined [Skill Level] of system operator, under defined [Operating Conditions].

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.

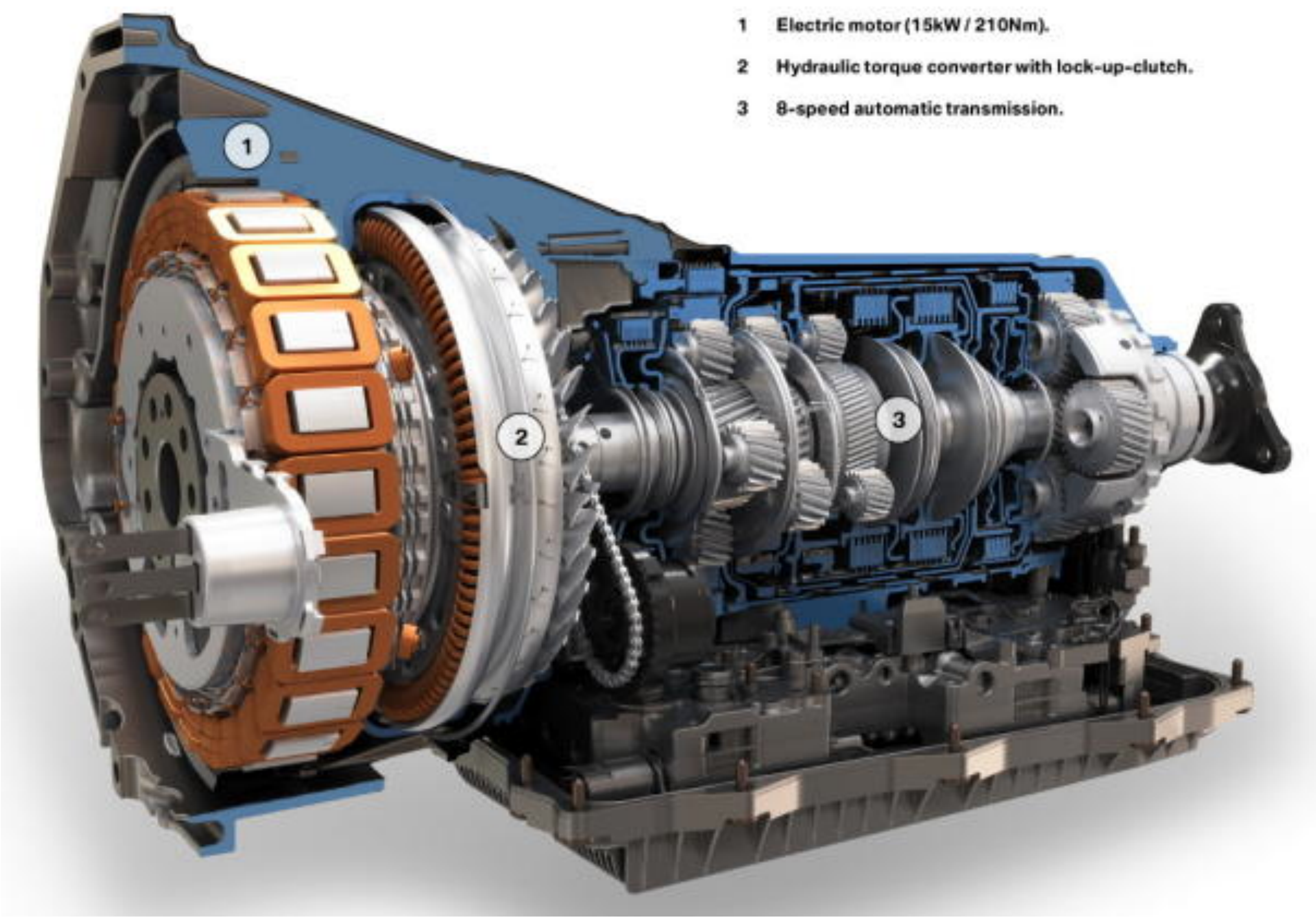
<u>Design Hypothesis</u>*: Tool Simulators, Reverse Cracking Tool, Generation of simulated telemetry frames entirely in software, Application specific sophistication, for drilling – recorded mode simulation by playing bac dump file, Application test harness console <-6.2.1 HFA*



3. Assuring that Designs give Qualities



Design Quality In



A FEW REASONS
WHY THE
ROLLS - ROYCE
IS THE
BEST SIX - CYLINDER
CAR IN THE WORLD.

Because of its

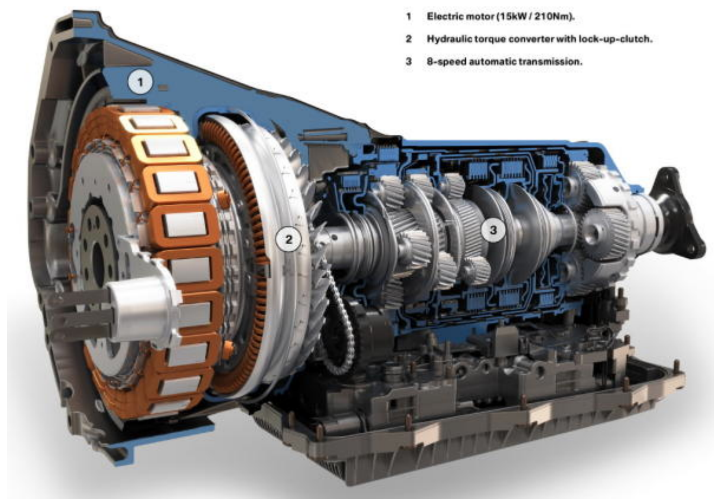
(1) Flexibility.
(2) Lightness and cheapness in tyres.
(3) Reliability.
(4) Silence.
(5) Efficiency and cheapness in upkeep.
(6) Safety—brakes, steering gear, etc.
(7) Ease of manipulation, lightness of steering, clutch operation, etc.

You don't get quality by testing it in

but by ‘Engineering’ Quality In

Work hours

$ € Kr.



Reliability

Performance

Security

Usability

Maintenance

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	$\frac{\text{Benefits}}{\text{Cost}}$	$\frac{\text{Benefits}}{\text{Cost}}$	$\frac{\text{Benefits}}{\text{Cost}}$	

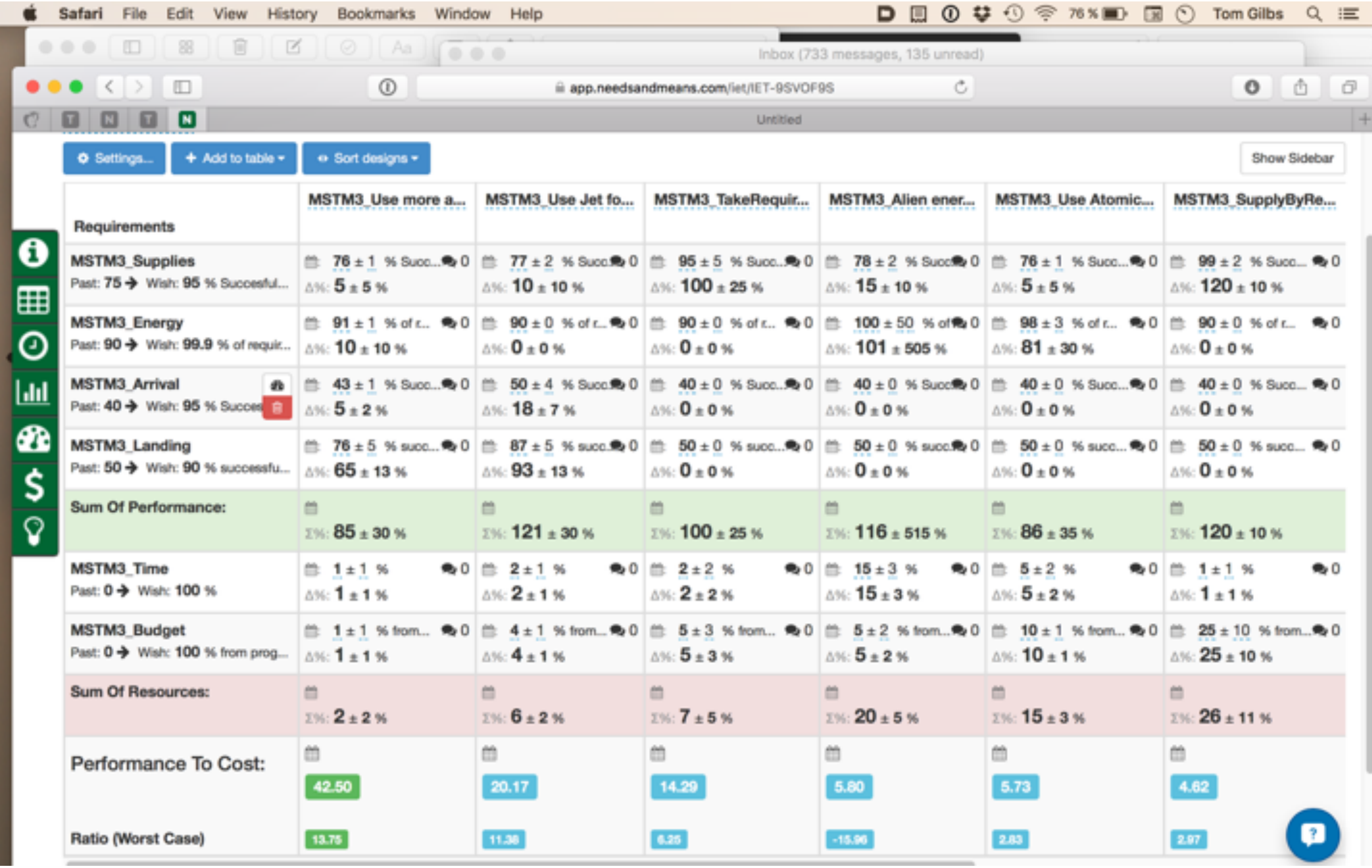
Designing to meet Quality within Costs

Design Ideas

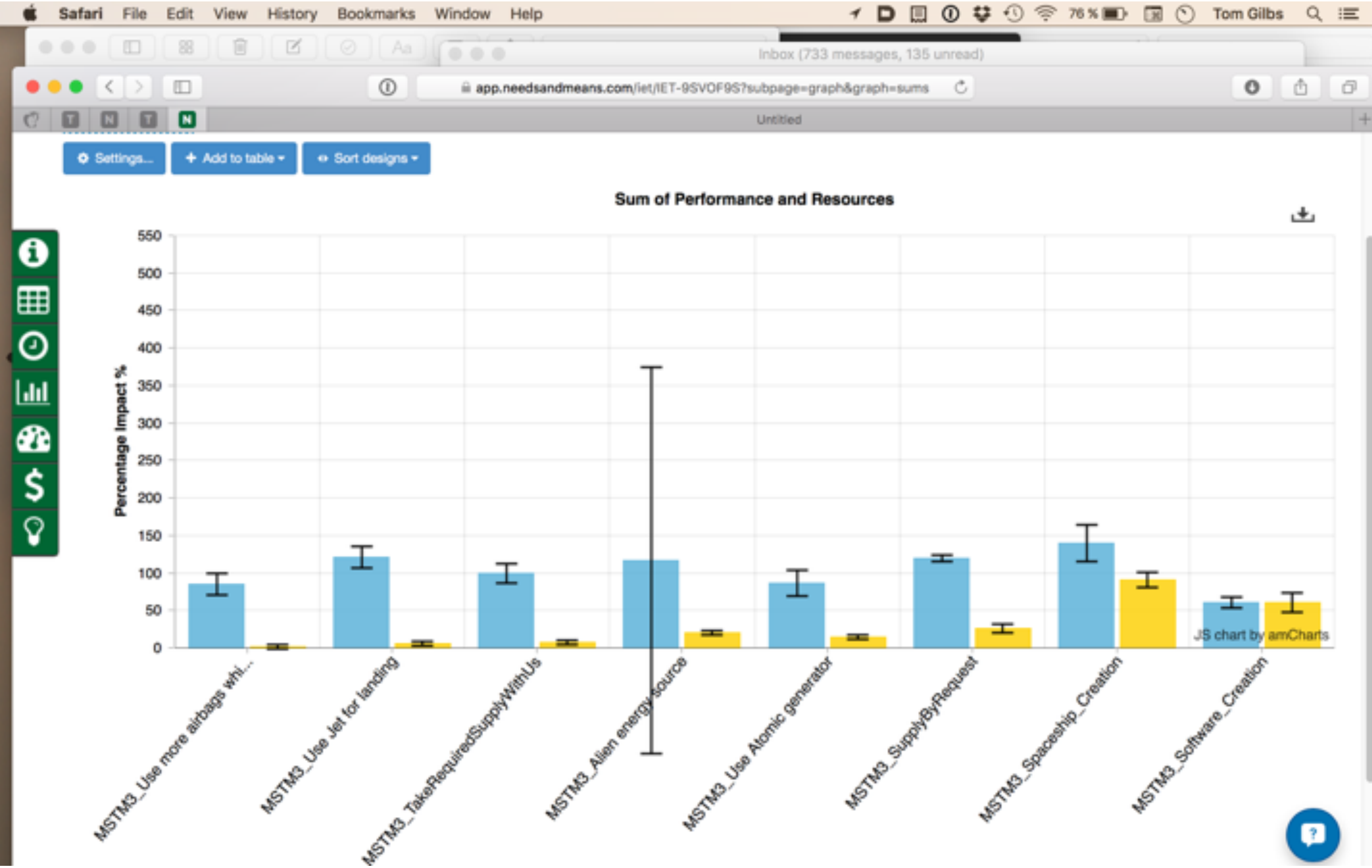
Qualities

€ $

				Estimated Impact		Estimated Impact		Estimated Impact		Estimated Impact	
Prooduct 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		5,55



