

October 2010

Why you need to measure up to your stakeholders – always!

Isabel Evans

isabel.evans@bcs.org

Who is Isabel Evans?

- *More than twenty years in IT industry*
- *Author Achieving Software Quality Through Teamwork*
- *Chartered IT Professional and FBCS*
- *Worked in many types of project and business domain*

- *From October 2010, Quality Manager at **Dolphin Computer Access**
– Dolphin's range of computer software delivers independence to
computer users with vision impairments and learning disabilities.*



Dolphin's Vision

“It is our vision to deliver independence to people with vision and print impairments all over the World. With over 160 million visually impaired people and 600 million dyslexic or learning disabled people, that's no small challenge. But we're determined to make a difference and have been working hard on that goal since Dolphin started in 1986.”

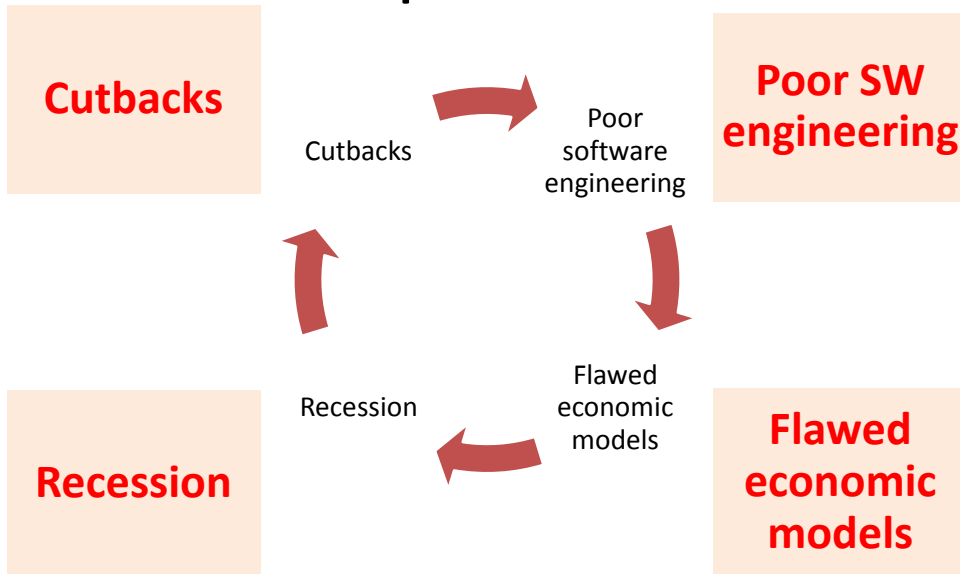


Noel Duffy
Managing Director
Dolphin Computer Access

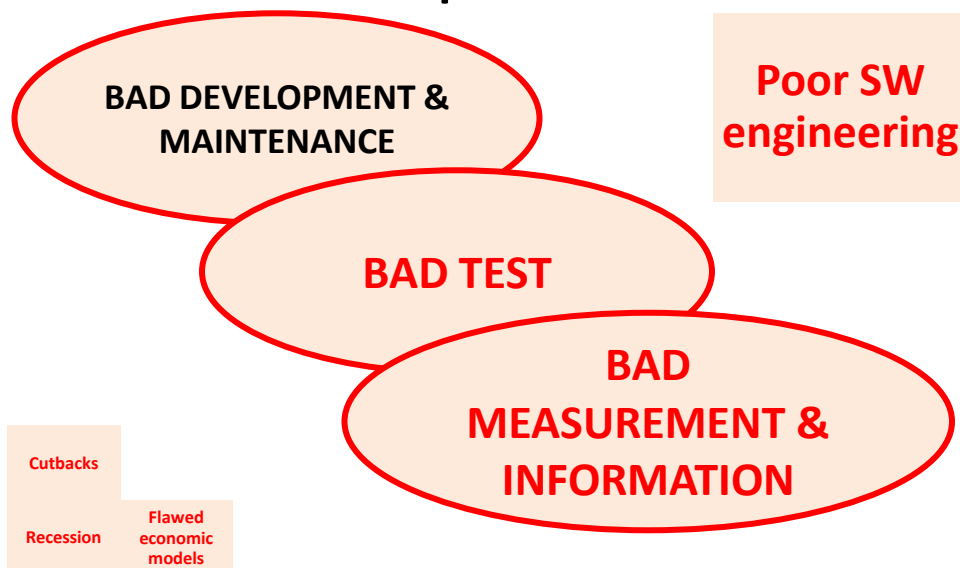
This talk...

- Builds on experiences of the last 30 years
- And on published information from the IT industry
- Looks across the whole IT / test industry
- Discusses the need for all aspects of IT to improve
- Focuses on our stakeholders
 - Whatever the business we are in...
- Is not **entirely** accessible... Because I am still learning...

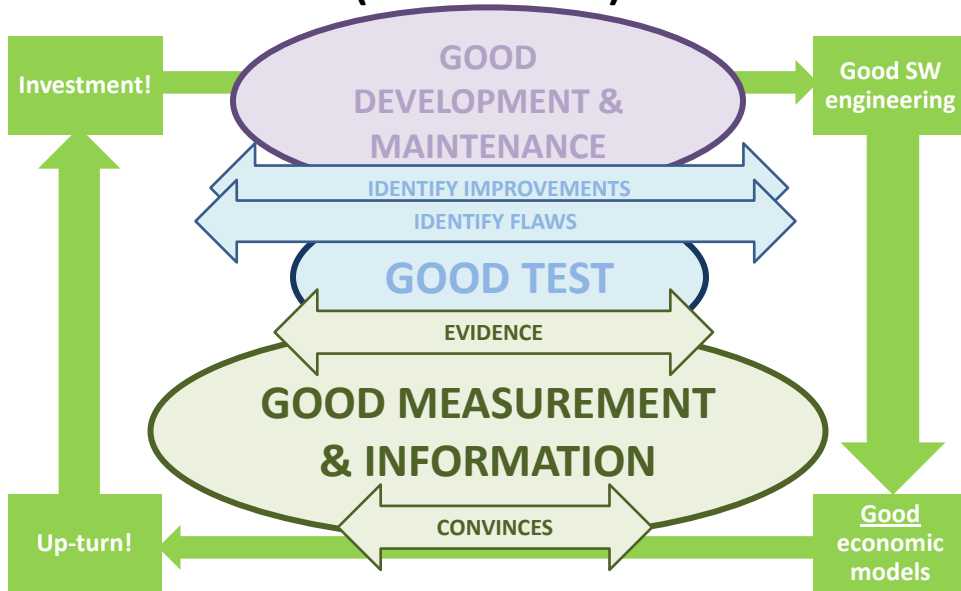
A challenge to our comfort from Capers Jones



