

# Does real Software Practice Advancement need yet another 'Manifesto'?

\_"AGILE HAS DOOMED ITSELF - TO BECOME YET ANOTHER FAD: XP IS ALREADY DEAD.

What is Seriously Wrong with Agile practices and interpretations - why AGILE, AS CURRENTLY PRACTICED, is PROJECT-failure-prone as a culture

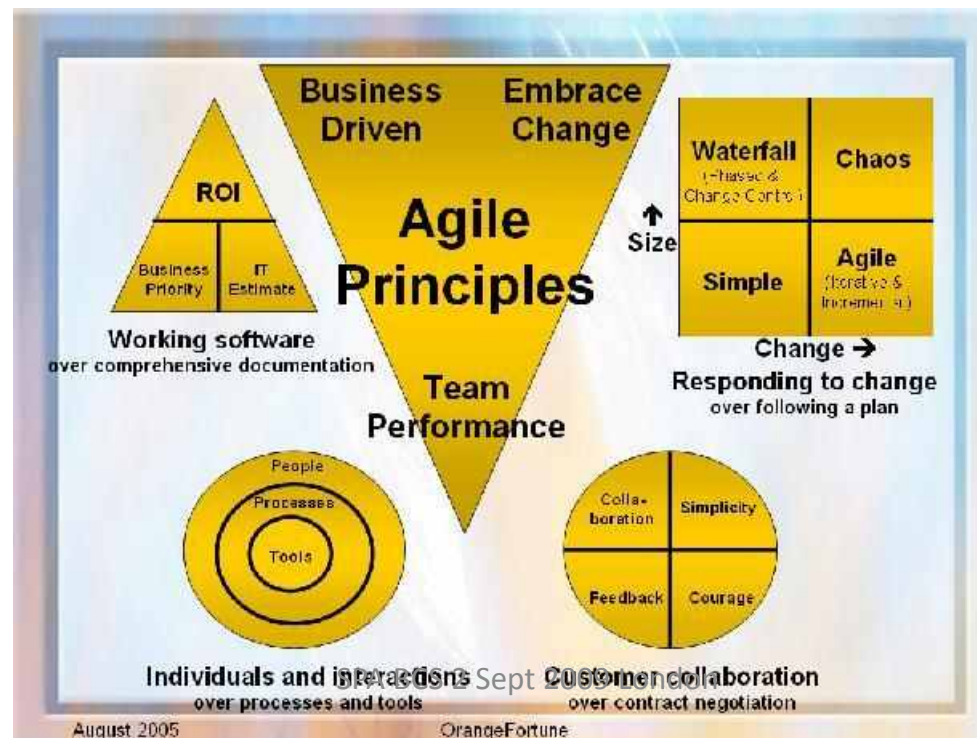
"What is Tom's advice, his own more value-oriented 'agile' principles and values (see below) and metrics-oriented agile practices in Evo?

The SHORT talk will be followed by a debate and questions and answers:

**challenge** the assumptions stated by Tom about Agile weaknesses

**suggest** additional weaknesses with agile and specific practices

**ask** any questions about specific practices



September 12, 2009

# Gilb's Ten Key Agile Principles

to avoid bureaucracy and give creative freedom!

**Control projects by quantified critical-few results. 1 Page total !**

(not stories, functions, features, use cases, objects, ..)

**Make sure those results are business results, not technical**

**Align your project with your financial sponsor's interests!**



**Give developers freedom, to find out *how* to deliver those results**

**Estimate the impacts of your designs, on *your* quantified goals**

**Select designs with the best impacts for their costs, do them first.**

**Decompose the workflow, into weekly (or 2% of budget) time boxes**

**Change designs, based on quantified experience of implementation**

**Change requirements, based in quantified experience, new inputs**

**Involve the stakeholders, every week, in setting quantified goals**

**Involve the stakeholders, every week, in *actually using* increments**

# My 10 Agile *Values*?

- " **Simplicity**
  - " 1. Focus on real stakeholder values
- " **Communication**
  - " 2. Communicate stakeholder values quantitatively
  - " 3. Estimate expected results and costs for weekly steps
- " **Feedback**
  - " 4. Generate results, weekly, for stakeholders, in their environment
  - " 5. Measure all critical aspects of the improved results cycle.
  - " 6. Analyze deviation from your initial estimates
- " **Courage**
  - " 7. Change plans to reflect weekly learning
  - " 8. Immediately implement valued stakeholder needs, next week
    - " *Don't wait, don't study (analysis paralysis), don't make excuses.*
    - " *Just Do It!*
  - " 9. Tell stakeholders exactly what you will deliver next week
  - " 10. Use any design, strategy, method, process that works quantitatively well - to get your results
    - " Be a systems engineer, not a just programmer (a 'Softcrafter').
    - " Do not be limited by your craft background, in serving your paymasters

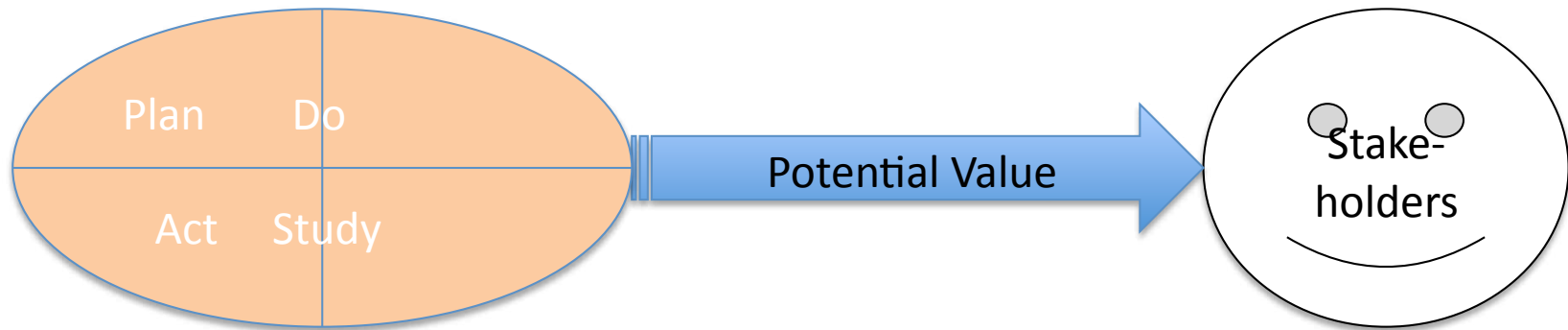


## **Gilb's 'Value Driven Planning' Principles:**

- 1. Critical Stakeholders determine the values**
- 2. Values can and must be quantified**
- 3. Values are supported by Value Architecture**
- 4. Value levels are determined by timing, architecture effect, and resources**
- 5. Value levels can differ for different scopes (where, who)**
- 6. Value can be delivered early**
- 7. Value can be locked in incrementally**
- 8. New Values can be discovered (external news, experience)**
- 9. Values can be evaluated as a function of architecture (Impact Estimation)**
- 10. Value delivery will attract resources.**



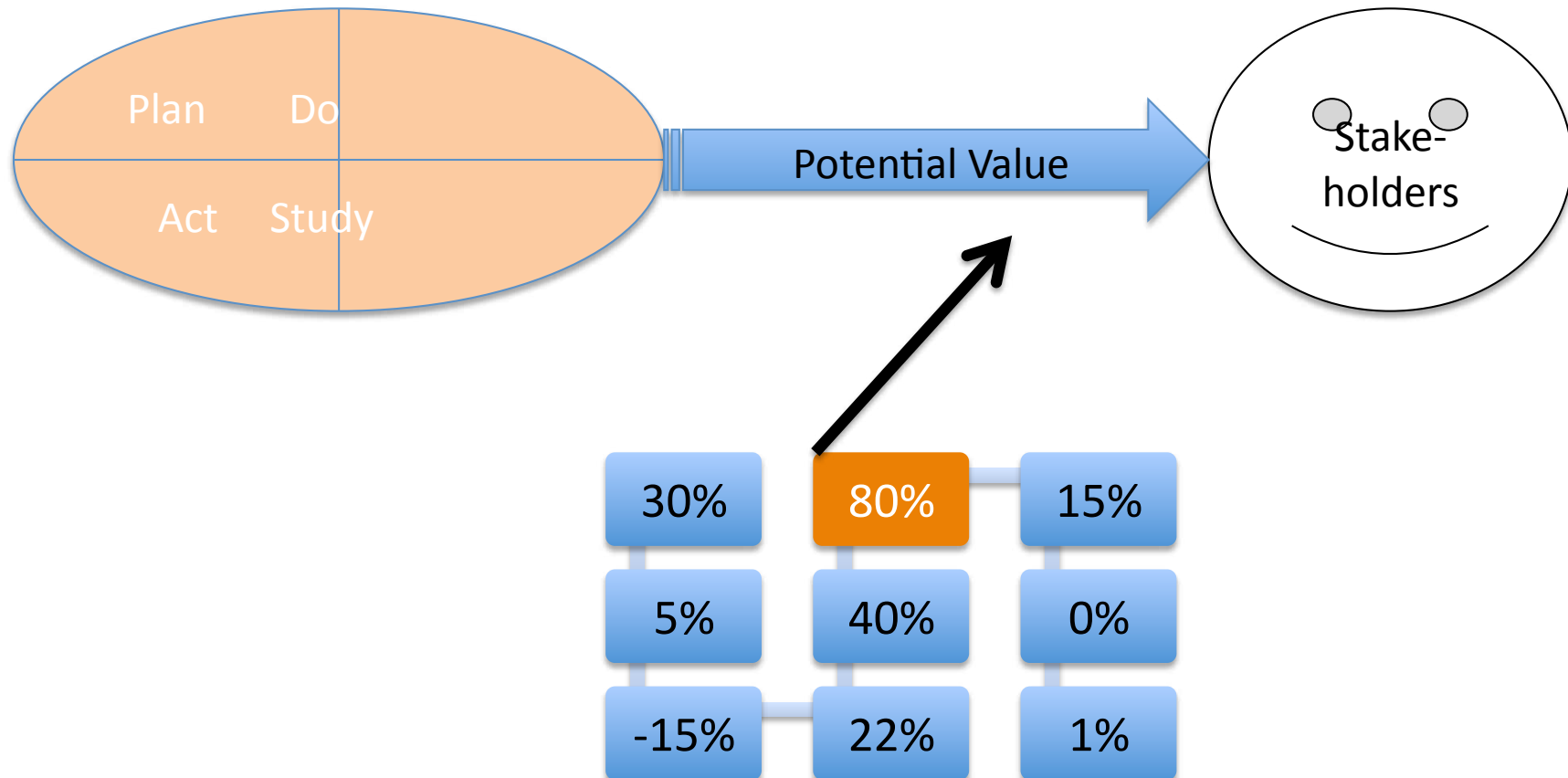
# Primary Evo Concept: Deliver *Potential* Value