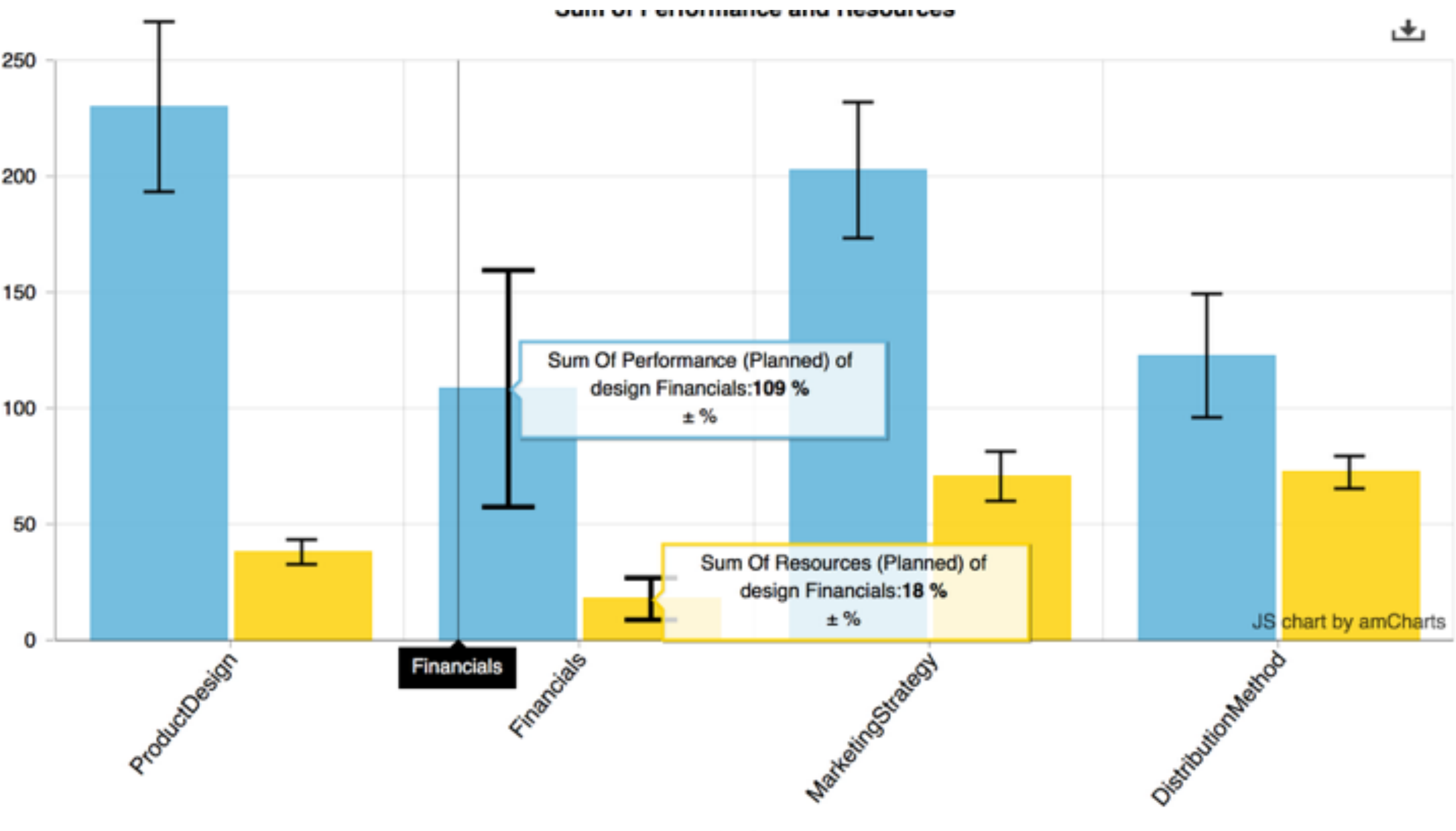
Requirements	MSTM3_Use more a...	MSTM3_Use Jet fo...	MSTM3_TakeRequir...	MSTM3_Alien ener...	MSTM3_Use Atomic...	MSTM3_SupplyByRe...
MSTM3_Supplies Past: 75 → Wish: 95 % Succesful...	76 ± 1 % Succ... 0 Δ%: 5 ± 5 %	77 ± 2 % Succ 0 Δ%: 10 ± 10 %	95 ± 5 % Succ... 0 Δ%: 100 ± 25 %	78 ± 2 % Succ 0 Δ%: 15 ± 10 %	76 ± 1 % Succ... 0 Δ%: 5 ± 5 %	99 ± 2 % Succ... 0 Δ%: 120 ± 10 %
MSTM3_Energy Past: 90 → Wish: 99.9 % of requir...	91 ± 1 % of r... 0 Δ%: 10 ± 10 %	90 ± 0 % of r... 0 Δ%: 0 ± 0 %	90 ± 0 % of r... 0 Δ%: 0 ± 0 %	100 ± 50 % of 0 Δ%: 101 ± 505 %	98 ± 3 % of r... 0 Δ%: 81 ± 30 %	90 ± 0 % of r... 0 Δ%: 0 ± 0 %
MSTM3_Arrival Past: 40 → Wish: 95 % Succes	43 ± 1 % Succ... 0 Δ%: 5 ± 2 %	50 ± 4 % Succ 0 Δ%: 18 ± 7 %	40 ± 0 % Succ... 0 Δ%: 0 ± 0 %	40 ± 0 % Succ 0 Δ%: 0 ± 0 %	40 ± 0 % Succ... 0 Δ%: 0 ± 0 %	40 ± 0 % Succ... 0 Δ%: 0 ± 0 %
MSTM3_Landing Past: 50 → Wish: 90 % successfu...	76 ± 5 % succ... 0 Δ%: 65 ± 13 %	87 ± 5 % succ 0 Δ%: 93 ± 13 %	50 ± 0 % succ... 0 Δ%: 0 ± 0 %	50 ± 0 % succ 0 Δ%: 0 ± 0 %	50 ± 0 % succ... 0 Δ%: 0 ± 0 %	50 ± 0 % succ... 0 Δ%: 0 ± 0 %
Sum Of Performance:	Σ%: 85 ± 30 %	Σ%: 121 ± 30 %	Σ%: 100 ± 25 %	Σ%: 116 ± 515 %	Σ%: 86 ± 35 %	Σ%: 120 ± 10 %
MSTM3_Time Past: 0 → Wish: 100 %	1 ± 1 % 0 Δ%: 1 ± 1 %	2 ± 1 % 0 Δ%: 2 ± 1 %	2 ± 2 % 0 Δ%: 2 ± 2 %	15 ± 3 % 0 Δ%: 15 ± 3 %	5 ± 2 % 0 Δ%: 5 ± 2 %	1 ± 1 % 0 Δ%: 1 ± 1 %
MSTM3_Budget Past: 0 → Wish: 100 % from prog...	1 ± 1 % from... 0 Δ%: 1 ± 1 %	4 ± 1 % from... 0 Δ%: 4 ± 1 %	5 ± 3 % from... 0 Δ%: 5 ± 3 %	5 ± 2 % from... 0 Δ%: 5 ± 2 %	10 ± 1 % from... 0 Δ%: 10 ± 1 %	25 ± 10 % from... 0 Δ%: 25 ± 10 %
Sum Of Resources:	Σ%: 2 ± 2 %	Σ%: 6 ± 2 %	Σ%: 7 ± 5 %	Σ%: 20 ± 5 %	Σ%: 15 ± 3 %	Σ%: 26 ± 11 %
Performance To Cost:	42.50	20.17	14.29	5.80	5.73	4.62
Ratio (Worst Case)	13.75	11.38	6.25	-15.96	2.83	2.97


Safari File Edit View History Bookmarks Window Help
76 %
Tom Gilbs
Inbox (733 messages, 135 unread)
app.needsandmeans.com/iet/IET-9SVDF9S?subpage=graph&graph=sums
Untitled
Settings...
Add to table
Sort designs
Sum of Performance and Resources
Percentage Impact %
550
500
450
400
350
300
250
200
150
100
50
0
JS chart by amCharts
MSTM3_Use more airbags whi...
MSTM3_Use Jet for landing
MSTM3_TakeRequiredSupplyWithUs
MSTM3_Alien energy source
MSTM3_Use Atomic generator
MSTM3_SupplyByRequest
MSTM3_Spaceship_Creation
MSTM3_Software_Creation

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	ΣΔ%: 160 ± 26 %
Millionaire Past: 1 → Wish: 1000000 $	450000 ± 150000 0 Δ%: 45 ± 15 % 45	400000 ± 100000 0 Δ%: 40 ± 10 % 85	100000 ± 50000 0 Δ%: 10 ± 5 % 95	200000 ± 100000 0 Δ%: 20 ± 10 % 115	ΣΔ%: 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	ΣΔ%: 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	ΣΔ%: 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	ΣΔ%: 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	ΣΔ%: 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	ΣΔ%: 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	



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
Financials
ProductDesign
Financials
MarketingStrategy
DistributionMethod

4. Measure Quality Levels in Specifications with Inspection

Defect Rates
in 2003 Pilot Financial Shop, London, Gilb Client
Spec QC/Extreme Inspection + Planguage Requirements

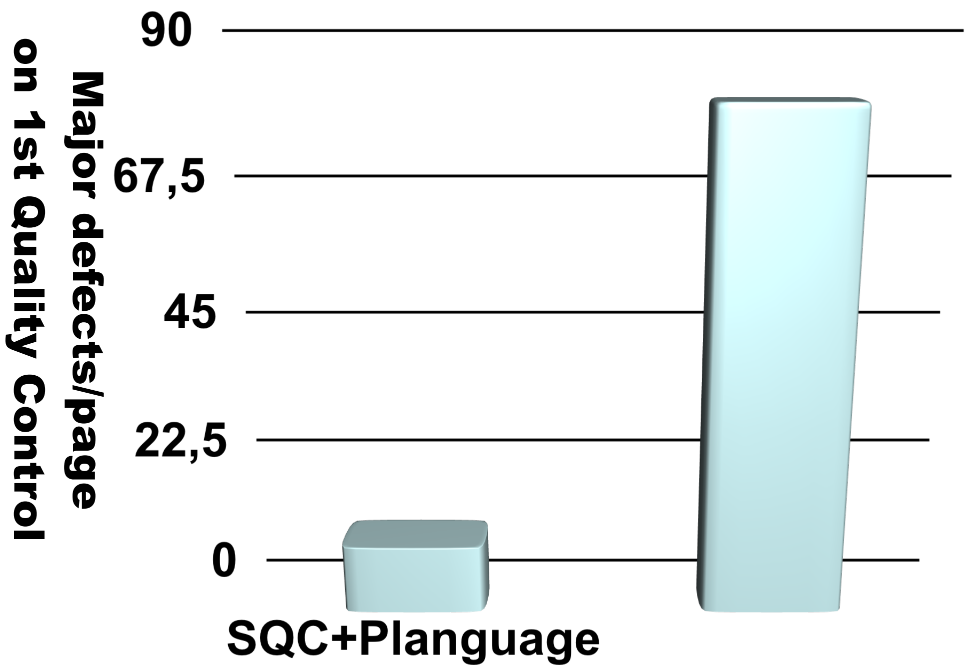
Across 18 DV (DeVelopment) Projects using the new requirements method, the average major defect rate on first inspection is 11.2.

4 of the 18 DV projects were re-inspected after failing to meet the Exit Criteria of 10 major defects per page.

A sample of 6 DV projects with requirements in the 'old' format were tested against the rules set of:

- The requirement is uniquely identifiable
- All stakeholders are identified.
- The content of the requirement is 'clear and unambiguous'
- A practical test can be applied to validate it's delivery.

The average major defect rate in this sample was 80.4.



An Advanced Example

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

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

Case:
Real Inspection

- of System Requirements Specification (SRS) of 82 pages for a major US corporation.

This presentation

shows

how we carried out a short

specification quality control

process

with senior/middle managers.

The purpose is to
make managers aware
that they play a key-role
in creating projects
delays
by approving poor
quality of requirements
specifications.

The results shown in this real-life example successfully predicted a project delay of at least 2 calendar years.

Poor quality marketing requirements documents prove time and again to be

a good predictor of project delays.

- The clue is that
- requirements documents
  - with a high defect density
  - are an indicator of
  - a truly *unprofessional* engineering

culture.

Framework

- Demonstration of power of Inspection
  - 8 Managers
  - 2 hours
  - 4 real requirements specifications offered ,
    - 1 used

We Introduced best practice **Rules** for Requirements

- 1. **Unambiguous** to intended Readership
- 2. **Clear** enough to test.
- 3. **No** unintentional **Design**

We Explained the definition of <u>Defect</u>

- A **Specification Defect** is a *violation of a Specifciation Rule (a 'standard')*
  - Note: If there are 10 ambiguous terms in a *single* requirement
    - then there are 10 defects!

Explain the definition of Major defect

- **<u>Major</u>:**
  - a Defect that *potentially*

costs more

- to find and fix
- later in the development process
- than it would cost *now*.
- We need to get rid of it NOW!