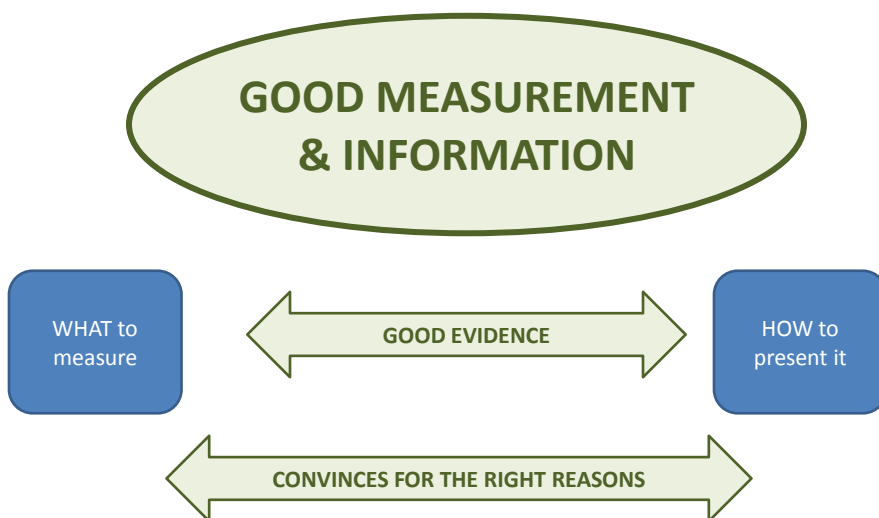
Test community is (partially) culpable...



Let other pens dwell on guilt & misery...
(Jane Austen)



Let's look at what we need to do



(1) Quality

WHAT TO MEASURE –

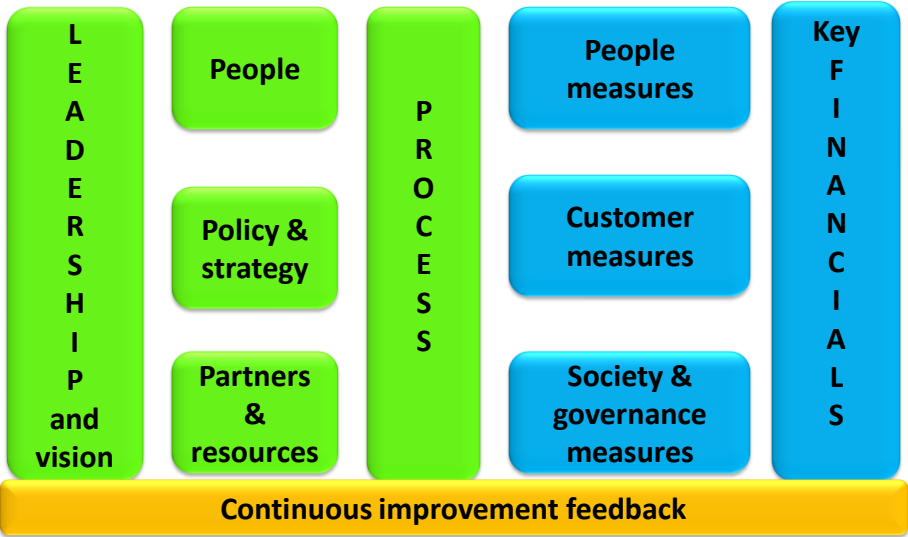
Don't get too fretful!
Don't worry about the jargon

Think of the basics
-Quality
-Organisational frameworks
-Stakeholders
-Costs
-Messages

- IT
 - Manufacturing: farm method, blemishes
 - Product: Size, shape, taste, colour
- Business
 - Value: Shelf-life, price
 - User: right for our recipe, taste
- Transcendent: We always source our tomatoes from...

(2) Organisational excellence models

e.g. EFQM™

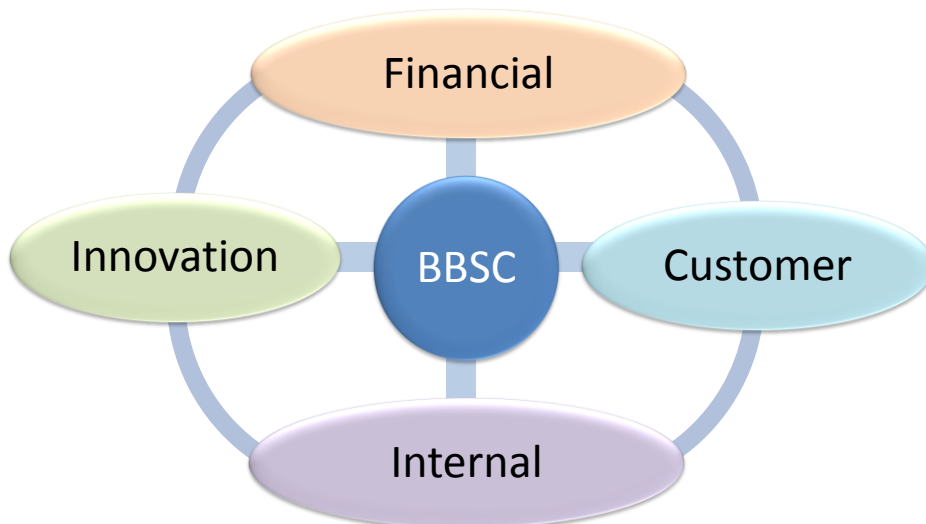


Rowing boats and liners...