The Evo Cycle:  
Viewed as a Deming PDSA Cycle

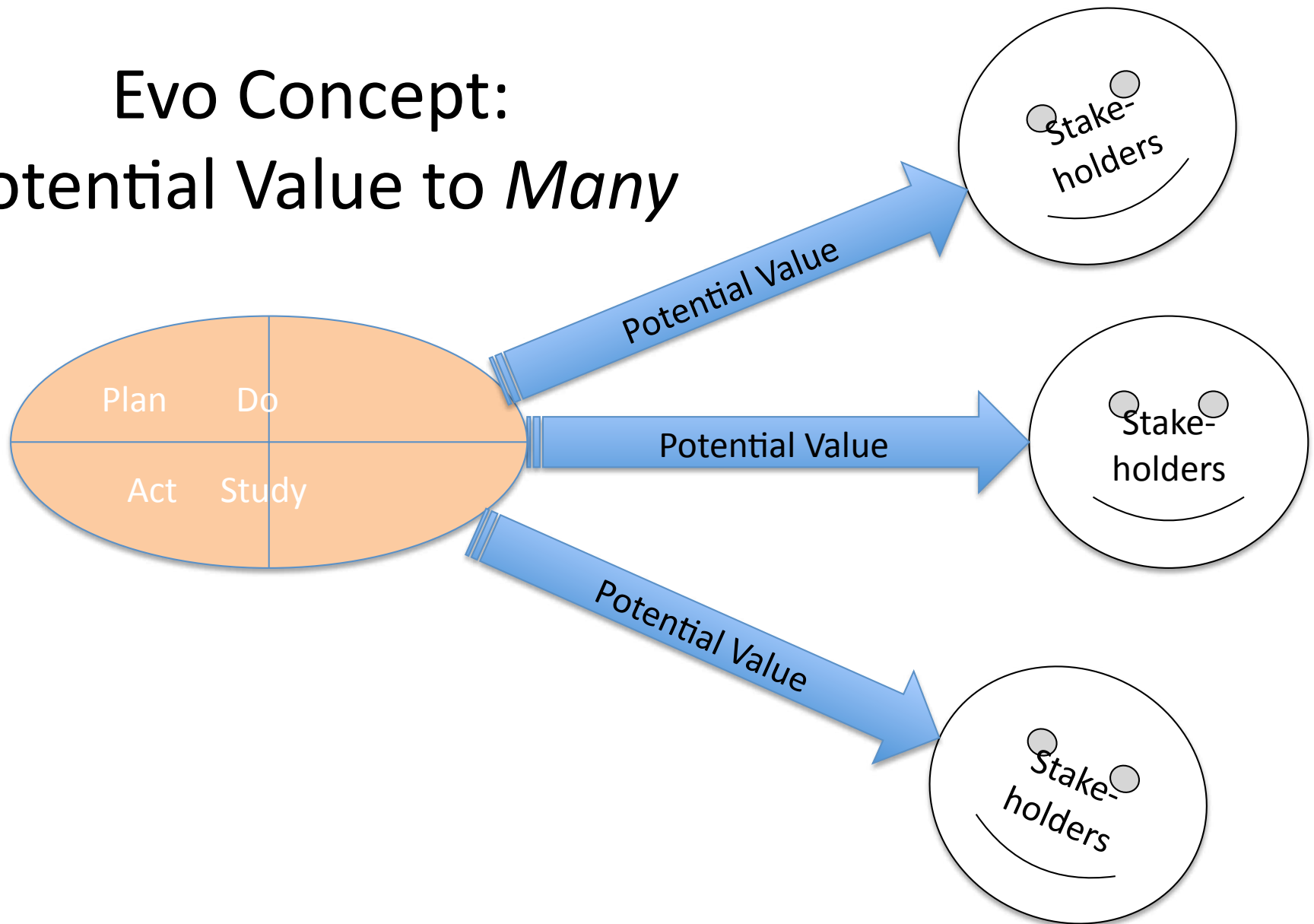
- " Incremental Value Delivery to Stakeholders

# Deliver the highest value for resources



**HIGHEST AVAILABLE Incremental Value Delivery to Stakeholders**

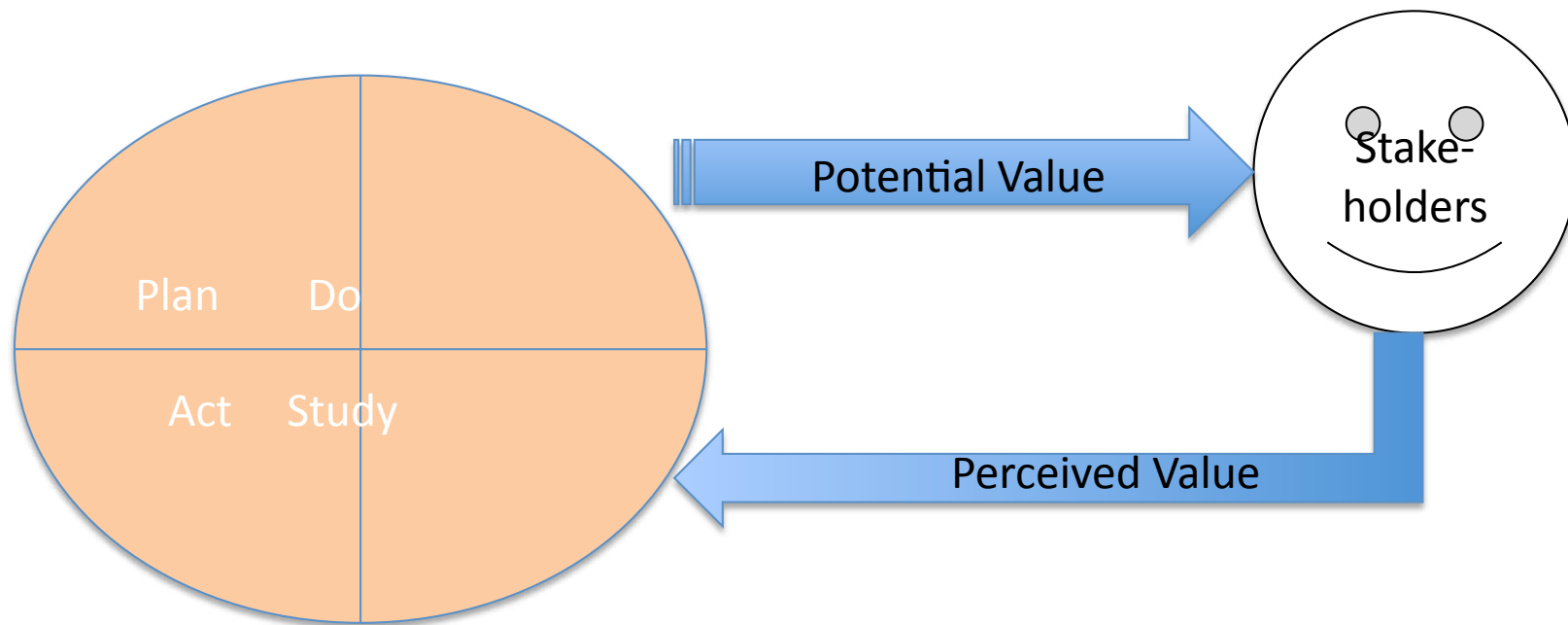
# Evo Concept: Potential Value to *Many*