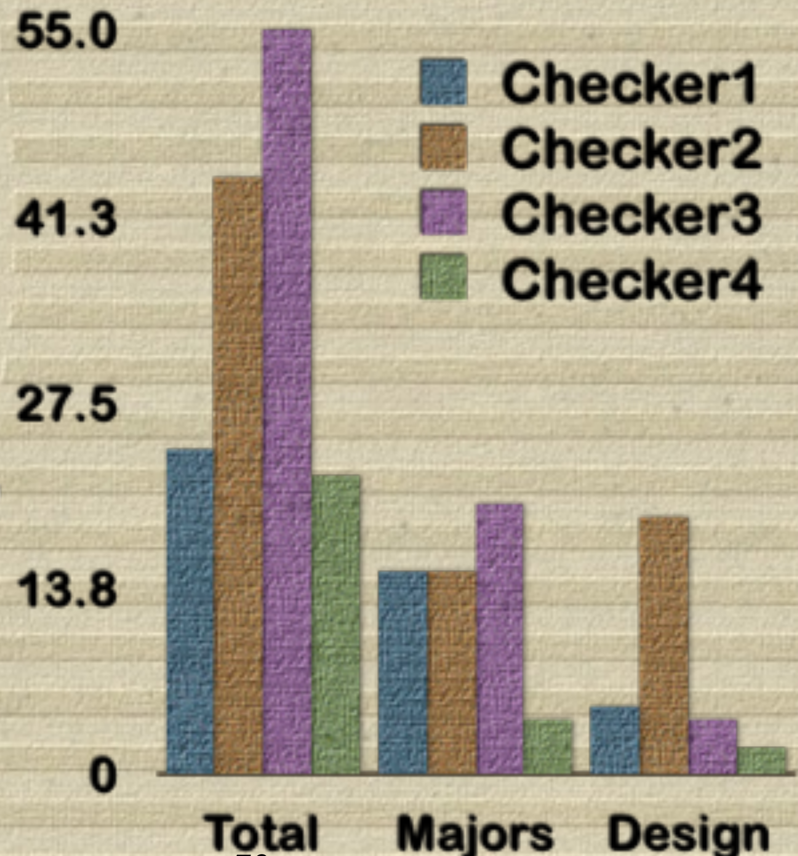
Agree with Management on Exit level

- **Exit Conditions:** (when Requirements can go forward to Design, Test etc with little risk)
  - **Maximum 1 Major Defect/ (Logical) Page**
  - **Logical Page = 300** Non commentary words.

the Job

- You have up to 30 minutes
  - check 1 sample requirements page (from an 82 page document)
- Count all *potential* Rule Violations
  - = Defects
- *Classify* Defects as Major or minor

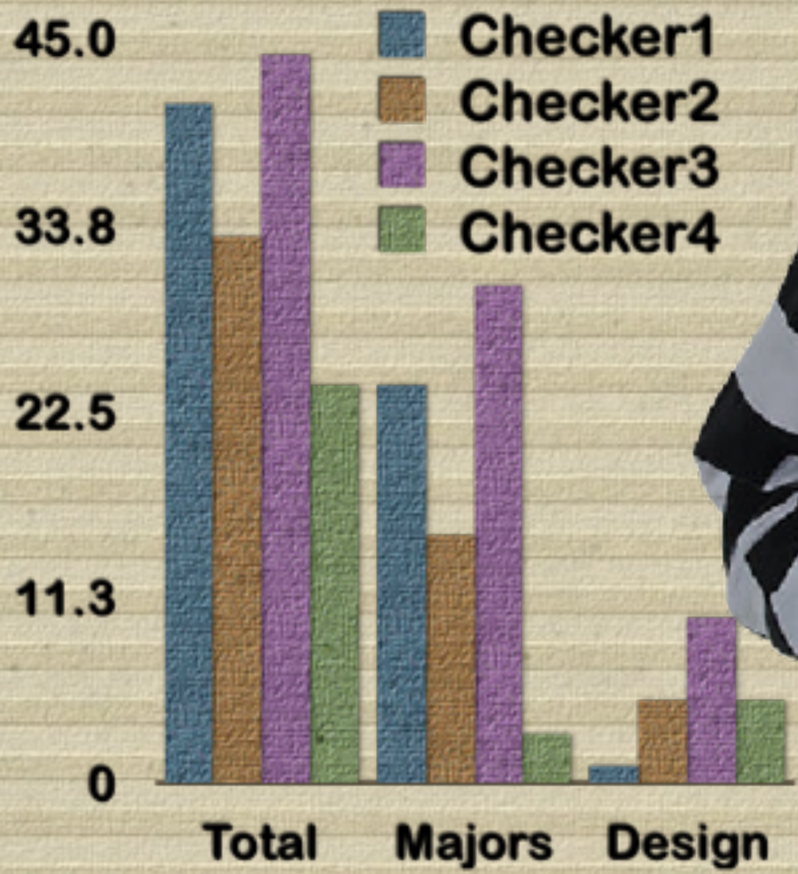
Report Page 81

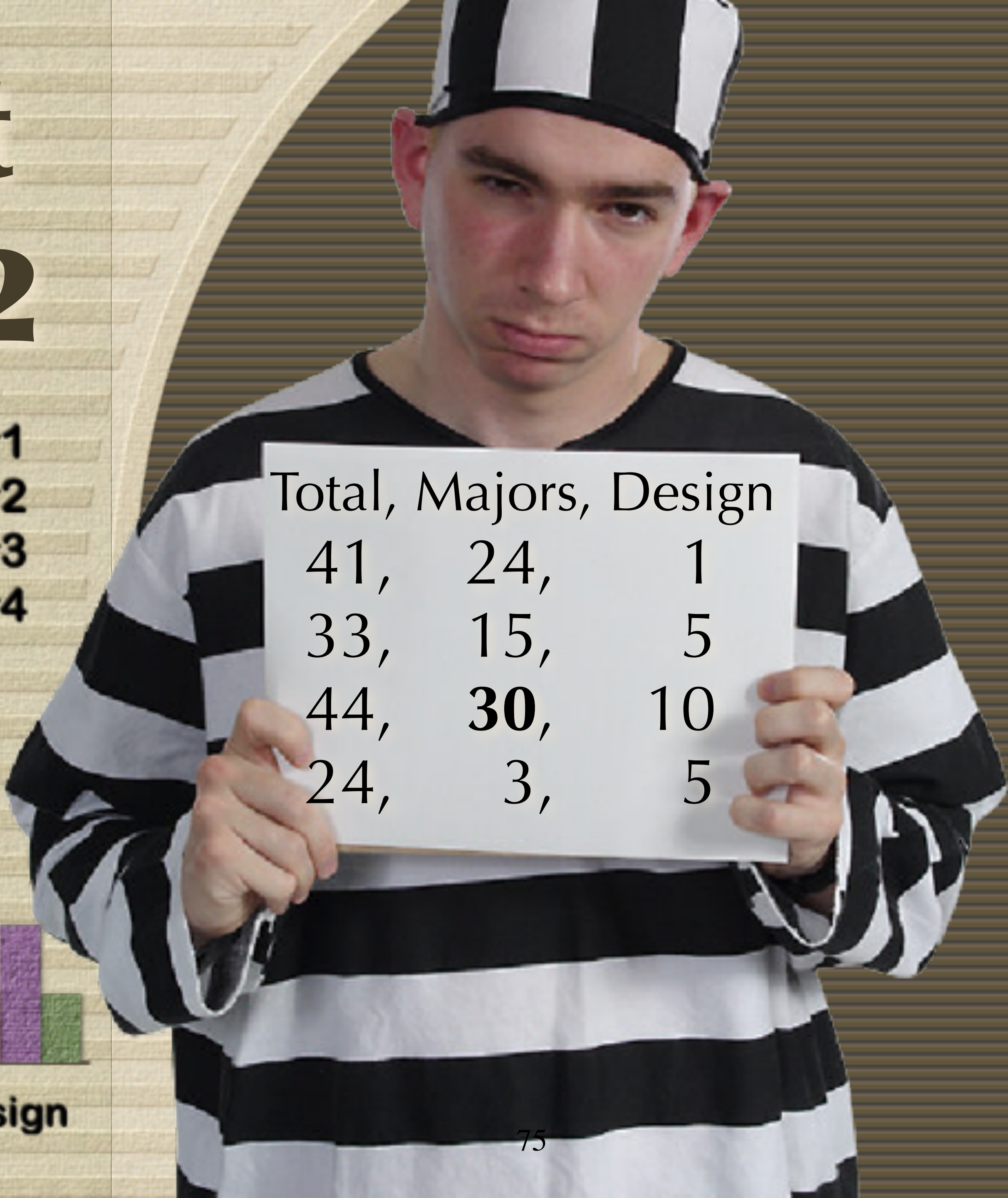


Defect Density Estimation

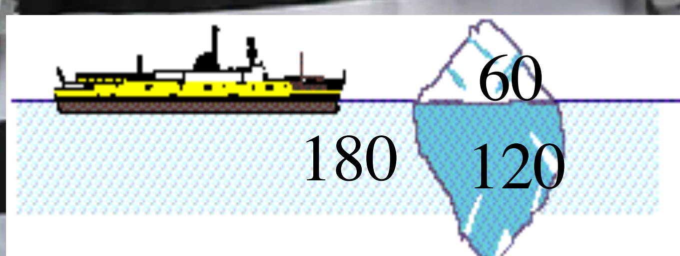
- Total for group (page 81)
  - 20 x 2 = 40 Majors
  - assume 40 are unique
- If 33.333% effective,
  - total in page = 3x 40= **120**
- Of which 2/3 or 80 were *not yet found.* .
- If we fix all we found (40),
  - then the estimated remainder of Majors would be 80 (not found)
  - +8 "not fixed for correctly"
  - = **88** Majors remaining.

Report Page 82





Defect Density Estimation



- Total for group (page 82)
  - 30 x 2 = 60 Majors
  - assume are unique.
- If 33.333% effective,
  - total in page = 3x 60 = **180**
- Of which 2/3 or 120 were not yet found.
- If we fix all we found (60),
  - then the estimated remainder of Majors would be 120 (not found)
  - +10 “not fixed correctly”
  - = **130** Majors remaining.

Conclusions

- Human *defect removal* by Inspections/reviews/SQC is
  - a hopeless cause: not worth it.
- Spec QC can be used, in spite of imperfect effectiveness,
  - to *accurately estimate* major defect level density.
- This measurement can be used to motivate engineers to
  - dramatically (100x! Over about 7 learning cycles)
  - reduce their defect insertion (rule violation)
- to a *practical* exit level
  - (like *less than* 1.0 Majors/page)

Extrapolation to Whole Document

- **Average: 150 Majors/page**
  - Page 81: 120 majors/page
  - Page 82: 180 Majors/page

Chapter 1

Hal Spacejock was sitting at the Black Gull's flight console, his attention riveted to a small chessboard balanced amongst the toggle switches, flashing lights and status displays. Recently he'd skimmed an article extolling the benefits of the ancient game: how playing it would sharpen his mind, improve his memory and increase his attraction to the opposite sex. Chess had been an important part of his daily routine ever since, but after two hundred and seventy-six losses in a row Hal was beginning to doubt the article's claims. He didn't feel any smarter and he couldn't remember the last time he'd spoken to a member of the opposite sex, let alone attracted one. Briefly, he wondered whether it was such a clever idea to play against the Navcom, the Black Gull's onboard computer. Underpowered and outdated, it was still more than capable of running the ship's accounts, navigation and life support systems while beating humans at simple board games. However, since Hal was the only human aboard the Black Gull, his choice of opponents was limited.

'Your turn,' said the Navcom, in a neutral female voice.

'I'm thinking.'

'While you're planning your opening move, can I tell you about a special offer?'

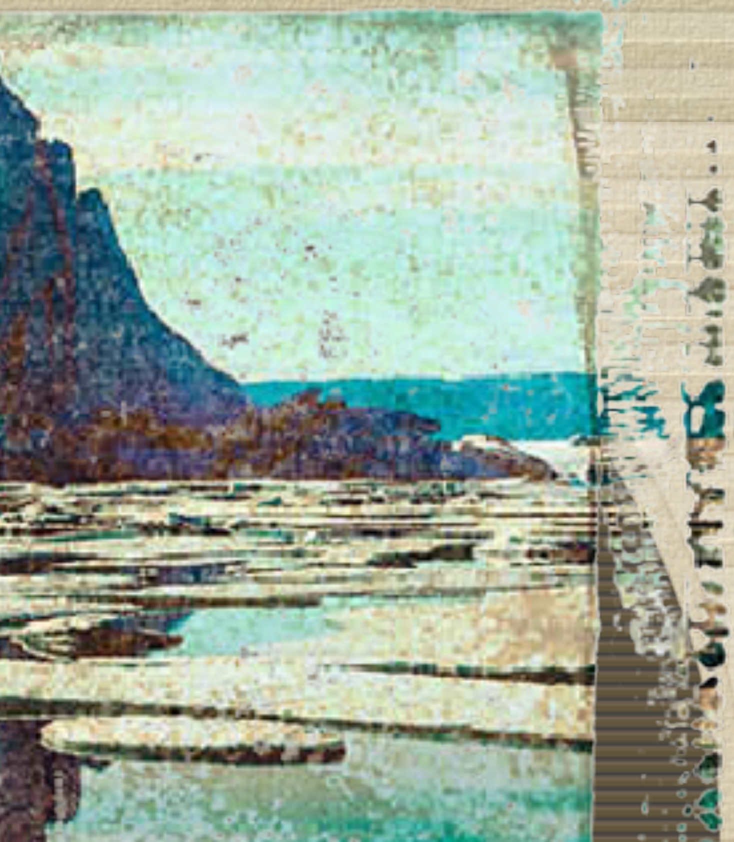
'What kind of offer?' asked Hal suspiciously.

'Planet Books have a chess title on sale.'

'Really? Put it on main.'

2

- **Total in whole document:**
  - **12,300 Majors**
  - 150 Majors/page x 82 pages.



Estimated Project Loss

- If a Major has
  - 1/3 chance of causing loss
- And each loss caused by a Major is
  - avg. 10 hours
- then total project Rework cost is
- about ***41,000 hours loss.***
- (This project <u>was</u> over a year late)
- 1 year = 2,000 hours x 10 people

Feedback on this “simple “formula

Tom Since returning from the QAI Conference in Orlando, I've been attempting to lay the foundation for our product team to develop clear requirements and implement productive inspections as opposed to just going through empty motions. It's definitely been an uphill effort.

One bright moment was my use of the formula that you provided me to

estimate the # of high-severity bugs still in a software product.

I applied it to our product's Test Pass 1 and then forwarded the estimated

number of remaining bugs after Test Pass 1 to the count estimated to

still be in the product when we began Test Pass 2.