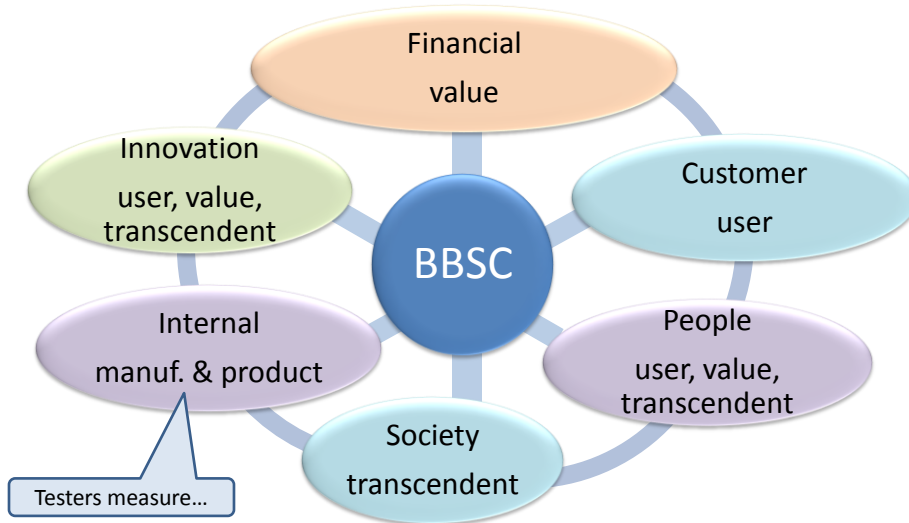
“...a rowboat does not need the same rigor of development as does a cruise ship...”
Capers Jones

“... you can't build a cruise ship by assembling 8000 rowboats...”
Capers Jones

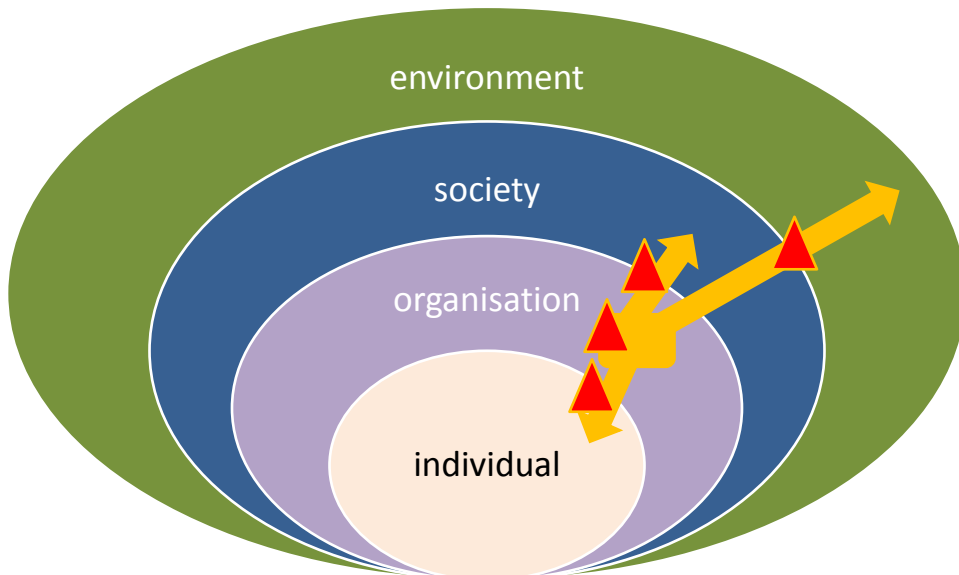
Organizational frameworks e.g. Balanced business scorecard



Combined scorecard – EFQM+Qualities

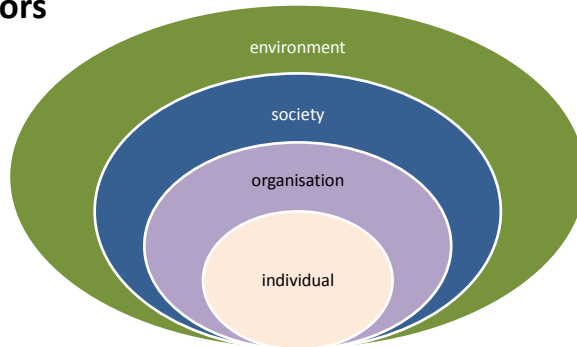


(3) Stakeholders...?

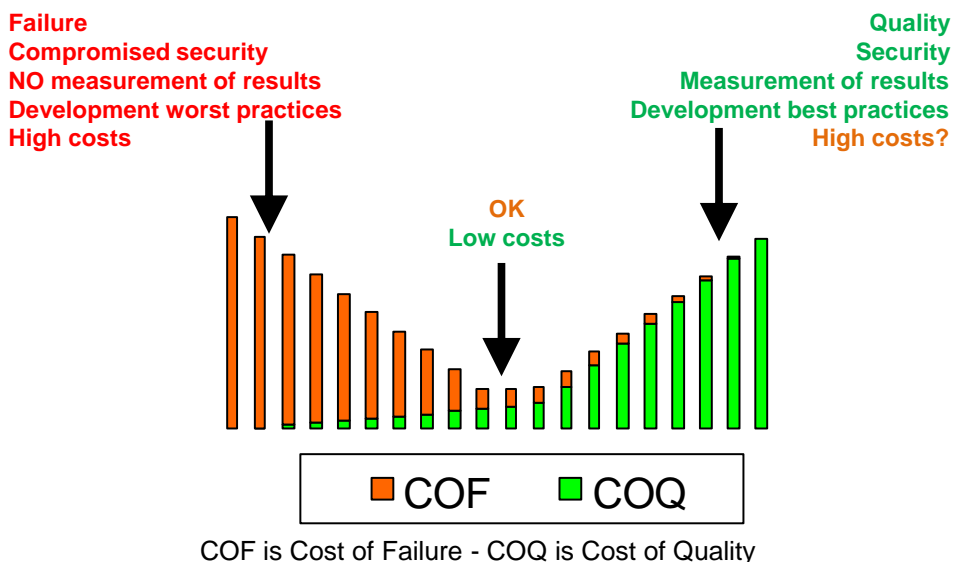


Technical measures become corporate – financial, marketplace, and societal

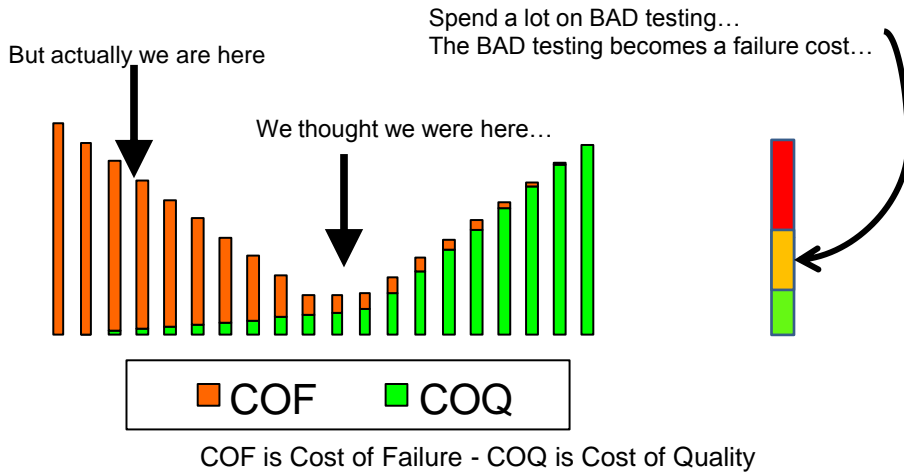
- We MEASURE the same things
- We analyse those measures more thoroughly
- We build up metrics and predictive indicators
- We REPORT indicators