- " Incremental Value Deliveries to ***Many*** Stakeholders

# Evo Concept: Short Term Feedback

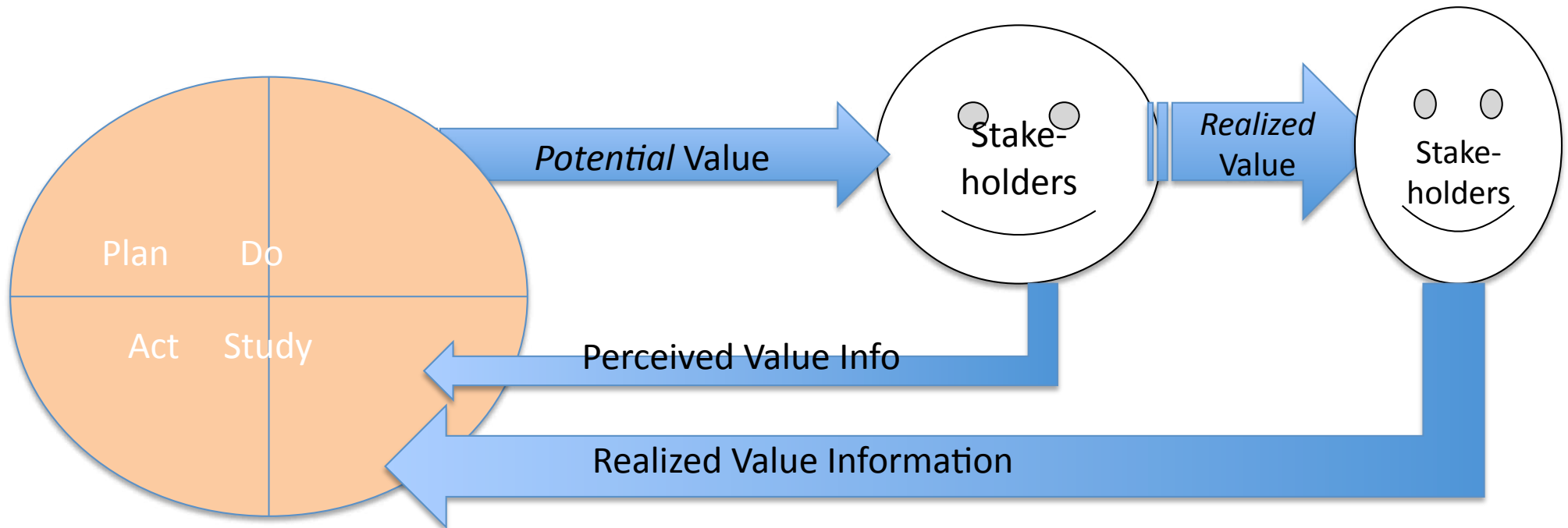
*“This looks like a change I can get value from!”*



- " Initial Feedback from Stakeholders, after Evo Cycle delivery

# Long-Term *Real* Value Feedback

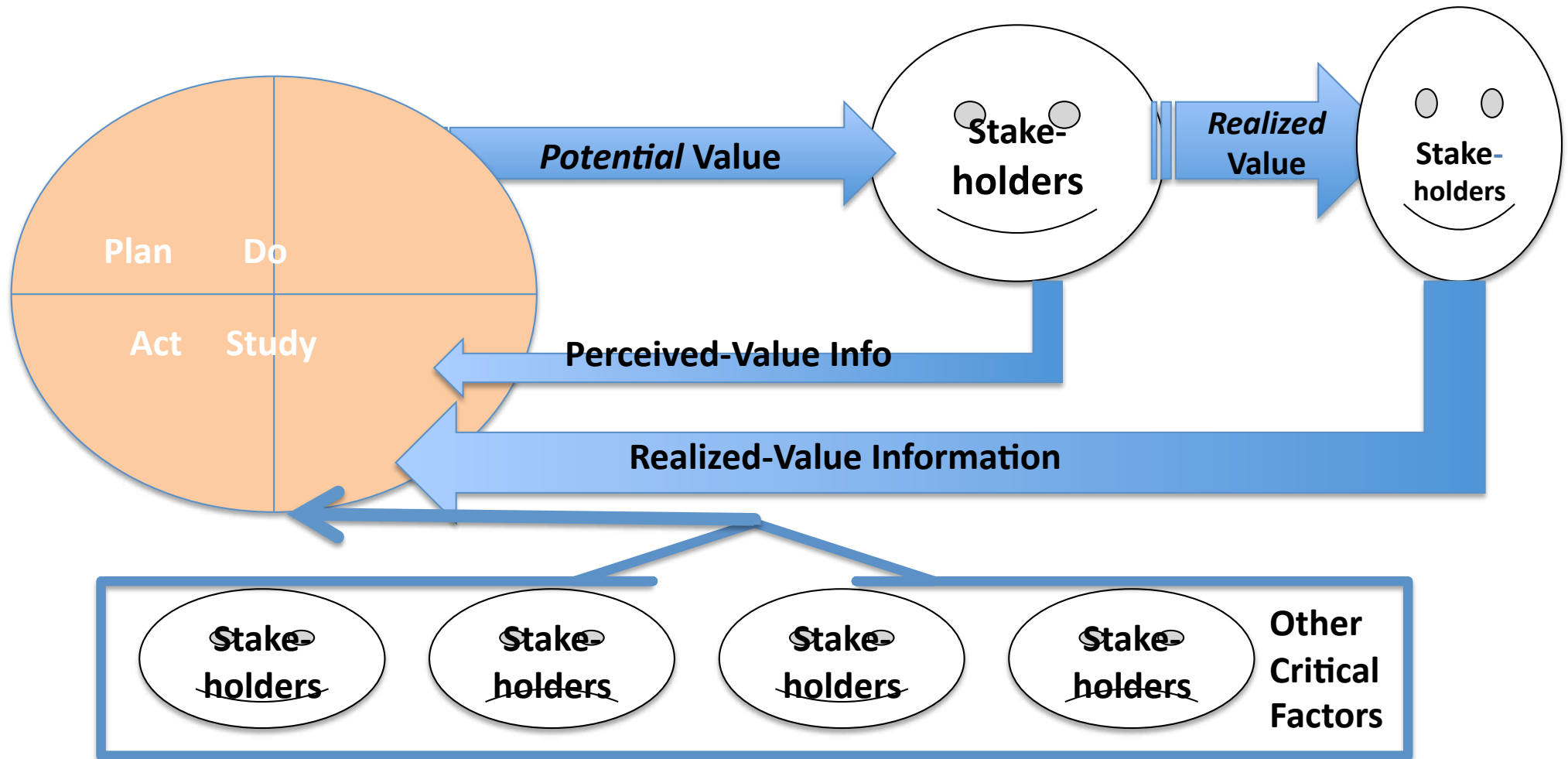
*"This is **the real value** we have gotten to date, and what we **expect** to get in the future!"*



- " 2 Kinds of Feedback from Stakeholders, when value increment is *really* exploited in practice after delivery

# Study critical factors in your environment

*"Budget cut, Deadline nearer, New CEO, Cheaper Technology"*



- 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.