This provided me with

a prediction of the number of high-severity bugs that would be found which was within 5% of the number actually found during Test Pass 2. :-)

I can't tell you how much that relatively simple activity buoyed my spirits. Thank you for the time you spent with me in Orlando.

Thanks, Jeff Finn, CSTE, CQA, Microsoft SharePoint Portal Server, 425-703-4213

jfinn@exchange.microsoft.com, May 22 2001

Details of a Real Process Definition for Agile Inspection

- We do not expect to lecture with these slides. They are background information.

Extreme Inspection.
Version:January 12, Originated 2003

Authors: Tom Gilb Tom@Gilb.com & Kai Gilb Kai@Gilb.com

Intended Purpose:

Extreme Inspection <client> Variation:

a simple but powerful version of inspection (Specification Quality Control – SQC) that <CLIENT> can install immediately at low cost.

Rules

- ***The primary Rules we check against are the same Rules that writers will use when writing specifications.***
- ***Initially they will be Clarity, Unambiguousness, Consistency, Traceability, separation of requirements and solutions, and separation of Performance, Functions and Designs.***
- ***See separate document: "Rules for Specification Writers."***

Extreme Inspection Outcome

- **The outcome of this type of inspection is to give a fair measure of Major defect density.**

Intent of Outcome

- **The intents of the Major defect density measure are:**
- **Clean: to make sure that polluted specifications do not enter the next working processes.**
- **Learn: to motivate specification writers to learn and follow <CLIENT> best practice specification rules.**

Internal Extreme Inspection Goals

- *"The expected effects of rigorously carrying out this process are:"*
- **Density:**

Scale: Estimated remaining Major defect density per logical page (300 Non Commentary words)

Past [December 2002] 50-100 Majors/Page <- Multiple sample inspections

Goal [Jan 2003] less than 10 Majors/Page

Goal [Jan 2004 or sooner if feasible!] less than 1 Major/ Page

External Extreme Inspection Goals

Project Efficiency

Scale: Total project time to successfully complete a project

Past [Dec 2002] ???

Goal [Dec 2003] = 70% of Past [Dec 2002]

Goal [Dec 2004] = 50% of Past [Dec 2002]

Comment:

This will be accomplished by

less back and forth,

and reviewing of requirement documents,

and by shorted coding and test times,

and by less effort when work is contracted out of country or to sub-suppliers.

More time at the requirement stage is expected.

Process Management of Extreme Inspection: 1

– 1. *Inspection Outcome Justification*

- The outcome of this variation on conventional Inspection processes is **to determine 'specification exit'** by measuring and estimating Major defect density. The outcome is NOT (as with conventional inspection) to 'clean up' bad work.
- The result of this outcome limitation is that many of the time honored conventions of Inspections (as in Gilb & Graham: Software Inspection) are NOT necessary or desirable. We only need to do whatever gives a **reasonable measure of defect density**. We only need to focus on determining that the specification is exit-able or NOT.
  - So we do not need to get maximum effectiveness by having a large team or by using one hour per page or by looking at all pages (we can sample in 10-40 minutes and use one or 2 people).
- In simple terms **if we find (checker detects) one or more Majors** in a page, it is NOT exit-able, because the real estimated quantity of majors actually there, exceeds the Exit limit of 'one per page'. If we find less than one major defect on 4 pages, it probably is *economic* to exit the spec.
- ***Economic*** is the key word. We are trying to determine **if it pays off** to exit now, or to rewrite the spec to a cleaner level now.

2. Inspection Cost Charging.

- All costs for the writer, the checker and a possible process guide, will be
  - charged to the project the writer is working on,
  - and to the QC process costs specifically.
  - *Rationale: so we can track the true costs of doing this and the degree to which it is done*.

3. Auditing this process:

- The Inspection (Spec QC) process must be regularly (monthly) audited
  - to make sure it is really conducted according to intent
  - and is not corrupted or misunderstood.
- This includes double checks on audits
  - to see if the conclusions of the check and the audit are reasonably consistent.
- Frequent audits are necessary in the beginning and with newcomers.
- Auditing will be done by the process owners.

Process Management of Extreme Inspection

- ***4. Process Improvement***
  - –The process needs to be continuously updated
    - •mainly in the tools kit which defines and supports the inspection process:
    - •the checklists,
    - •the process definitions,
    - •the computer data collection support
    - •by the official process owner.

5. Process Ownership

- There must be an official process owner to champion (and to manage ‘local’ champions),
  - spread,
  - audit,
  - and improve the process,
  - as experience and insight dictates.
- This can be a group.

6. Process Sponsorship

- The executive sponsor of this process should be official and visible

7. Confidentiality

- The checker shall *never* reveal the numeric result of an Inspection to anyone else except the writer.
  - The writer may reveal the results if they want to, but they are not obliged to do so even to their direct manager (who should not even ask!).
  - The results of an inspection, as recorded in the Specification Quality Control Database, are never to be released, revealed or reported with the name of the writer or information (such as document ID) that can lead to their identification.
- *Rationale:*
  - *to prevent fear of defamation leading to false reporting of results.*
  - *To emphasize that the process is there to help the writer reach the corporate quality level required.*
  - *It is not in any way of time to be used for personal job performance evaluation.*
  - *Evaluation should be based on EXITED specifications, and their timeliness only.*
  - *Managers need to be informed and reminded of this cultural paradigm by the process owners.*

Process Management of Extreme Inspection: 3

- ***8. Expected Effectiveness***
- We expect that the Major defect finding effectiveness of the checking process will be in the range of 10% to 35% of the actual real Majors present in a specification.
- This is quite sufficient to *estimate* the actual total number of majors actually present.
- We can then estimate with *sufficient accuracy* (say ±20%) determine levels of Majors in entire spec and in spec after correction of listed (by checkers) defects.

Defect Rates (repeat of earlier slide intentional)
Here is what really happened afterwards
in 2003 Pilot Financial Shop, London, Gilb Client
Spec QC/Extreme Inspection + Planguage Requirements

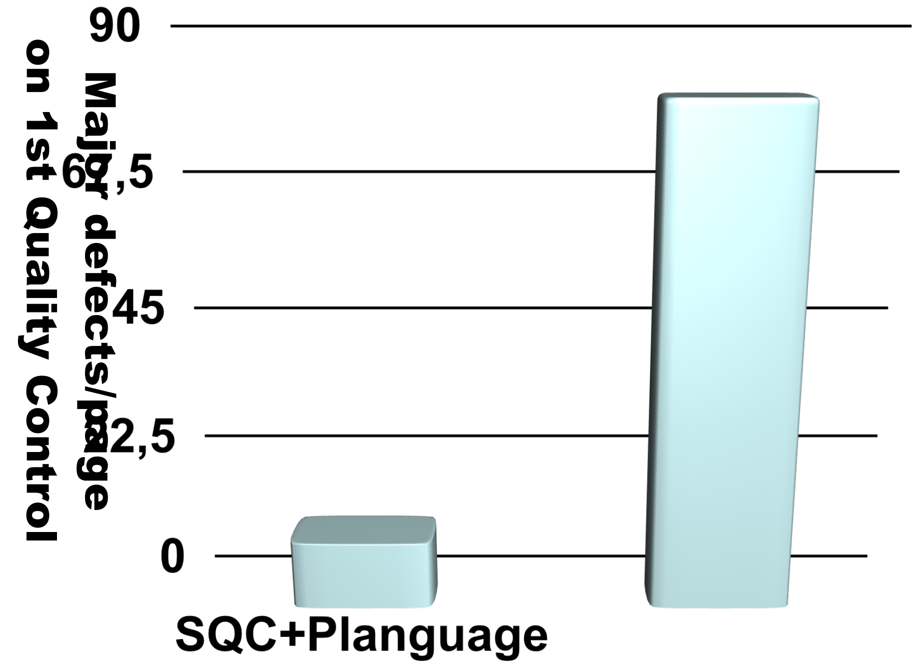
Across 18 DV (DeVelopment) Projects using the new requirements method, the average major defect rate on first inspection is 11.2.

4 of the 18 DV projects were re-inspected after failing to meet the Exit Criteria of 10 major defects per page.

A sample of 6 DV projects with requirements in the ‘old’ format were tested against the rules set of:

> The requirement is uniquely identifiable
> All stakeholders are identified.
> The content of the requirement is ‘clear and unambiguous’
> A practical test can be applied to validate it’s delivery.

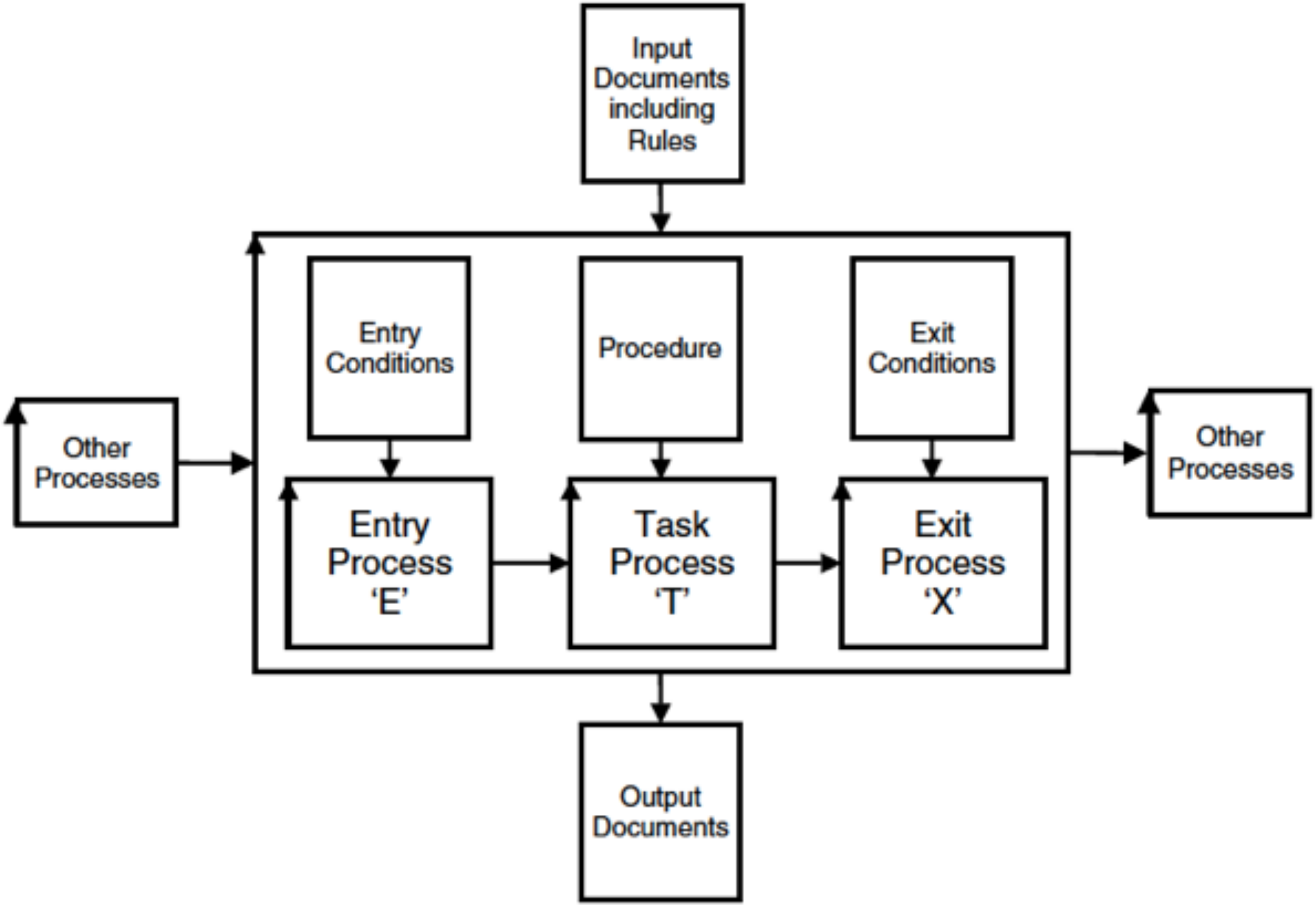
The average major defect rate in this sample was 80.4.