Financial results: What is your cost of failure? Comparing organisations

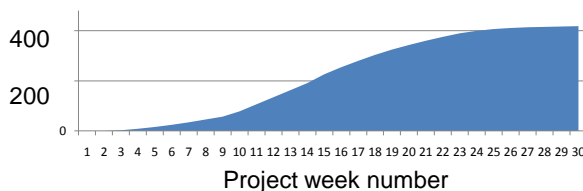


Financial results: What is your cost of failure? Comparing organisations – other possibilities...



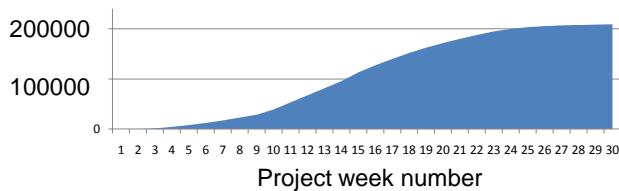
Financial and value quality –
tester’s “S curve” – is it enough?

Typically, testers will track an S curve
or similar model to show the
number of defects found
peaking and tailing off...

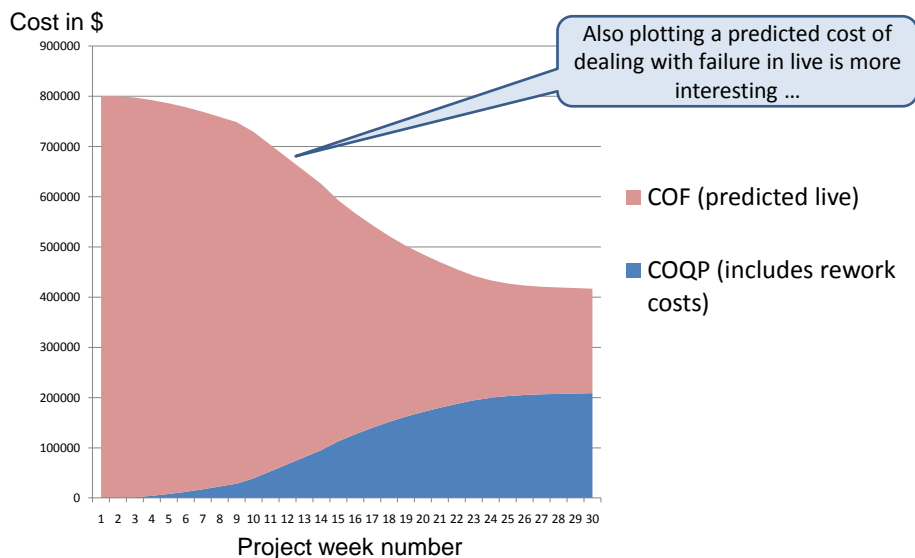


Financial and value quality – tester’s “S curve” – is it enough?

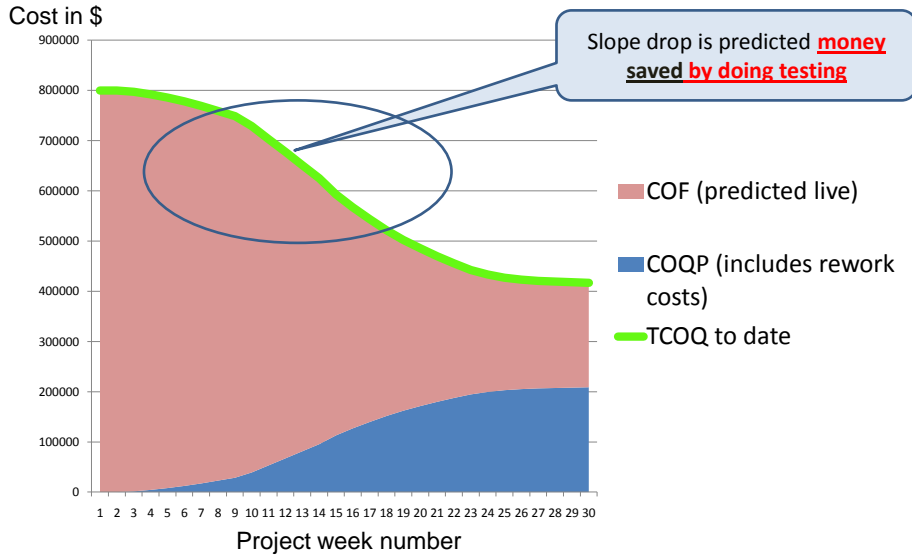
Could show it as a cost of testing and
rework (as fewer defects are found
the rework and retest costs
smooth out...)



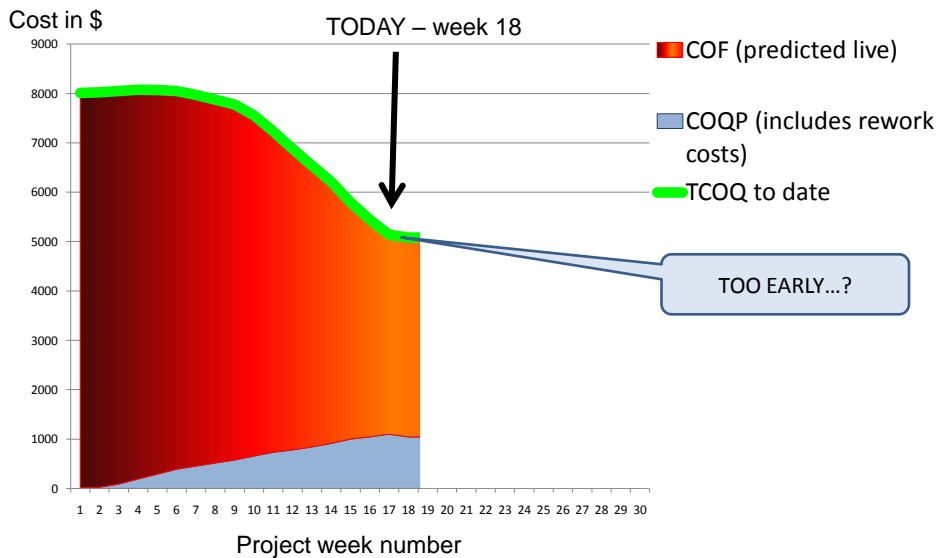
Financial and value quality – project estimation of cost of failure