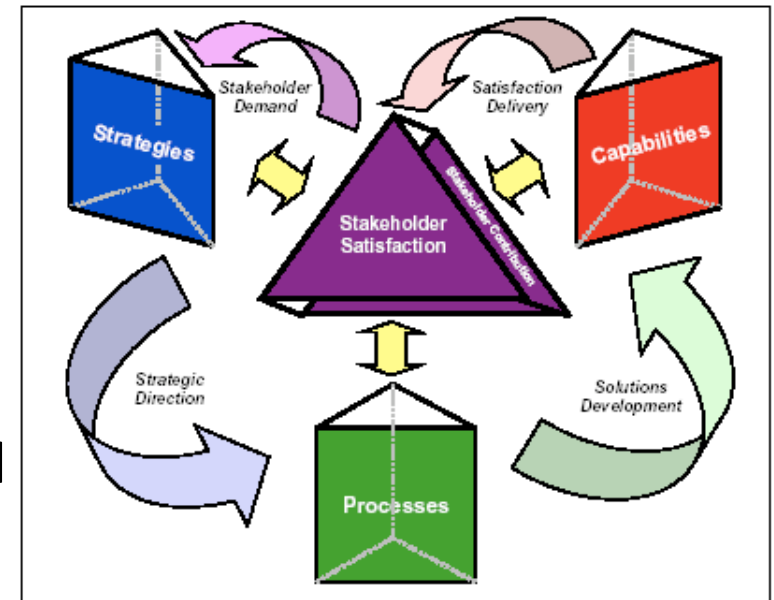


# Value Driven Planning Principles in Detail:

# 1. Critical Stakeholders determine the values

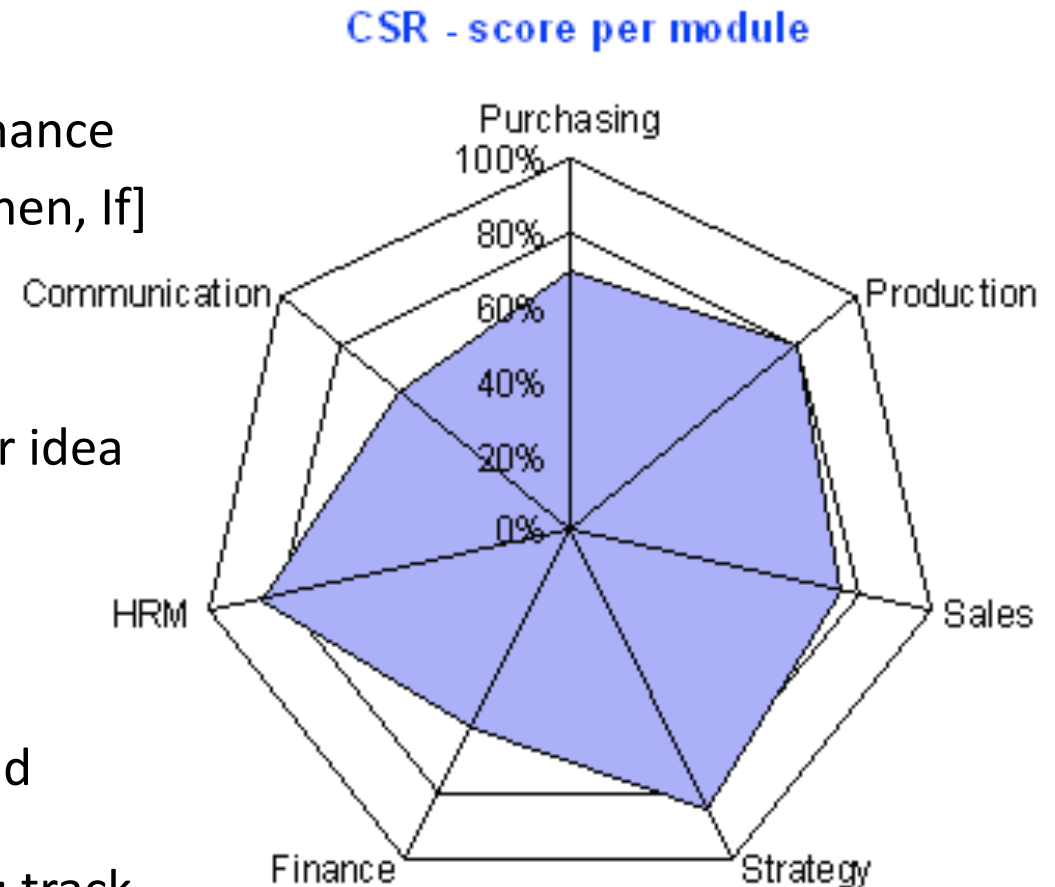
Critical: “having a decisive or crucial importance in the success or failure of something” <-Dictionary

- " The primary and prioritized values we need to deliver are determined by
  - " analysis of the needs and values of stakeholders
    - " stakeholders who can determine whether we *succeed or fail*.
- " We cannot afford to satisfy *other (less critical)* levels, at other times and places, yet.
  - " Because that might undermine our ability to satisfy the more critical stakeholders –
  - " and consequently threaten our overall project success.



## 2. 'Values' can and must be *quantified*

- " Values can, if you want, be expressed numerically.
  - " With a defined scale of measure
  - " with a deliverable level of performance
  - " and with qualifier info [Where, When, If]
- " Quantification is useful:
  - " to clarify your own thoughts
  - " to get real agreement to one clear idea
  - " to allow for varied targets and constraints
  - " to allow direct comparison with benchmarks
  - " to put in Request for bids, bids and contracts
  - " to manage project evolutionarily : track progress
  - " as a basis for measurement and testing
  - " to enable research on methods



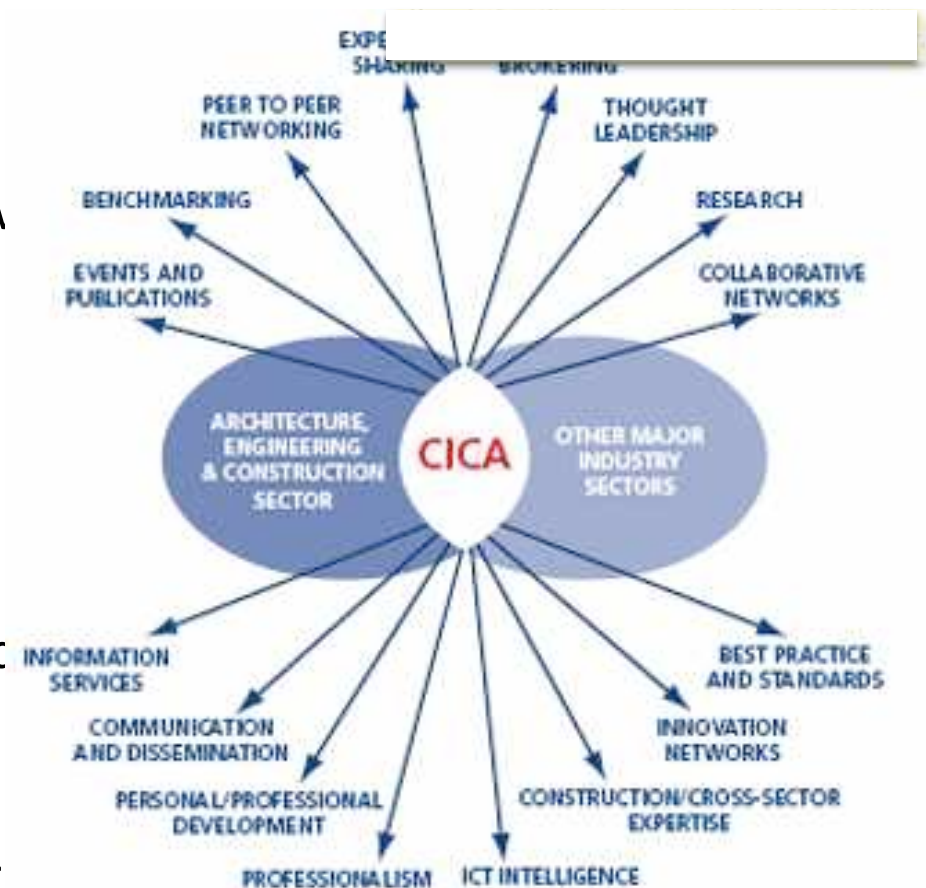
- Figure 1: Real (NON-CONFIDENTIAL version) example of an initial draft of setting the objectives that engineering processes must meet.

Business objective	Measure	Goal (200X)	Stretch goal ('0X)	Volume	Value	Profit	Cash
Time to market	Normal project time from GT to GT5	<9 mo.	<6 mo.	X	X	X	X
Mid-range	Min BoM for The Corp phone	<\$90	<\$30	X	X	X	X
Platformisation Technology	# of Technology 66 Lic. shipping > 3M/yr	4	6	X	X	X	X
Interface	Interface units	>11M	>13M	X	X	X	X
Operator preference	Top-3 operators issue RFQ spec The Corp	1	2	X	X	X	X
Productivity				X	X	X	X
Get Torden	Lyn goes for Technology 66 in Sep-04	yes		X	X	X	X
Fragmentation	Share of components modified	<10%	<5%	X	X	X	X
Commoditisation	Switching cost for a UI to another System	>1yr	>2yrs	X	X	X	X
Duplication	The Corp share of 'in scope' code in best-selling device	>90%	>95%	X	X	X	X
Competitiveness	Major feature comparison with MX	Same	Better	X	X	X	X
User experience	Key use cases superior vs. competition	5	10	X	X	X	X
Downstream cost saving	Project ROI for Licensees	>33%	>66%	X	X	X	X
Platformisation IFace	Number of shipping Lic.	33	55	X	X	X	X
Japan	Share of of XXX sales	>50%	>60%	X	X	X	X
Numbers are intentionally changed from real ones							

Business Values Quantified

### 3. Values are supported by Value Architecture

- " Value Architecture: defined as:
  - " anything you *implement* with a view to satisfying stakeholder values.
- " Value Architecture:
  - " includes product/system objectives
    - " Which are a 'design' for satisfying stakeholder values
  - " Has a multitude of performance and cost impacts
  - " can impact a given system differently, depending on what is in the system, or what gets put in later
  - " Needs to try to maximize value delivered for resources used.