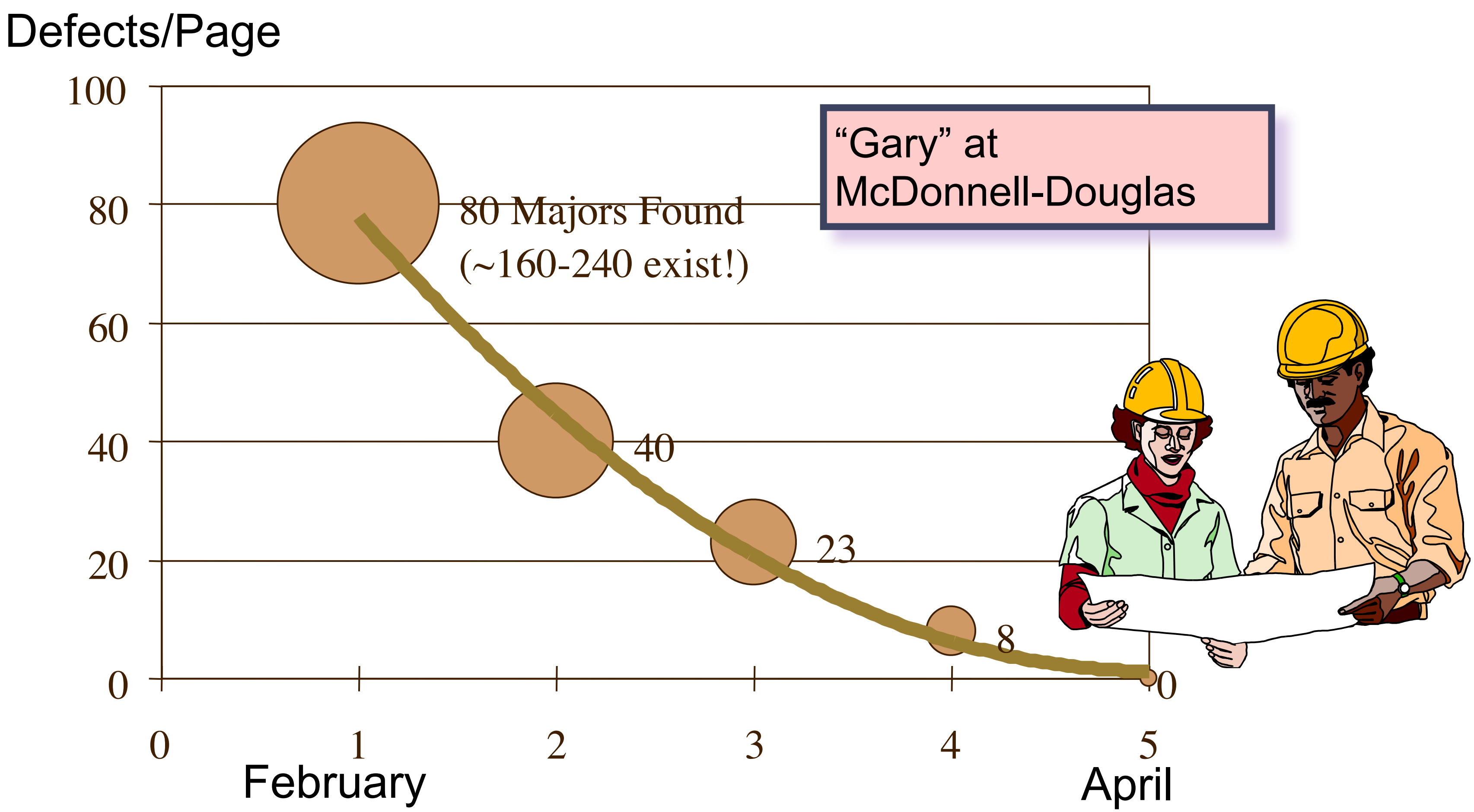
9. True Measure of Inspection Progress.

- The correct and relevant measure of how effective the Inspection process is working, is NOT as many would assume the quantity of Major defects found and fixed by an Inspection.
  - In fact we strongly recommend that this measure is well hidden from public view! (It has its uses!).
- The true measure is the average level of Major defects/Page which we can consistently release.
  - We need to move from about 100 Majors/Page down towards about less than one per page.
  - This cannot be achieved by finding and fixing defects (because we cannot find a large percentage at all)!
  - It can only be achieved in practice by *motivating* writers to reduce defects actually injected in their work, from 100, and move them down towards one maximum injected/page.
  - This is the ‘individual defect injection learning rate’.
  - Individuals seem capable of reducing their own defect injection by about half (50% fewer for each cycle of learning (write, inspect and rewrite with 50% less cycle).
- The measure of real progress is the released defect density, and it is this measure which will most closely correlate with later statistics on quality and productivity of projects.

5a. Numeric Quality Gateways



5a. Numeric Quality Gateways Improve Quality of work



Inspections of Gary's Designs

An Advanced Example

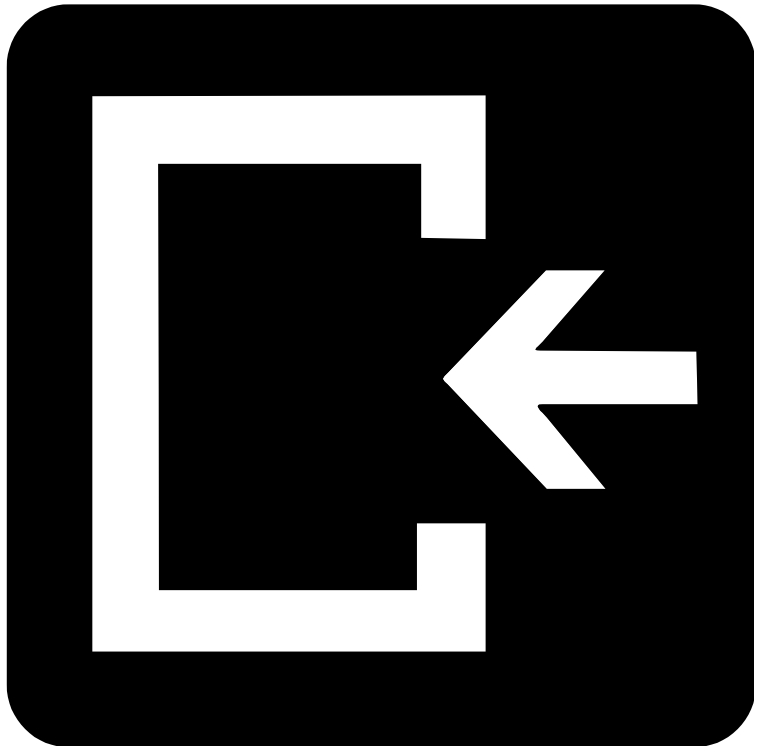
Source Erik Simmons, erik.simmons@intel.com 25 Oct 2011
Personal Public Communication

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

EI Entry Conditions



EI.E1:

- At least one of the participants
  - has done a well conducted successful inspection once before,
  - or been briefed by a competent practitioner,
  - or will be guided through the process by a competent guide (ideally an expert in this process).
- *Rationale: people need to have some reasonable sense of how to do this process, otherwise it can become corrupted. We believe we can avoid formal training in the method, but we need some knowledge and experience of it in place.*

EI.E2:

- The specification writer sincerely believes that
  - the defect level is low enough to exit.
  - They have done personal checking against the rules themselves and find no defects.
- *Rationale: the writer should*
  - *take the trouble to make sure the spec is as clean as possible before inspections.*
  - *They should not misuse people and time to compensate for sloppy work.*

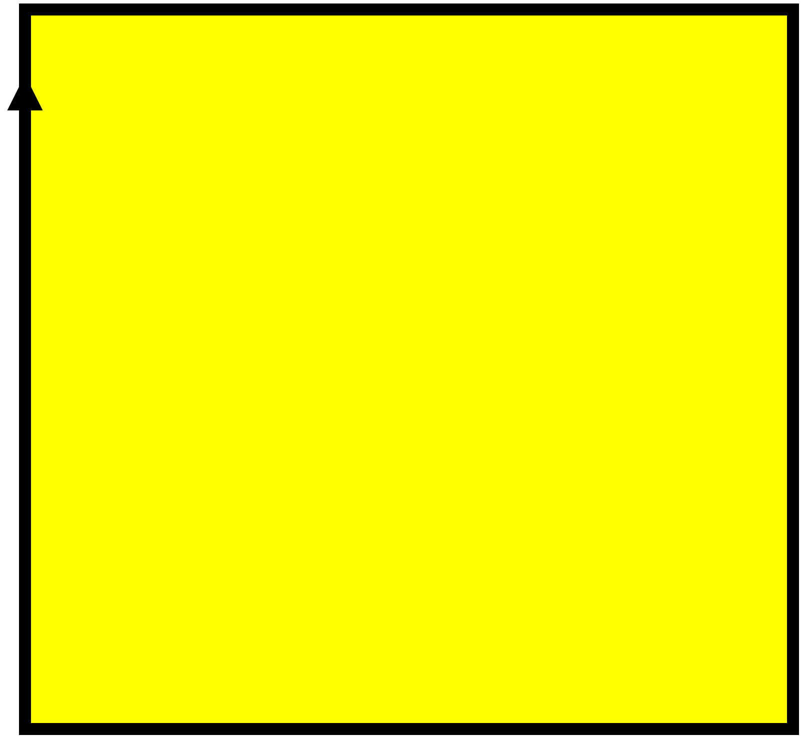
EI.E3:

- Exited copies of all source specifications are available.
  - *Rationale: there is little point in checking consistency against highly polluted source specifications.*
  - *(example by using bad Business Requirements to check new System Requirements).*

EI.E4:

- An updated 'Inspection Toolkit' (with specification Rules, Checklists (for learning to apply the rules in practice), Process descriptions, forms, electronic support, intended readership role information) is available and is understood by the participants.
  - *Rationale: This tool kit is the real definition of the Inspection process. This really determines correct use of the method.*

Ex In Procedure



EI.P1:

- The specification writer (‘writer’)
  - finds one other person (called a Checker)
  - to (help) carry out the QC (Quality Control) of their specification.

EI.P2:

- **a meeting time, with maximum duration 1.0 hour is agreed.**
- **(if the Checker is experienced, they can in fact do their checking at any time, alone, and report their results to the writer.)**

EI.P3:

- The writer makes sure the checker is knowledgeable about the following:
- the spec's intended readership and their uses of the spec.
- the specification Rules that apply (and their practical interpretation)
- The definition of Major defect, and how to spot them
- the purpose of the Spec QC process (to help the writer get to real exit-able level of defect density).

EI.P4:

- **The writer and the checker will each select the same one logical page 'at random' (300 Non-commentary words) sample to check.**
- **The writer is now performing the role of a 'checker' on their own work.**
- **They should agree that the page selected is representative of the quality of the rest of the document.**

EI.P5:

- **checking will be done individually**
  - **(but maybe in same room)**

EI.P6:

- the initial checking time will be 10 minutes.
- If NO Major defects are found by either checker.
- The checking process will continue for another 30 minutes.
- Even if no further Majors are found.

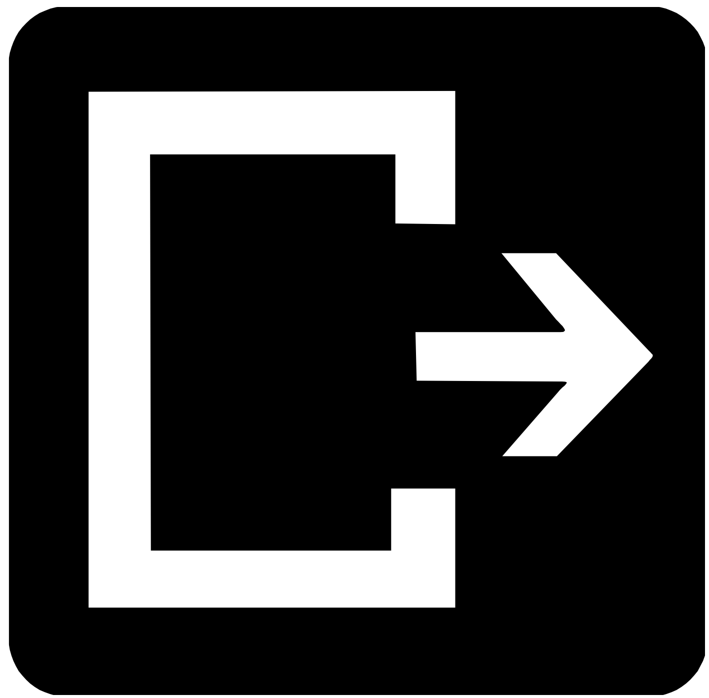
EI.P7:

- **If any Major defect is found**
  - **(and acknowledged by the writer as a real Major defect)**
  - **in the first 10 minutes of checking,**
  - **then this will be considered a sign that the spec contains many more major defects.**
  - **The writer will consider whether they want to stop the QC process and improve the spec,**
    - **or whether they want to continue for another 30 minutes to gather more Major defect cases**
      - **(to better signal what they need to rewrite).**

EI.P8:

- **At the end of the checking time,**
  - **the writer**
    - **(or the checker if they decide to take reporting responsibility)**
    - **will calculate the estimated Majors/Page in the current document**
    - **(using formulas or tools supplied)**
    - **and will report (on a form or to a database)**
      - **all time used and results**
      - **(Majors found,**
      - **Majors/page estimated,**
      - **decision to Exit or not, etc.)**

EI Exit Conditions



EI.X1: Defect Density Condition:

- Estimated Major Defects remaining per page is less than 1 per 300 Non commentary words (initially until end 2003 10 Majors, to get a lenient start).
- FORMULA FOR ESTIMATION:
- Assume 33% effectiveness of the 2-checker checking-process.
- Total Unique Majors acknowledged by writer, found in the sample logical page, times 3, gives a reasonable estimate of Majors/Page. This is before writer correction of known Majors.
- Note: the effectiveness for a 3 checker group is slightly higher say about 40%. This figure needs to be determined by your own measurement.