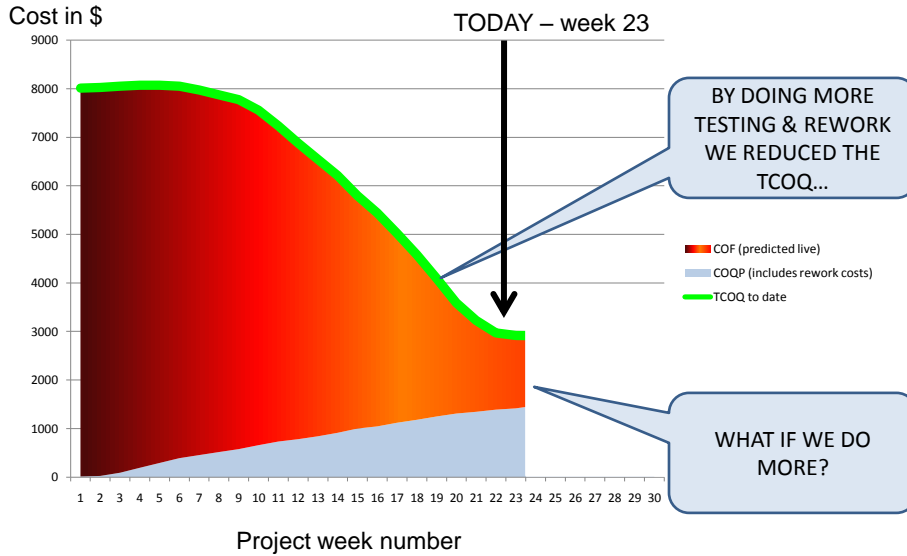
Financial and value quality – project estimation of TCOQ



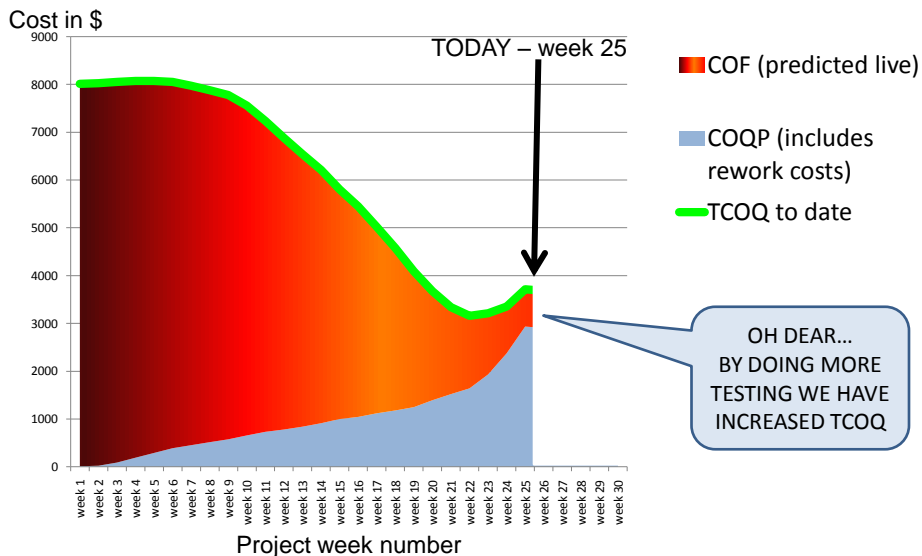
Example TCOQ: testing in progress – shall we go live yet?



TCOQ – testing in progress – shall we go live yet?



TCOQ – testing in progress – shall we go live yet?



Defect predictions/potentials and defect removal efficiency

Defect potential (after Capers Jones)		Language	Code size	Non-code defects	Code defects	Total defects
		C	125,000	3000	2000	5000

Project A	Non-code defects	Code defects	Total defects	Removed	Leaving	Removal efficiency
Inspections	2550	800	3350	3350 / 5000	450 +1200	67%
Static analysis	0	800	800	800 / 1200	450+400	66%
Testing	200	300	500	500 / 850	250+100	58%
Pre release	2750	1900	4650	4650/5000	250+100	93%
Post release	250	100	350			

Project B	Non-code defects	Code defects	Total defects	Removed	Leaving	Removal efficiency
Testing	1600	1500	3100	3100/5000	1900	62%
Post release	1400	500	1900			

Made into time and money

Defect potential (after Capers Jones)		Language	Code size	Non-code defects	Code defects	Total defects
		C	125,000	3000	2000	5000

	Project A COST	Project B COST
Inspections	\$168,750	\$0
Static analysis	\$81,250	\$0
Testing	\$150,000	\$775,000
Pre release	£400,000	\$775,000
Post release	\$175,000	\$950,000
Total cost	\$575,000	\$1,725,000

An advantage of working on maintenance is that we can control through comparison and prediction

	# defects release 1	~ %age	# defects release 2	~ %age	# defects release 3	~ %age	
req's review	0	0	60	15	80	27	
design review	0	0	40	10	70	20	
code review	0	0	20	5	50	13	
test defects	200	57	200	48	100	27	
live failures	150	43	100	22	50	13	
total	350	100	420	100	350	100	

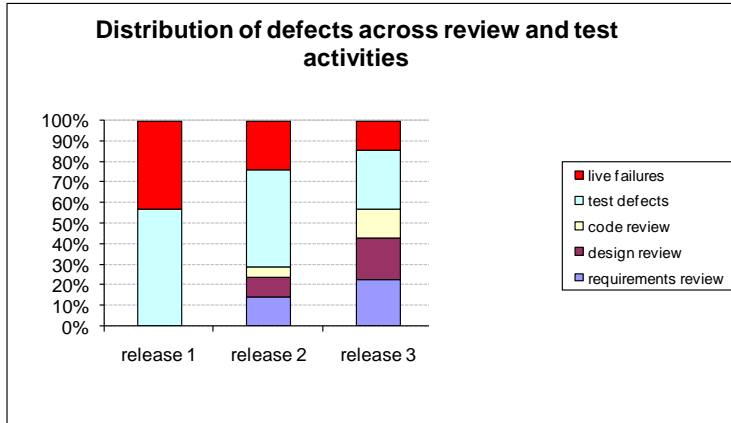
defects by LOC	release 1	release 2	release 3	prediction release 4	This is the size prediction for the next project – we can make a defect prediction
size	15000	20000	15000	15000	
defects	350	420	350	360	
defects by LOC	43 appr.	48 appr.	43 appr.	42 appr.	