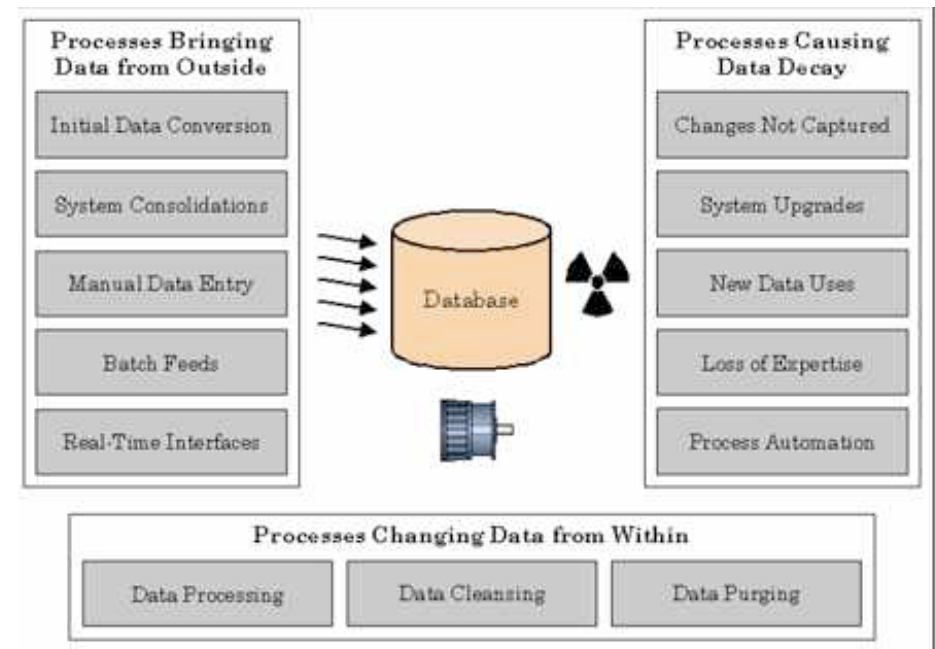


#### 4. Value levels are determined by *timing, architecture effect, and resources*

Value levels: defined as:  
the degree of satisfaction of value needs.

Value level:

- " depends on *when* you observe the level
  - " The environment, the people, other system performance characteristics (security, speed, usability)
- " depends on the *current incremental power of particular value architecture* components
- " depends on *resources available* both in development and operation



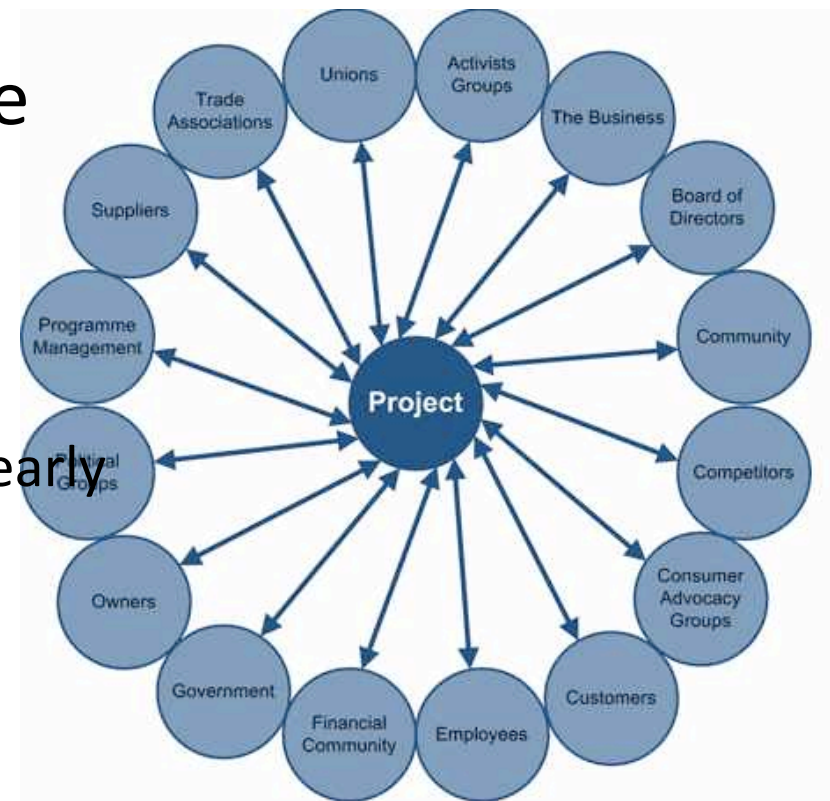


## 5. Required Value *levels* can differ for different scopes (where, who)

The level of value needed, and the level of value delivered - for a single attribute dimension (like Ease of Use) can vary for:

- " different stakeholders
- " at different times
  - " (peak, holiday, slack, emergency, early implementation)
- " for different 'locations'
  - " countries, companies, industries

There is nothing simple like 'one level for all'



- 6. Value can be delivered early

You do not have to wait until 'the project is done' to deliver useful stakeholder value satisfaction.

You can intentionally target the highest priority stakeholders, and their highest priority value area, and levels.

You can deliver them early and continuously

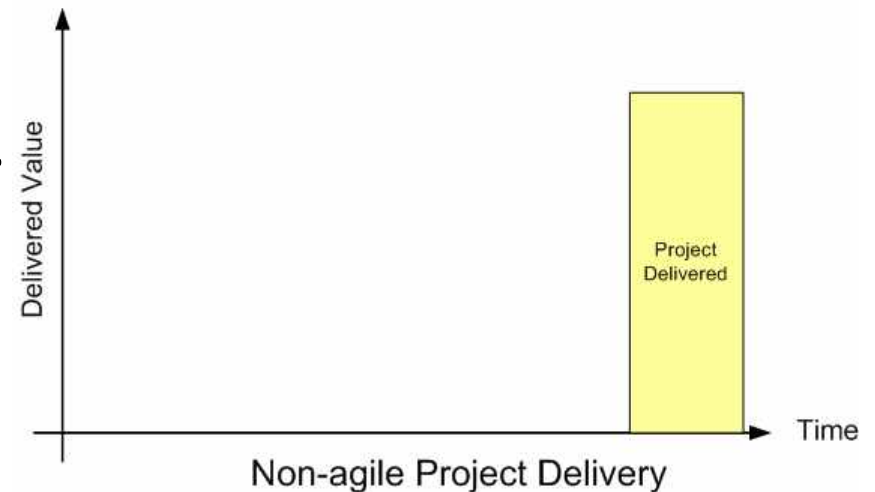
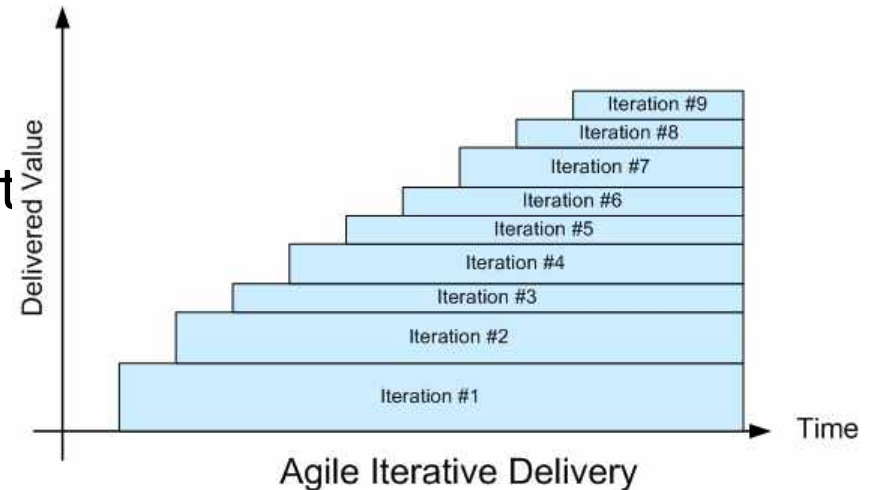
You can learn what is possible

And what stakeholders really value.

Discover new value ideas

Discover new stakeholders

Discover new levels of satisfaction



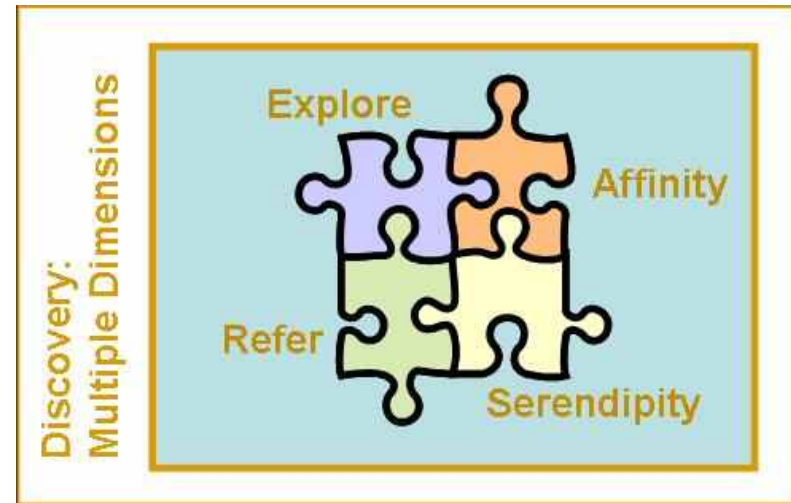
- 7. Value can be locked in incrementally

- " You can increment the value satisfaction
  - "*towards* longer term Goal levels
- " You can spread the value deliveries
  - " that are *proven* in *some* places,
  - "more widely in the next increments
- " This probably assumes that you have really handed over real results to real people.
  - "Not just developed systems without delivery



## 8. New Values can be discovered (external news, experience)

- " *Expect*, and try to discover,
  - "entirely new stakeholder values.
- " These will of course emerge *after you start delivering* some satisfaction, because:
  - "Stakeholders believe you can help
  - "Things *change*



## 9. Values can be *evaluated* as a function of *architecture* (using 'Impact Estimation')

- It is possible to get an **overview** of

- the totality of impacts
- that your **architecture**
- (all designs and strategies)
- **might** have
- on all your defined stakeholder needs