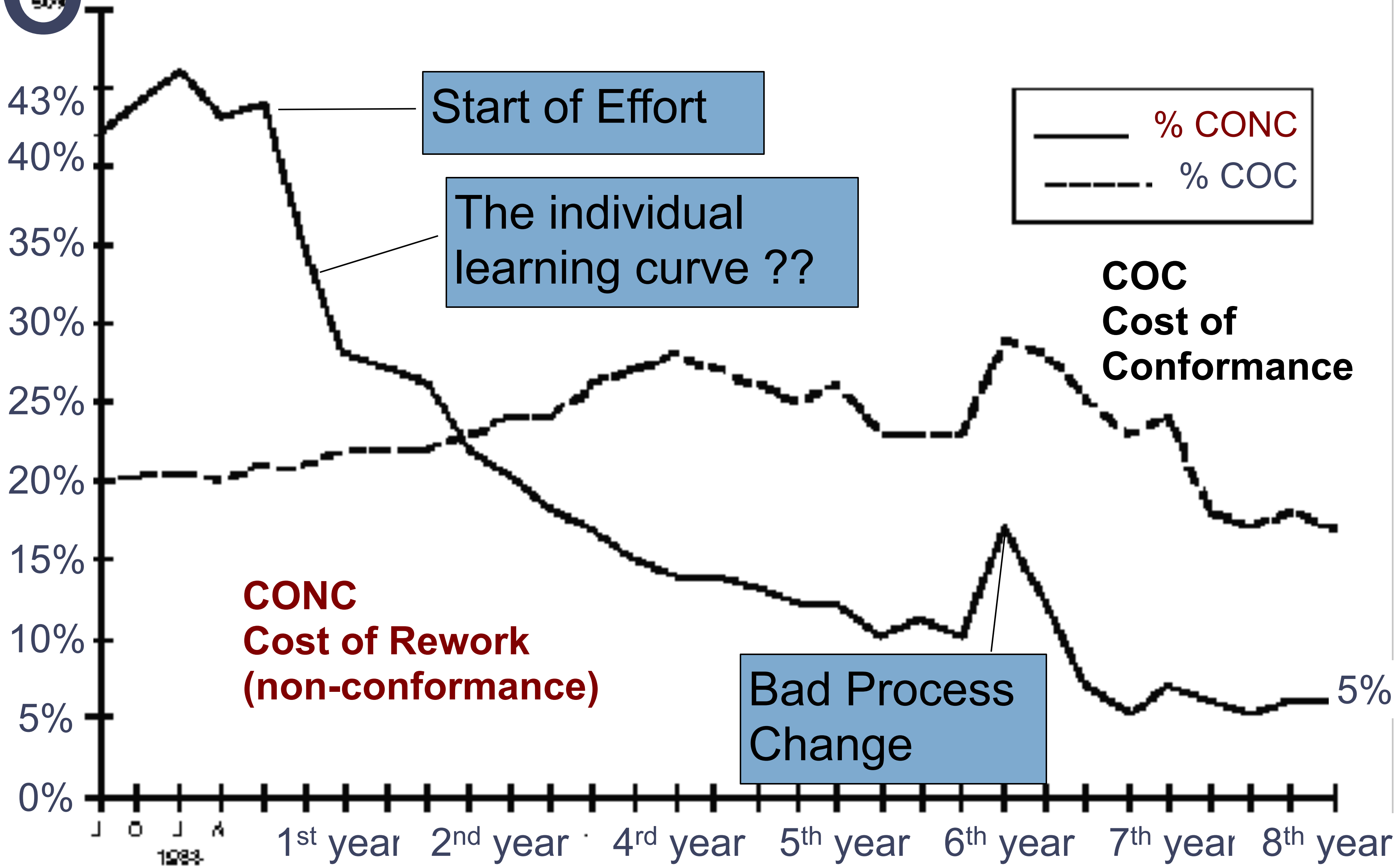
- OPTION: we might manage the exit level at an individual writer level to gradually motivate them to improve by about 50% (defect injection) less per iteration of the write and check cycle. <- KM idea – TG likes it!

- NOTE: THE 33% effectiveness is based on experience, but it could vary, for example depending on the rate of checking used. The rate is controlled here because the time and the volume (a logical page) are controlled in the process.

EI.X2:

- **Writer Veto**
- **The specification cannot exit if the spec writer wants more time to improve it.**

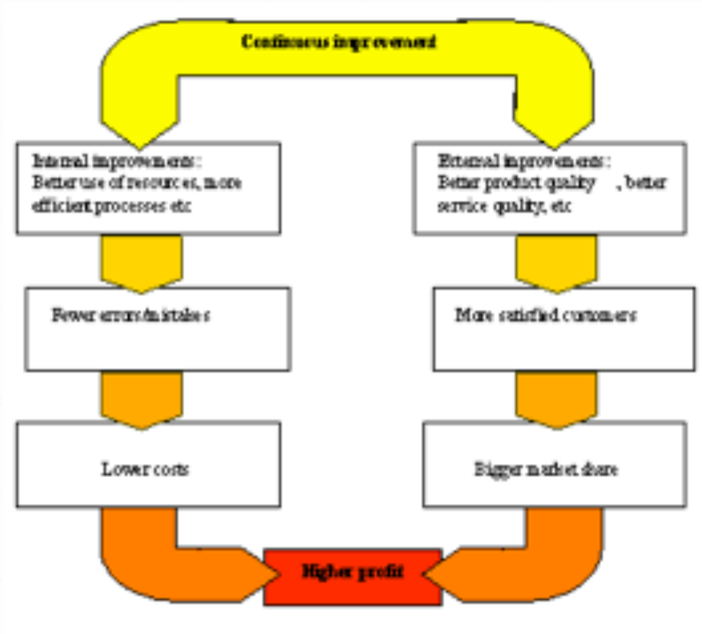
6 DPP Improves Quality by 10x: Raytheon



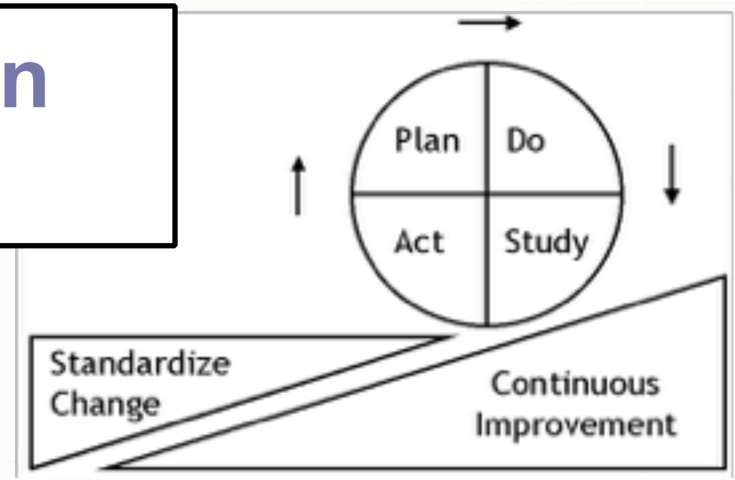
www.sei.cmu.edu/publications/documents/95.reports/95.tr.017.html

Defect Detection strategies versus Defect Prevention strategies

- Defect detection
  - (inspection, test, customer reports)
  - Is *ineffective* for getting high bug-freeness into systems
  - It is better than nothing
  - Inspection is cheaper than test-and-debug
- Defect Prevention - is at 2 levels
  - process improvement
    - (CMMI Level 5)
  - individual capability improvement
    - (50% per motivated cycle)
- Defect prevention is BY FAR the smartest one



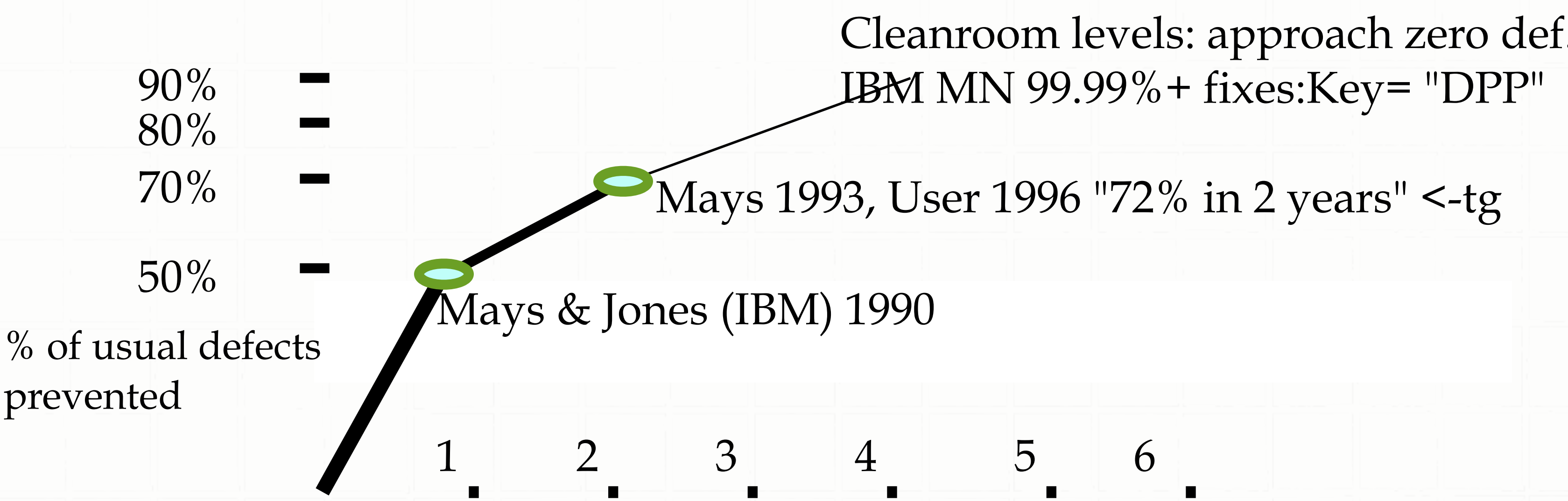
Prevention Costs



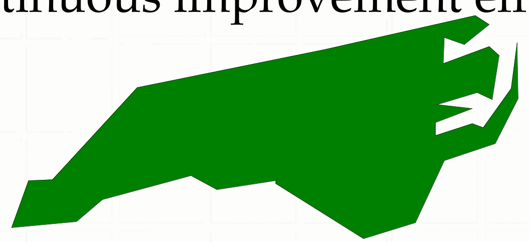
Deming Cycle

- 5%, stable at 5%
  - of development costs
  - (Raytheon 1993)
- 0.5 % of development costs
  - (Mays 1995)

Defect Prevention Experiences: Most defects can be prevented from getting in there *at all*



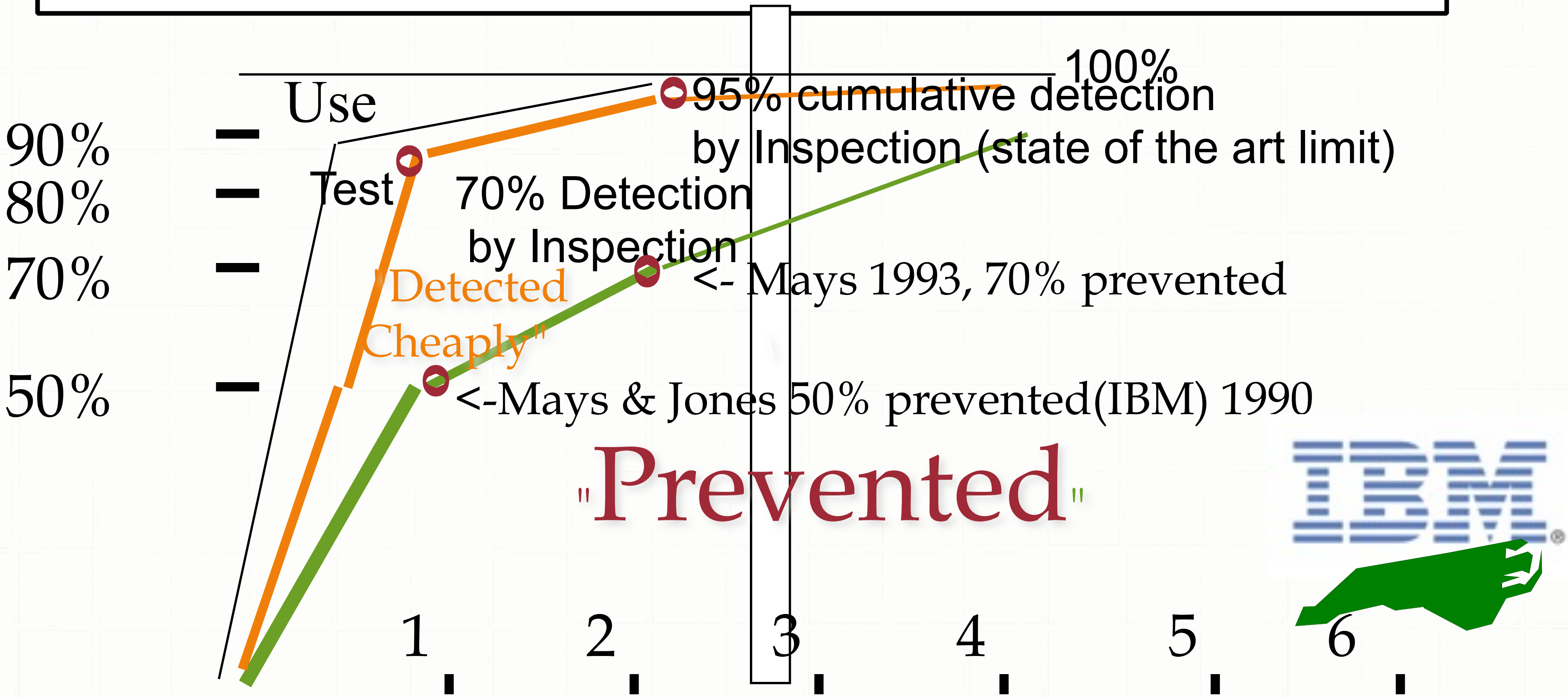
- Years of continuous improvement effort



North Carolina

IBM Research Triangle

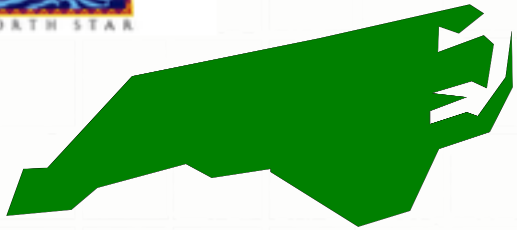
Prevention + Pre-test Detection is the most effective and efficient