An advantage of working on maintenance is that we can control through comparison and prediction

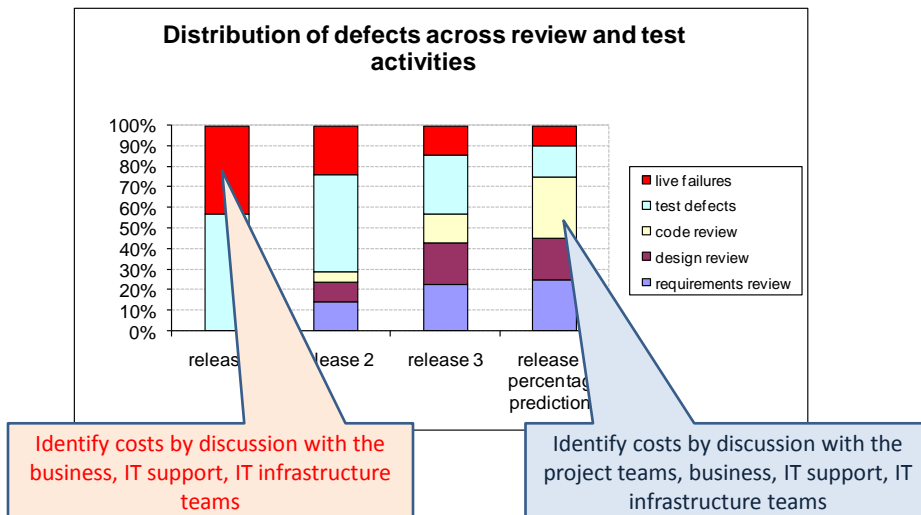
	# defects release 1	~ %age	# defects release 2	~ %age	# defects release 3	~ %age	release 4 hunting target
req's review	0	0	60	15	80	27	25%= 90
design review	0	0	40	10	70	20	20%=72
code review	0	0	20	5	50	13	30%=108
test defects	200	57	200	48	100	27	15%=54
live failures	150	43	100	22	50	13	10%=36
total	350	100	420	100	350	100	360

defects by LOC	release 1	release 2	release 3	prediction release 4	We can set hunting targets for the test activities
size	15000	20000	15000	15000	
defects	350	420	350	360	
defects by LOC	43 appr.	48 appr.	43 appr.	42 appr.	

Control through comparison and prediction



Control through comparison and prediction



Example (a)

- **Role:** Project managers
- **Financial goal for the project:** To reduce cost / time
- **Questions they ask**
 - Cost and time of last test project:
 - How much did it cost and how long did it take?
 - Why was it so expensive last time?
 - Why did it take so long?
 - For the next test project:
 - Can I have it cheaper and better?
 - How much effort will it take?
 - When will you finish?
 - What resources do you need?
 - And what will those cost?
- Note possibly conflicting **quality goal** for the project

Example (b)

- **Metrics they would find useful**
 - Cost of testing: preparation/execution/managing; supporting activities; early preventative activities...
 - Cost of failure to test efficiently: e.g. man-days lost waiting for environment...
 - Cost of failure to test effectively: e.g. time wasted doing or inappropriate wrong tests
 - Cost of failure: cost of defects/failures in live systems
 - Money saved by doing testing

Example (c)

- Test measures, metrics and indicators that can be collected and translated
 - Number of **significant defects** found during previous release testing, **cost** to repair those defects
 - Number of **failures in live and cost** to repair the damage from those defects
 - **Time wasted**
 - **Estimated cost** of tests that **would have** found those problems
 - **Size of previous release and size of next release**
 - **Predicted # defects in next release** (see Capers Jones)
 - **Predicted cost to find those defects** (see Capers Jones)
 - **Predicted cost of not finding those defects** (see Capers Jones)

Example (d)

	# defects release 1	~ %age	# defects release 2	~ %age	# defects release 3	~ %age	release 4 hunting target
req's review	0	0	60	15	80	27	25%= 90
design review	0	0	40	10	70	20	20%=72
code review	0	0	20	5	50	13	30%=108
test defects	200	57	200	48	100	27	15%=54
live failures	150	43	100	22	50	13	10%=36
total	350	100	420	100	350	100	360

defects by LOC	release 1	release 2	release 3	prediction release 4	We have numbers from our projects
size	15000	20000	15000	15000	
defects	350	420	350	360	
defects by LOC	43 appr.	48 appr.	43 appr.	42 appr.	

Defect potential (after Capers Jones)
compared with our projects

Example (e)

Language	Code size	Non-code defects	Code defects	Total defects	Defects/kloc
Example C	125,000	3000	2000	5000	5-25 (Capers Jones)
Our C project	15,000			360	40 (our history)

It looks like the code is about 8 times smaller

We have more defects per KLOC than the worked Capers Jones example

COSTS	Capers Jones A		our project	Capers Jones B		our project
	Capers Jones A	cost per LOC	with static test	Capers Jones B	cost per LOC	without static test
inspections	168,750.00	1.35	20,250.00	-	-	-
static analysis	81,250.00	0.65	9,750.00	-	-	-
dynamic test stages	150,000.00	1.20	18,000.00	775,000.00	6.20	93,000.00
total pre release	400,000.00	3.20	48,000.00	775,000.00	6.20	93,000.00
total post release	175,000.00	1.40	21,000.00	950,000.00	7.60	114,000.00
total cost	575,000.00	4.60	69,000.00	1,725,000.00	13.80	207,000.00

Example (f)

TIMESCALES	CJ A	Our project +ST	CJ B	Our project no ST
development schedule (months)	12	1.44	16	1.92
LIVE FAILURES				
Total defects	5000	600	5000	600
found by static test	4150	498	0	0
found by dynamic test	500	60	1600	192
found post release	350	42	1900	228

Note: this table uses Capers Jones' figures, but history shows our defect per LOC to be about twice as bad as that

See also spreadsheet for a starting point for your calcs...