Business Objective	Weight	Viking Deliverables											
		hardware adaptation	Telephony	Reference designs	IFace	Modularity	Defend vs Technology 66	Tools	User Experience	GUI & Graphics	Security	Defend vs OCD	Enterprise
Time to market	20%	20%	10%	30%	5%	10%	5%	15%	0%	0%	0%	5%	5%
Mid-range	10%	15%	0%	15%	0%	30%	15%	5%	10%	5%	5%	0%	0%
Platformisation Technology	5%	25%	10%	30%	0%	0%	10%	0%	5%	0%	10%	0%	5%
Interface	5%	5%	15%	15%	0%	5%	0%	5%	0%	0%	10%	0%	10%
Operator preference	10%	0%	10%	0%	15%	5%	20%	5%	10%	10%	20%	5%	10%
Get Torden	10%	25%	10%	10%	-10%	0%	20%	0%	10%	-20%	10%	10%	5%
Commoditisation	5%	20%	10%	20%	10%	-20%	25%	15%	0%	0%	5%	10%	5%
Duplication	10%	15%	10%	10%	0%	0%	40%	0%	0%	0%	5%	20%	5%
Competitiveness	5%	10%	15%	20%	0%	10%	20%	10%	10%	20%	10%	10%	10%
User experience	5%	5%	0%	0%	0%	20%	0%	0%	30%	10%	0%	0%	0%
Downstream cost saving	5%	15%	5%	20%	0%	10%	20%	0%	10%	0%	0%	10%	5%
Platformisation IFace	5%	10%	10%	20%	40%	0%	20%	5%	0%	0%	0%	0%	5%
Japan	5%	10%	5%	20%	0%	10%	0%	0%	10%	5%	0%	0%	0%
Contribution to overall result		15%	9%	17%	4%	7%	15%	6%	6%	1%	6%	6%	5%
Cost (€M)		£ 2.85	£ 0.49	£ 3.21	£ 2.54	£ 1.92	£ 2.31	£ 0.81	£ 1.21	£ 2.68	£ 0.79	£ 0.62	£ 0.60
ROI Index (100=average)		106	358	109	33	78	137	148	107	10	152	202	174

- Use an Impact Estimation table

- and you will be able to spot *opportunities* for
  - high value and
  - low cost early deliveries
  - by analyzing the numbers on the table

See next slide  
For enlargement



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

# Technical Strategies

Objectives		Technical Strategies											
↓ Business Objective Defined In earlier slide		Viking Deliverables											
		hardware adaptation	Telephony	Reference designs	IFace	Modularity	Defend vs Technology 66	Tools	User Expe'ce	GUI & Graphics	Security	Defend vs OCD	Enterprise
Time to market		20%	10%	30%	5%	10%	5%	15%	0%	0%	0%	5%	5%
Mid-range		15%	10%	20%	7%	7%	5%	5%	10%	5%	5%	0%	0%
Platformisation Technology		25%	10%	30%	0%	0%	10%	0%	5%	0%	10%	0%	5%
Interface		5%	15%	15%	0%	5%	0%	5%	0%	0%	10%	0%	10%
Operator preference		0%	0%	0%	0%	0%	20%	5%	10%	10%	20%	5%	10%
Get Torden		25%	10%	10%	-10%	0%	20%	0%	10%	-20%	10%	10%	5%
Commoditisation		20%	10%	20%	10%	-20%	25%	15%	0%	0%	5%	10%	5%
Duplication		15%	0%	10%	0%	0%	40%	0%	0%	0%	5%	20%	5%
Competitiveness		10%	15%	20%	0%	10%	20%	10%	10%	20%	10%	10%	10%
User experience		5%	0%	0%	0%	20%	0%	0%	30%	10%	0%	0%	0%
Downstream cost saving		15%	0%	0%	0%	0%	20%	0%	10%	0%	0%	10%	5%
Platformisation IFace		10%	10%	20%	40%	0%	20%	5%	0%	0%	0%	0%	5%
Japan		10%	5%	20%	0%	10%	0%	0%	10%	5%	0%	0%	0%
Contribution to overall result		15%	9%	17%	4%	7%	15%	6%	6%	1%	6%	6%	5%
Cost (£M)		£ 2.85	£ 0.49	£ 3.21	£ 2.54	£ 1.92	£ 2.31	£ 0.81	£ 1.21	£ 2.68	£ 0.79	£ 0.62	£ 0.60
ROI Index (100=average)		106	358	109	33	78	137	148	107	10	152	202	174



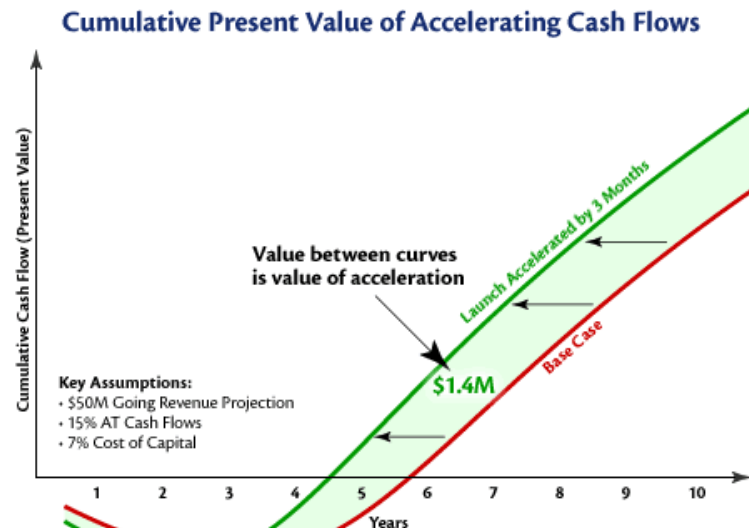
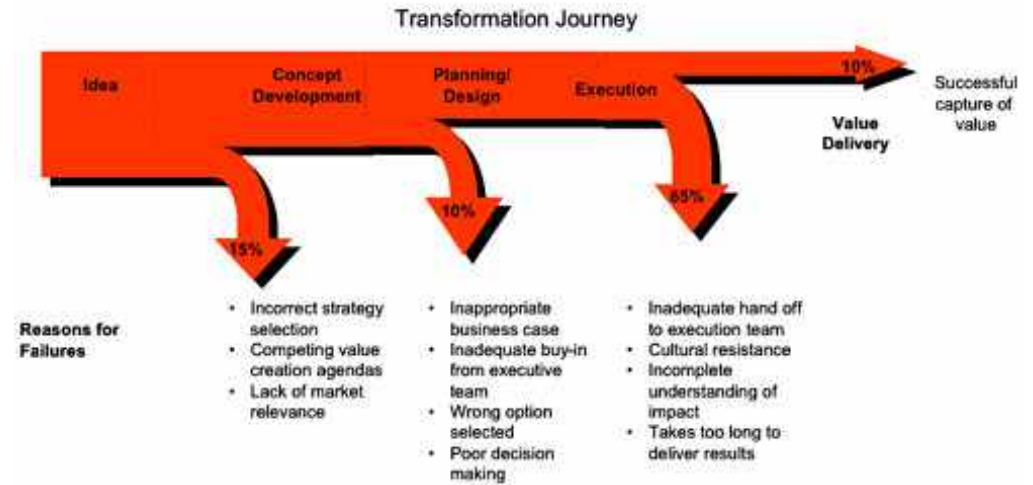
## 10. Value delivery will attract

- " If you are really good at delivering value
  - "You can expect to attract
    - "even more funding
  - "Managers like
    - "to be credited with success
  - " Money seeks
    - "best interest rates



# Gilb's Value Manifesto: A Management Policy?

- 1." Really useful value, for real stakeholders will be defined measurably.  
No nice-sounding emotive words please.
- 2." Value will be seen in light of total long term costs as a decent return on investment.
- 3." Powerful management devices, like motivation and follow-up, will make sure that the value for money is really delivered –  
or that the failure is punished, and the success is rewarded.
- 4." The value will be delivered evolutionarily –  
not all at the end.
- 5." That is, we will create a stream of prioritized value delivery to stakeholders, at the *beginning* of our value delivery projects;  
and continue as long as the real return on investment is suitably large.
- 6." The CEO is primarily responsible for making all this happen effectively.
  - 1." The CFO will be charged with tracking all value to cost progress.
  - 2." The CTO and CIO will be charged with formulating all their efforts in terms of measurable value for resources.



Source "Value Delivery in Systems Engineering" available at [www.gilb.com](http://www.gilb.com)  
 Unpublished paper [http://www.gilb.com/community/tiki-download\\_file.php?fileId=137](http://www.gilb.com/community/tiki-download_file.php?fileId=137)

# The Value Delivery Problem

- " Sponsors who order and pay for systems engineering projects, must justify their money spent based on the expected consequential effects (hereafter called 'value') of the systems.
- "
- " The value of the technical system is often expressed in presentation slides and requirements documents as a set of nice-sounding words, under various titles such as "System Objectives", and "Business Problem Definition"

# Some Assertions

Assertion 1. **When top management allows large projects to proceed, with such badly formulated primary objectives**, then

- " they are responsible as managers for the outcome (failure).
- " They cannot plead ignorance.