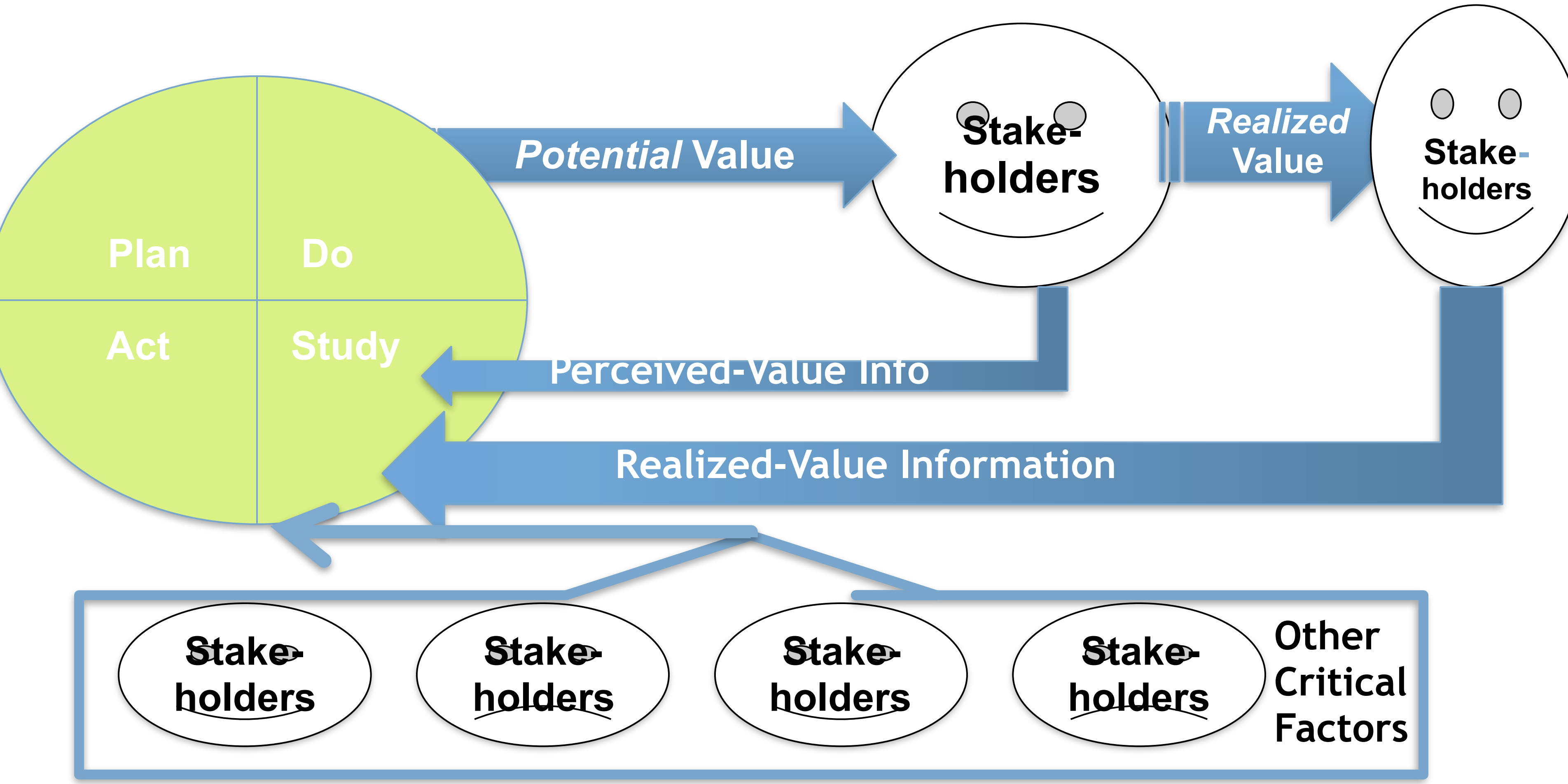
- Prevention data based on state of the art prevention experiences (IBM RTP), Others (Space Shuttle IBM SJ 1-95) 95%+ (99.99% in Fixes)
- Cumulative Inspection detection data based on state of the art Inspection (in an environment where prevention is also being used, IBM MN, Sema UK, IBM UK)

IBM MN & NC DP Experience

- 2162 DPP Actions implemented
  - between Dec. 91 and May 1993 (30 months)<-Kan
- RTP about 182 per year for 200 people.<-Mays 1995
  - 1822 suggested ten years (85-94)
  - 175 test related
- RTP 227 person org<- Mays slides
  - 130 actions (@ 0.5 work-years
  - 34 causal analysis meetings @ 0.2 work-years
  - 19 action team meetings @ 0.1work-years
  - Kickoff meeting @ 0.1 work-years
  - TOTAL costs 1% of org. resources
- ROI DPP 10:1 to 13:1, internal 2:1 to 3:1
- Defect Rates at all stages 50% lower with DPP



7a 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.