Caveat! review first - do not just use – jetlag related errors may be in it!

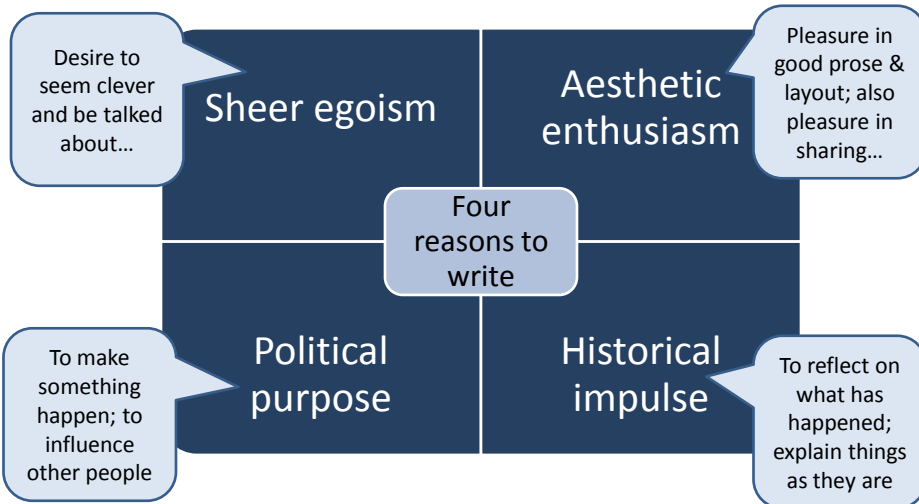
Example (g)

Get started early and...
we can spend 1.5 months and
48,000 dollars before release
for a total cost of 69,000 dollars,
with 40-80 defects reported post live

or get the testers in late...
and we can spend 2
months and 93,000
dollars before release
for a total cost of 207,000
dollars
with 200-500 defects
reported post live

Your choice....

George Orwell "Why I write" (essay)



Dashboards can be too simple

The size of the group
and the need for
hierarchy dictates
what is appropriate



Graphics show old Ford
With simple dashboard
and controls
Still drivable?



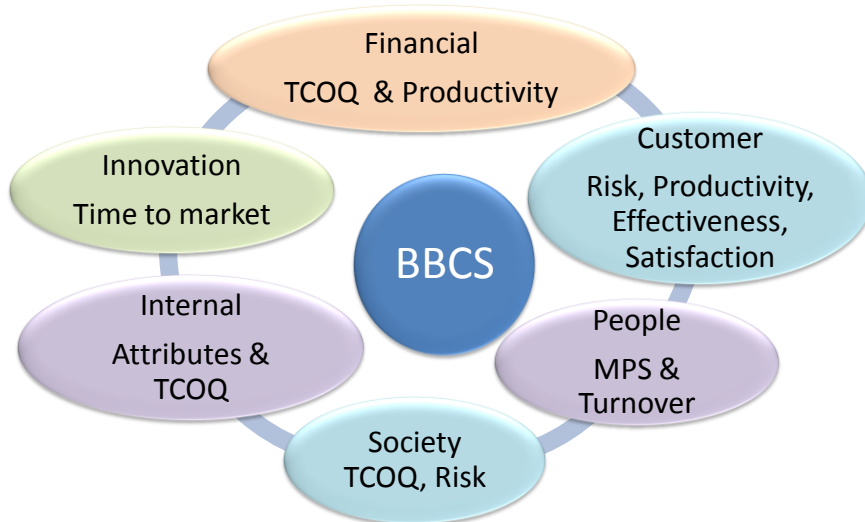
A picture tells a thousand words...

Unless you can't see the
picture...

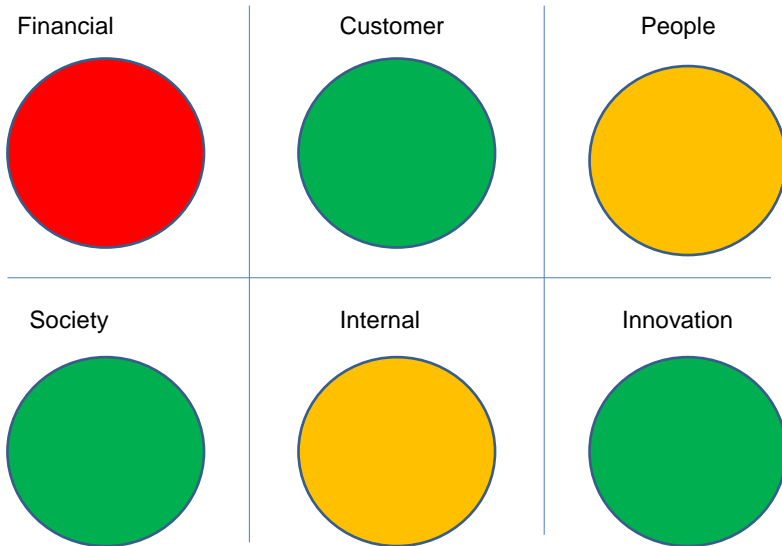
What I am learning
already about my new
role...



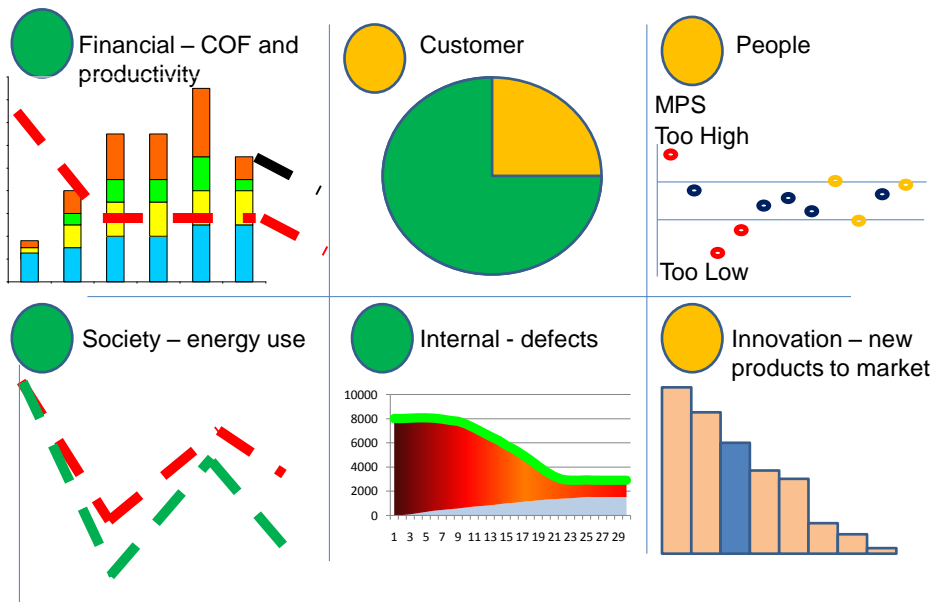
Dashboard A: Combined scorecard – Testing