Assertion 2. **The failure of technical staff (project management) to react to the lack of primary objective formulation by top management is also a total failure** to do reasonable systems engineering.

- " Management might have a poor requirements culture, but we should routinely save them from themselves.

Assertion 3. **Both top managers and project personnel can be trained and motivated to clarify and quantify critical objectives routinely.**

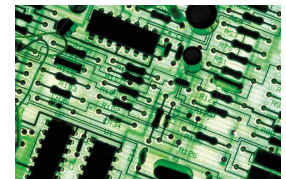
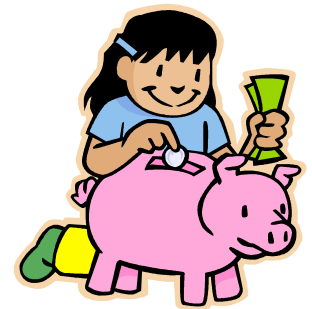
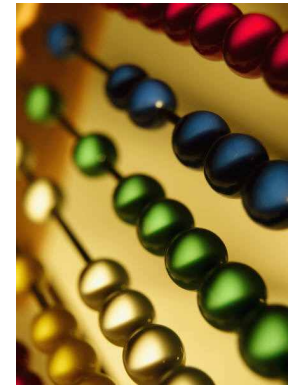
- " But until the poor external culture of education and practice changes, it may take strong CEO action to make this happen in your corporation.
- " My experience is that no one else will fight for this.

Assertion 4. **All top level system performance improvements, are by definition, variables.**

- " So, we can expect to define them quantitatively.
- " We can also expect to be able to measure or test the current level of performance.
- " Words like 'enhanced', 'reduced', 'improved' are not serious systems engineering requirements terms.

# Agile Methods

- have virtually no Quantified focus on the quality and performance levels of the software, PARTICULARLY the main reasons the project was funded (i.e. savings, more business, better service)
  - "This means you cannot control the main benefits
    - for users and stakeholders
  - "This means that you cannot really control the costs,
    - necessary in order to meet the quality needs of your users.
  - "It means the methods are not suitable for industrial products
    - where you cannot simply get the functionality, but must compete to deliver all sorts of qualities and performance attributes.



# So, what are Agile methods missing?

- " Stakeholder Focus
  - " Real projects have dozens of stakeholders
    - " Not just a customer in the next room
- " Results Focus
  - " It is not about writing code, it is about delivering value to stakeholders
  - " It is not about programming, it is about making systems work for real people
- " Systems Focus
  - " It is not about coding - again
  - " It is about reuse, data, hardware, training, motivation, sub-contracting, Outsourcing, help lines, user documentation, user interfaces, security
  - " So, a systems engineering scope is necessary to deliver results.
  - " Systems Engineering needs quantified performance and quality objectives
    - " To synchronize all necessary disciplines so that they deliver the results.



# So what extremes am I suggesting an eXtreme Programmer should go to?

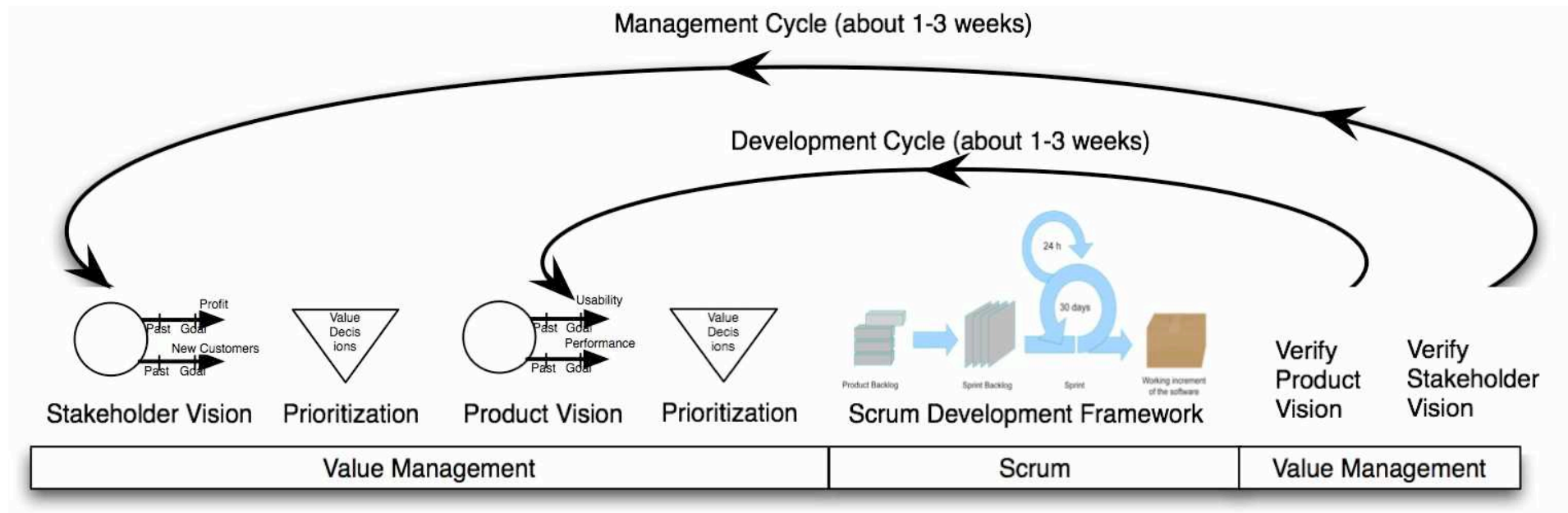
- " Learn to quantify quality objectives
- " Let your project be driven by the 'top ten' quality requirements -
  - " Everything else is 'design'
  - " *They* are why the project is funded
- " Learn to identify all critical stakeholders, and their requirements
- " Learn to deliver measurable results weekly
- " Prioritize delivery steps based on Value/cost, quantified
- " Brag about measurably improved products,
  - " Not simple methods



# Value Management (Evo) with Scrum development



- " developing a large web portal  
[www.bring.no/dk/se/nl/co.uk/com/ee](http://www.bring.no/dk/se/nl/co.uk/com/ee)  
at Posten Norge



# Value Management Process (Evo)



# Value Decision Tables

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

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

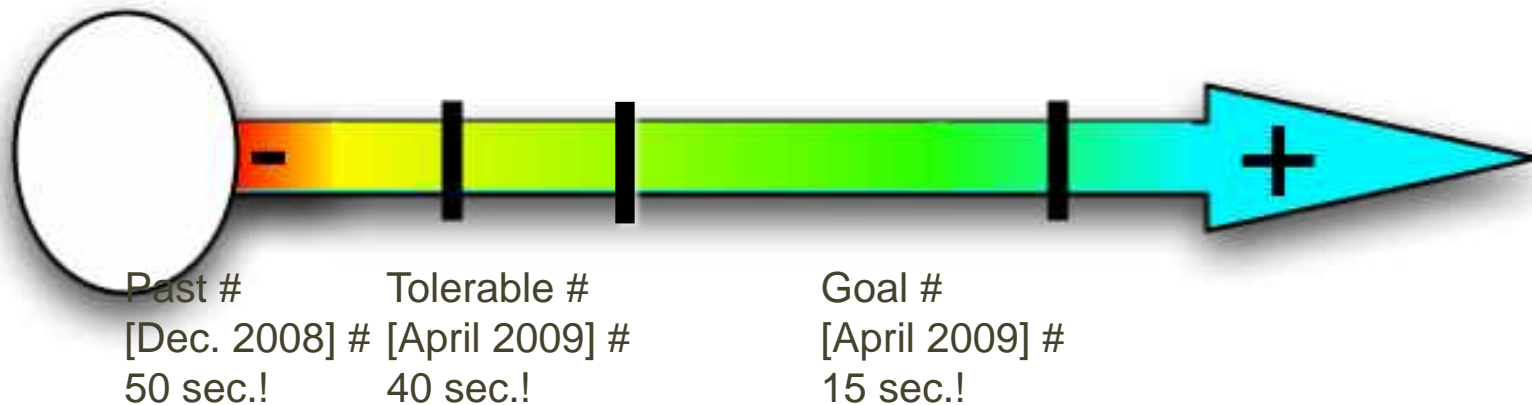
	Solution 1	Service Guide
Find.Fast	-10%	35 %
Product Value 2	50%	80 %
Resources	1 %	2 %

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

Scrum Develop  
We measure improvements  
Learn and Repeat

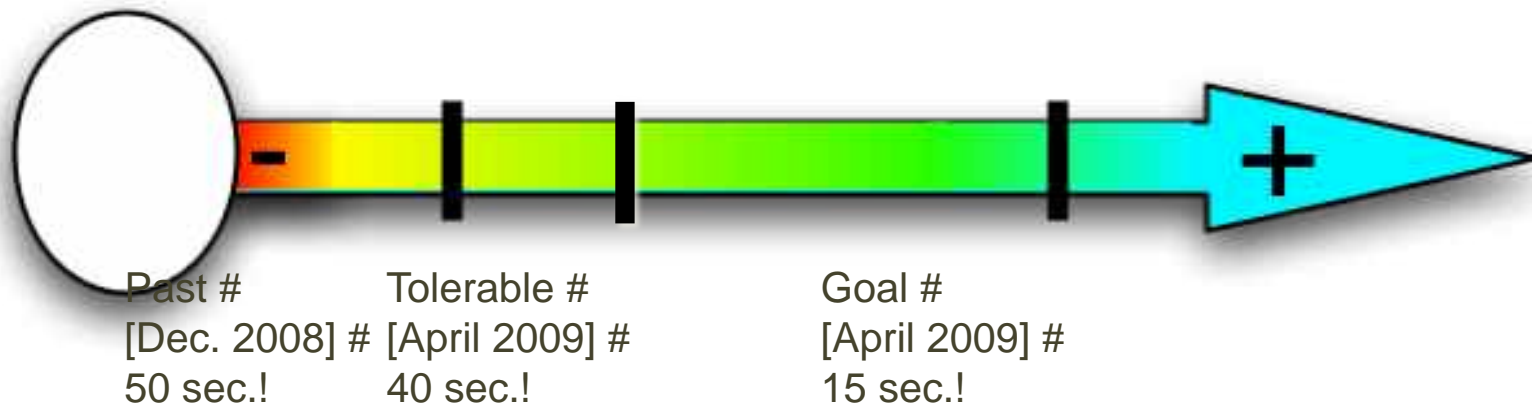
# Wargame

■ "The Developers (NetLife Research/Bekk) are challenged to **find several solutions** that can solve the challenge.