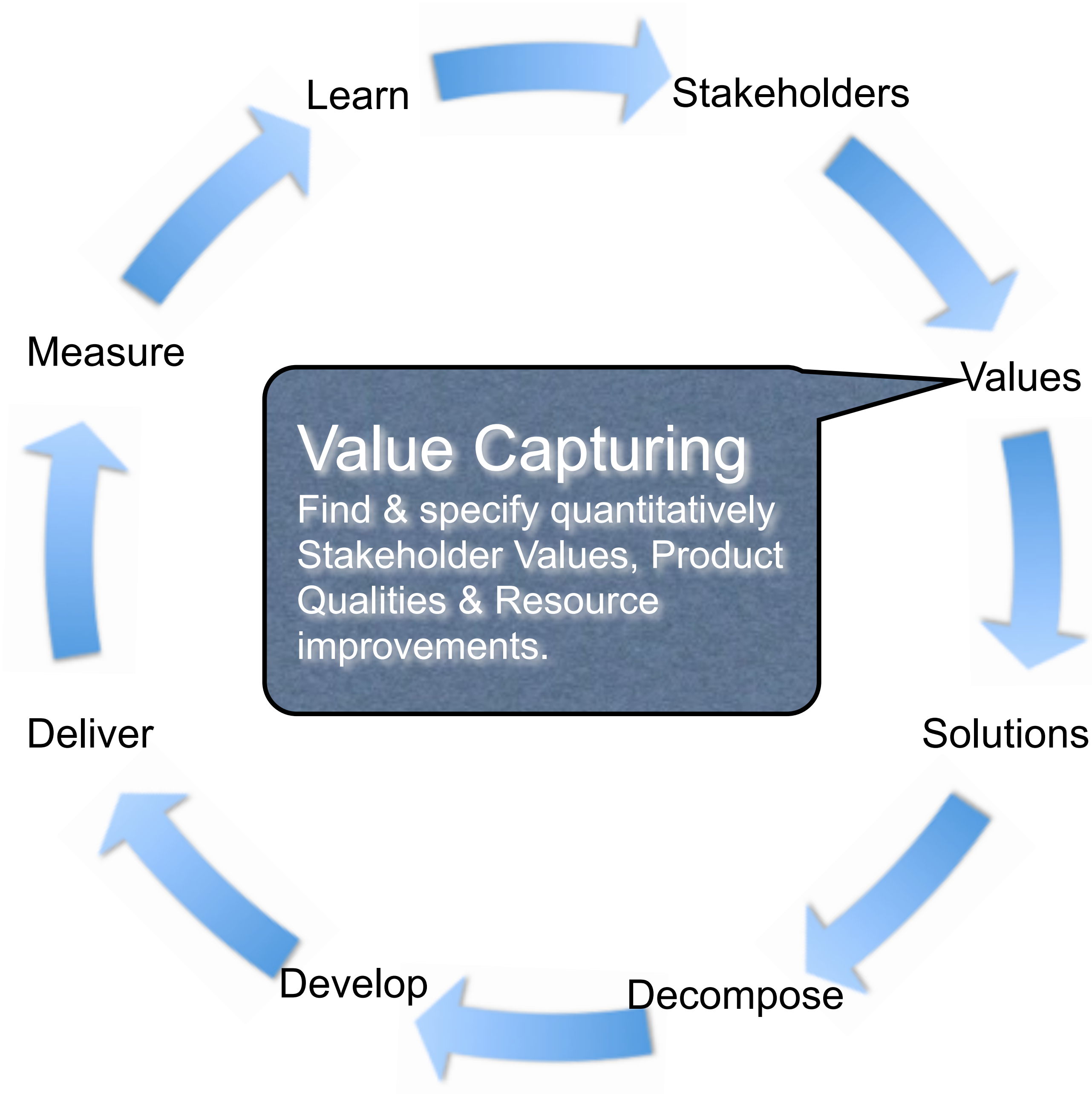
7b

Value Management Process

7b


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

7b



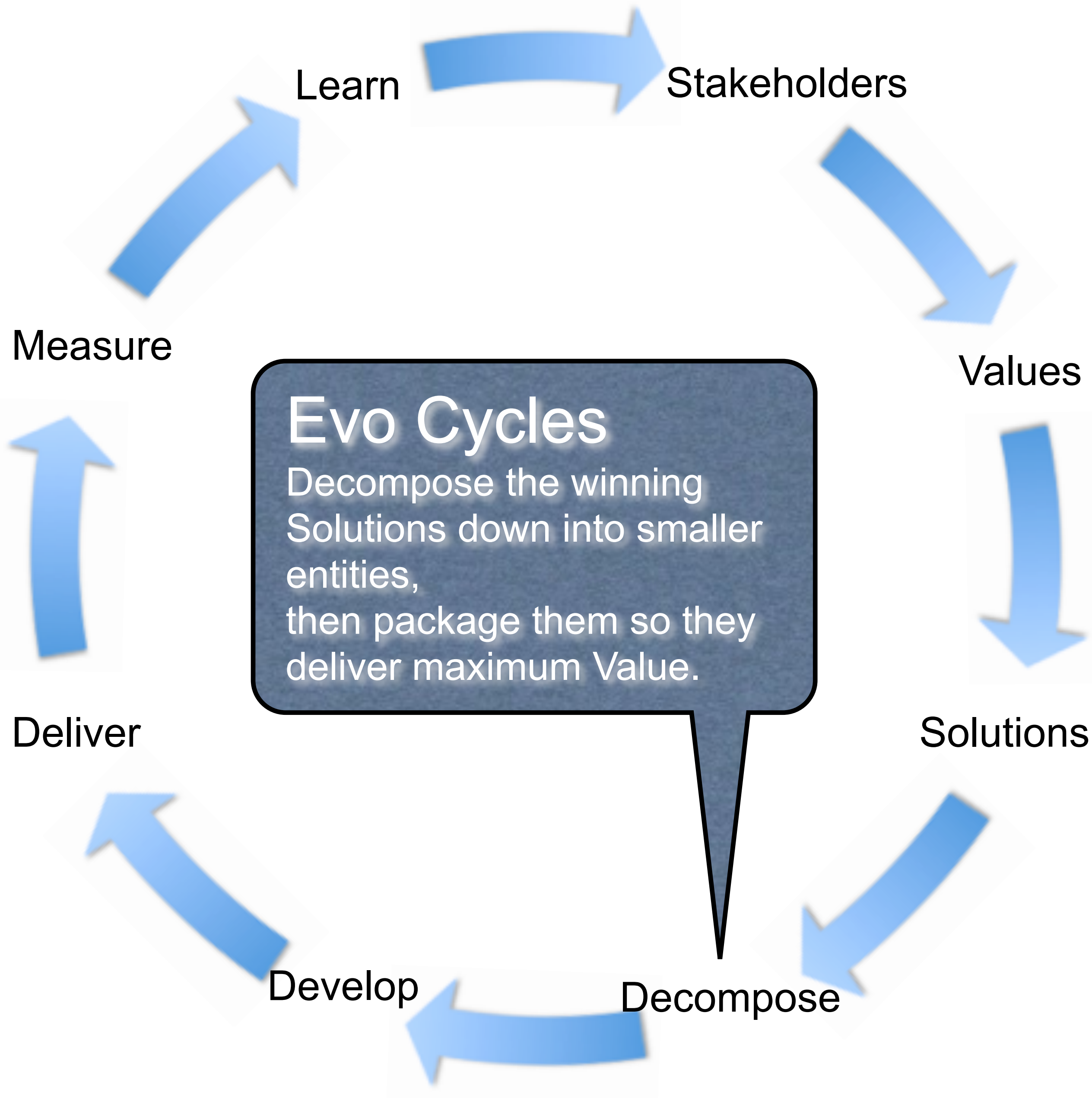
Value Capturing

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

7b


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

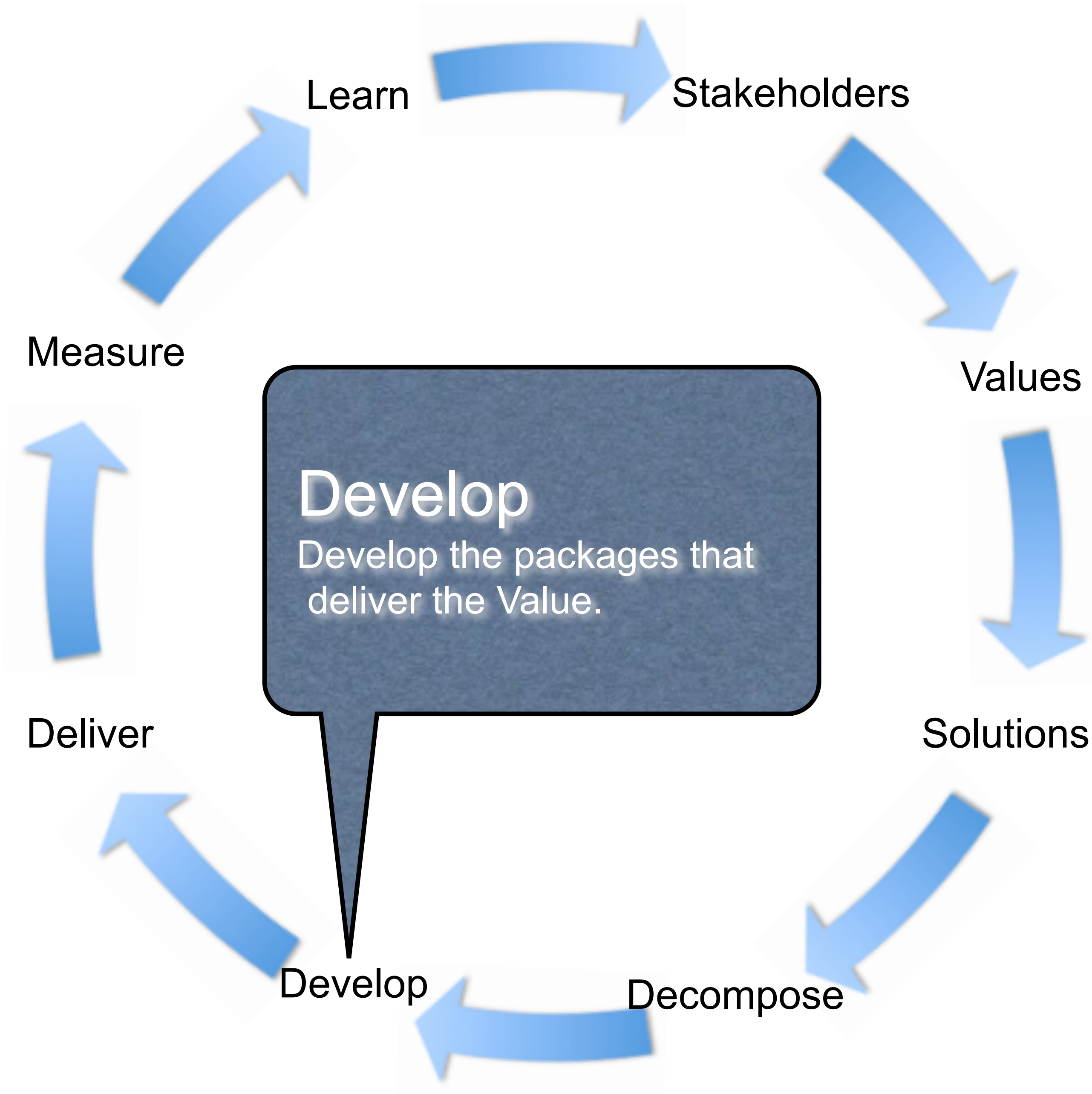
7b



Evo Cycles

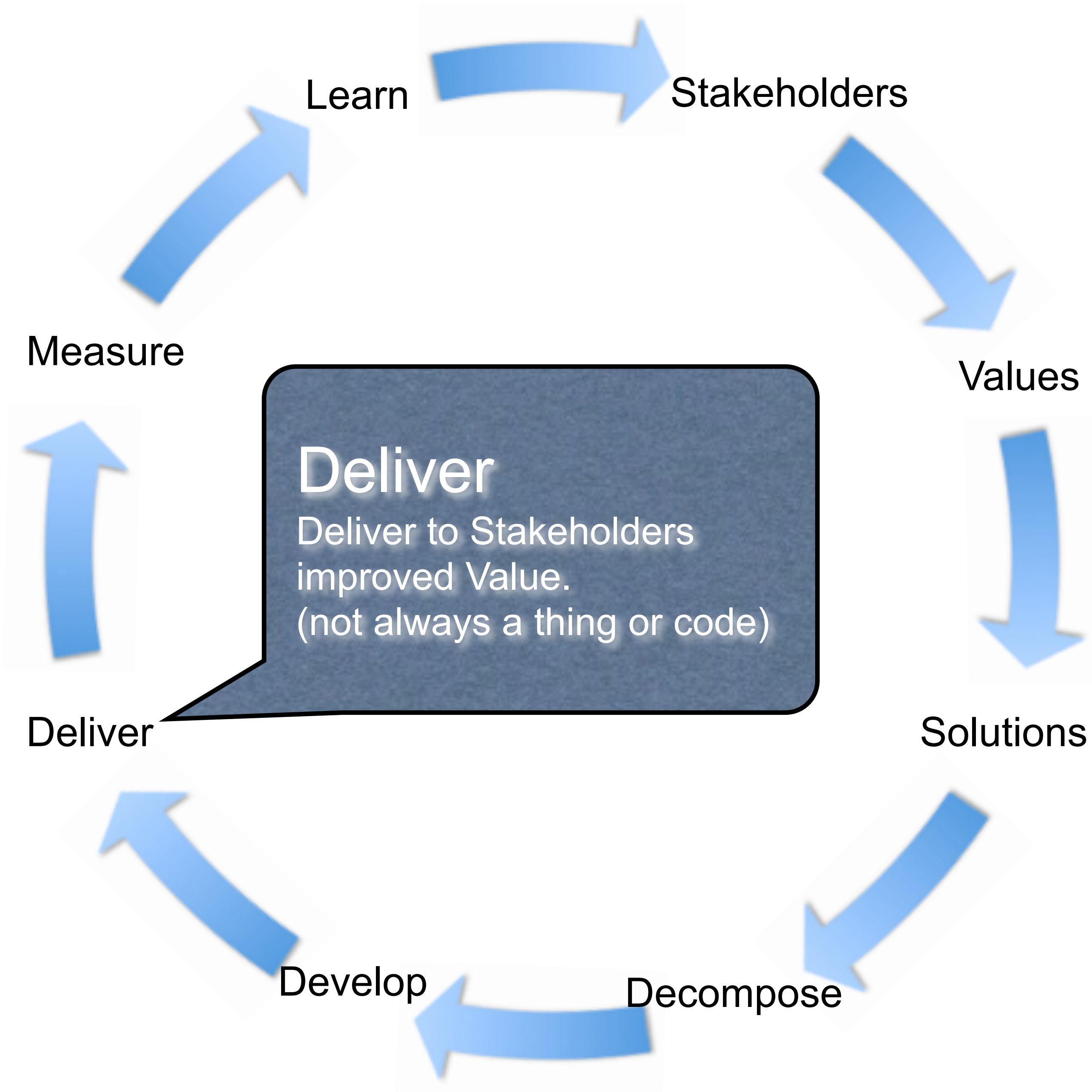
Decompose the winning Solutions down into smaller entities,
then package them so they deliver maximum Value.

7b

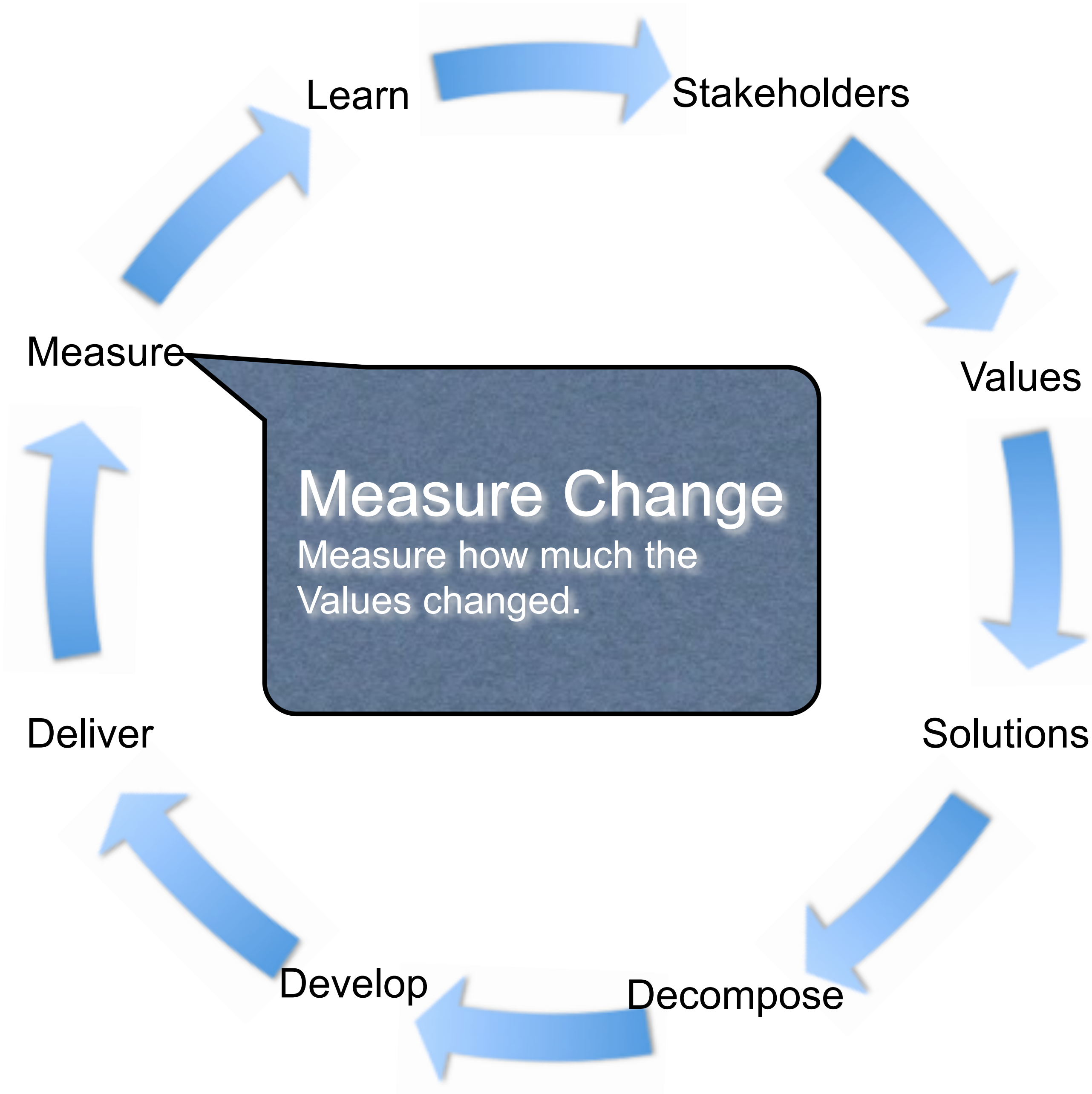


Learn
Stakeholders
Values
Solutions
Decompose
Develop
Deliver
Measure
Develop
Develop the packages that deliver the Value.

7b


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

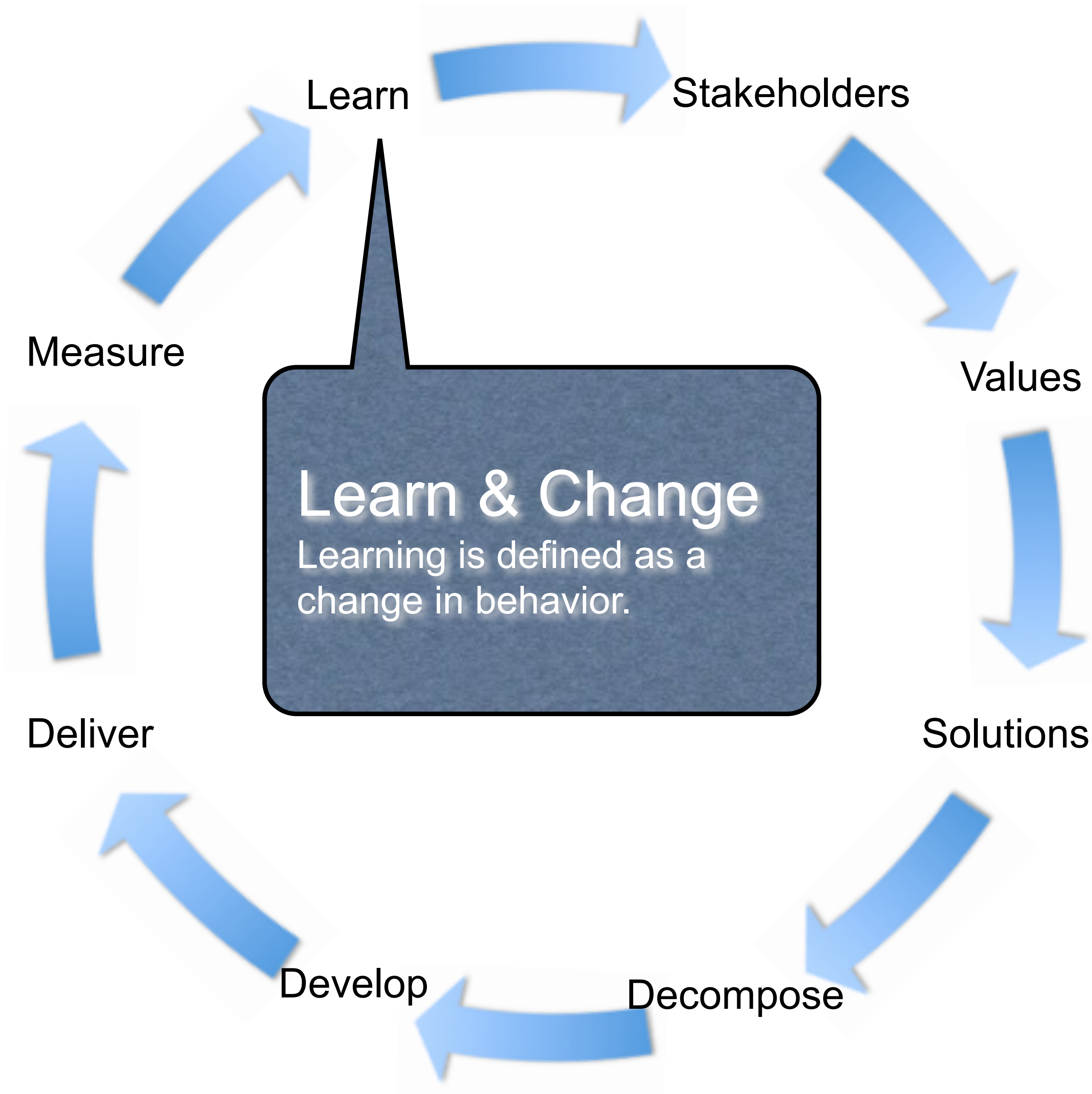
7b



Measure Change

Measure how much the Values changed.

7b


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

7b


Learn
Stakeholders
Values
Solutions
Decompose
Develop
Deliver
Measure
Value
Management
Process

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

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 (Power to the Programmers gilb.com/dl821)
  - 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"

Thanks!

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

Copy of these slides will be in Downloads/Slides:

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

For details on all subjects see my new E Book
leanpub.com/ValuePlanning (frree core, cheap rest)