Dashboard B: traffic lights



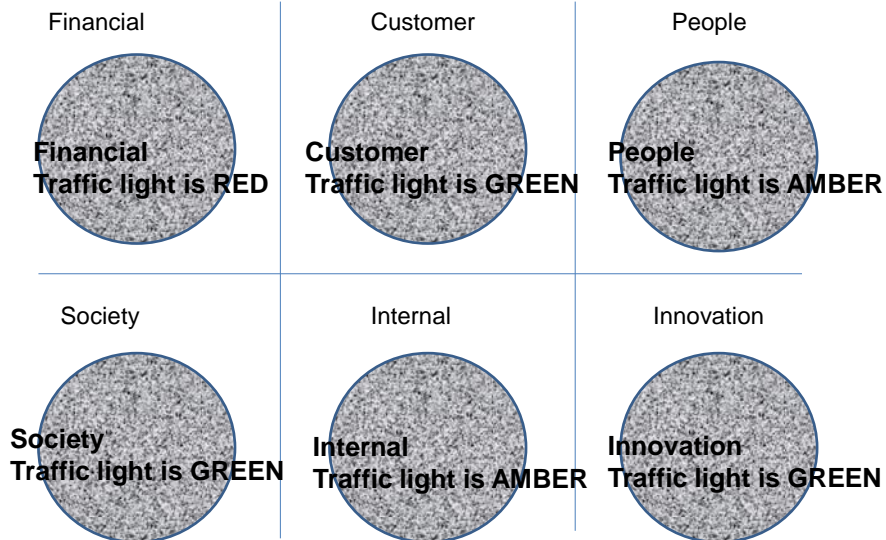
Dashboard C



Dashboards 2 and 3 are not readable with a screen reader – additional text is required
 This is a new skill and consideration for me!

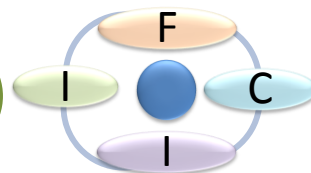
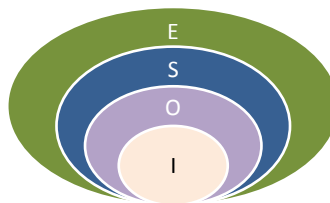
Dashboard D

ALT FORMAT



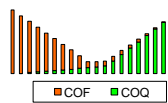
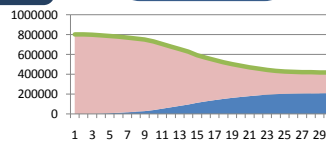
Key points:

Cutbacks **Poor SW engineering**
Recession **Flawed economic models**

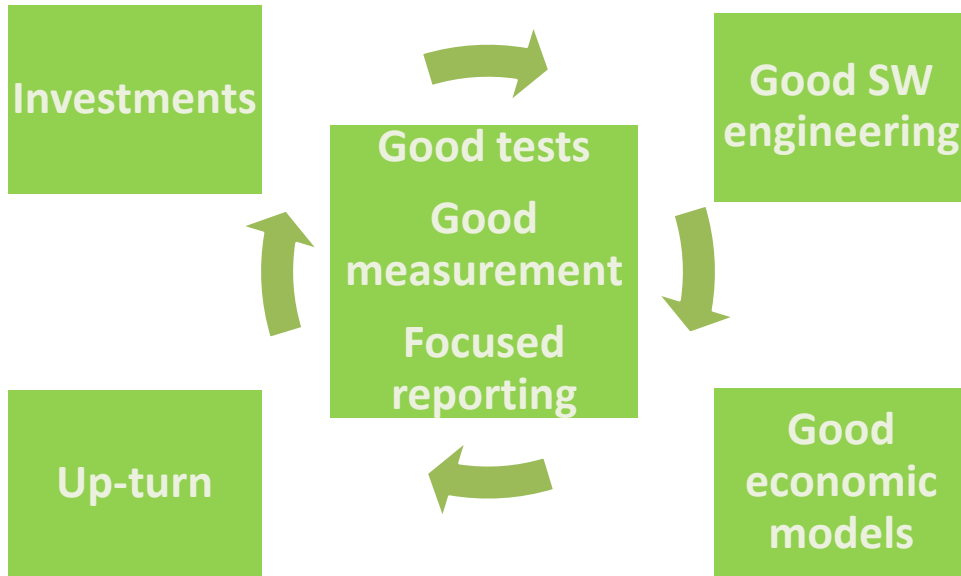


WHAT to measure

HOW to present it



Measure up to your stakeholders



October 2010

THANK YOU
FOR
LISTENING

Isabel Evans
Isabel.evans@bcs.org

Apologies for the low level of tagged graphics – I am still learning how to do this...!

References and thanks

- Capers Jones “Software Engineering Best Practice”
- Jane Austin “Mansfield Park”
- George Orwell “Why I Write”
- Kaplan & Norton “Balanced Scorecard” & “Alignment”
- Norman Lindsay “The Magic Pudding”
- Margaret Attwood “Payback”

Excellence models :

- <http://www.excellencemodels.org/ExcellenceModels/tabid/53/Default.aspx>
- <http://www.saiglobal.com/business-improvement/process/framework/excellence.htm>

Thanks and discussion:

- David Hayman & Stuart Reid for ideas and help
- *UK Testers' Retreat* for discussions
- Some of this material was developed while Isabel worked for Testing Solutions Group Ltd, and is available in the TSG tutorial “Measuring up to your stakeholders” written by Isabel, for TSG
- Isabel now works for Dolphin Computer Access