# Wargame

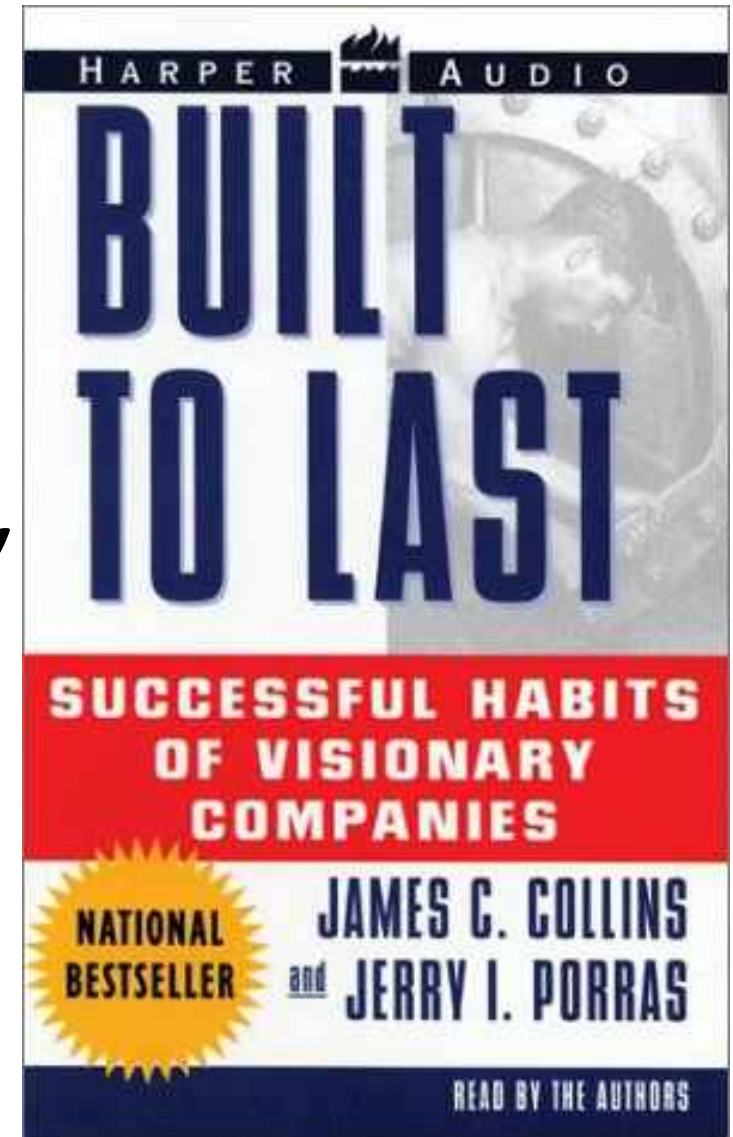
■ "And present those ideas to Management in a Value Decision Table with (gu)estimations about how much better things will become.





# Evolutionary Quotes

- *If well understood and consciously harnessed,*
  - *evolutionary processes can be a powerful way to stimulate progress.*
- *And that's exactly what the visionary companies have done*
  - *to a greater degree than comparison companies*
- "Jim Collins and Jerry Porras, "Built to Last"



# Evolve towards clear top goals

- *Instead of directing business according to detailed...strategic plan,*

- *"[Jack] Welch [General Electric CEO]*

- *believed in setting only a few clear, overarching goals.*

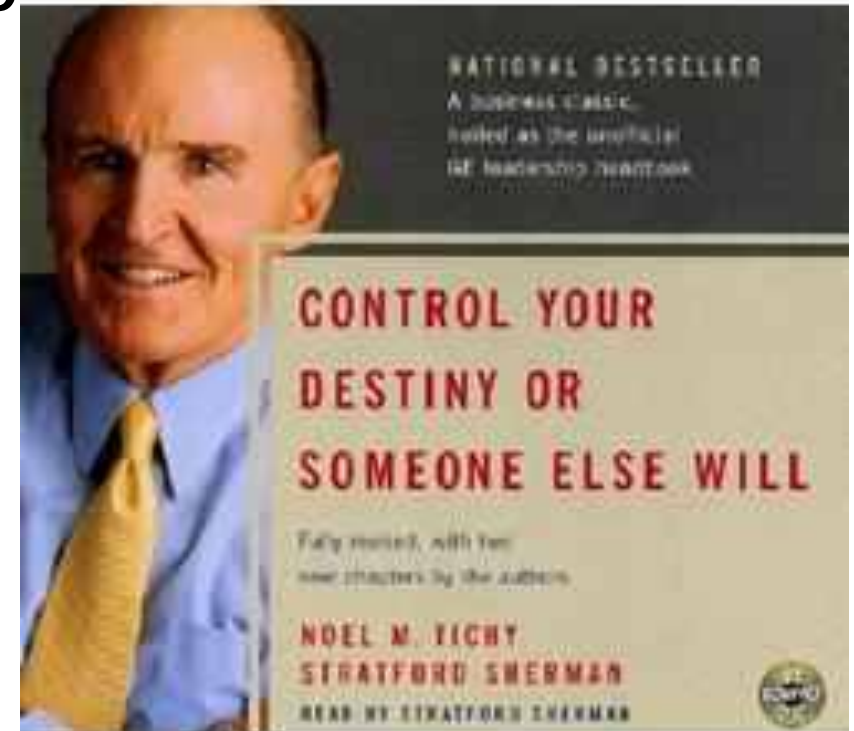
- *Then, on an ad hoc basis,*

- *"his people were free to seize any opportunities*

- *"they saw*

- *to further those goals. —*

- "Noel Tichy and Stratford Sherman,  
"Control Your Own Destiny or  
Someone Else Will"



# But first we need to think differently



We deliver value using time-boxed iterations and continue to fund projects only if they deliver measurable business results each release. Otherwise cancel the project (and preserve our resources for another project)!

## 2. The necessary *supplements* for successful IT

# The Simplest and Best Agile Project Method; 'XE'!

## •" Process Description

- " 1. Gather from all the key stakeholders the top few (5 to 20) most critical goals that the project needs to deliver.
  - " Give each goal a reference name (a tag).
- " 2. For each goal, define a scale of measure and a 'final' goal level.
  - " For example: *Reliable: Scale: Mean Time Before Failure, Goal: 1 month.*
3. Define approximately 4 budgets for your most limited resources
  - " (for example, time, people, money, and equipment).
4. Write up these plans for the goals and budgets
  - " (*Try to ensure this is kept to only **one page***).
- " 5. Negotiate with the key stakeholders to formally agree the goals and budgets.
- " 6. Plan to deliver some benefit
  - " (that is, progress towards the goals)
  - " in *weekly* (or shorter) increments (Evo steps).
- " 7. Implement the project in Evo steps.
  - " **Report** to project sponsors after each Evo step (weekly, or shorter) with your best available estimates or measures, for each performance goal and each resource budget.
  - " **On a single page**, summarize the *progress to date* towards **achieving** the goals and the costs incurred.
8. When all Goals are reached: 'Claim success and move on'
  - " a. Free remaining resources for more profitable ventures





# Agile project Management; XE Policy

## •" Policy

- " The project manager, and the project, will be **judged** exclusively on
  - " the relationship of progress towards achieving the goals
  - " versus the amounts of the budgets used.
  - " The project team will do anything legal and ethical to deliver the goal levels within the budgets.
- " The team will be paid and **rewarded** for
  - " benefits delivered
  - " in relation to cost.
- " The team will **find their own work process** and their own **design**.
- " As experience dictates, the team will be free to suggest to the project sponsors (stakeholders) adjustments to '**more realistic levels**' of the goals and budgets.



### 3. Examples of complimentary agile methods: Dominion Digital Case.

- " Ryan Shriver



- " rshriver@dominiondigital.com



# Summary

## “Give Value, not Code”

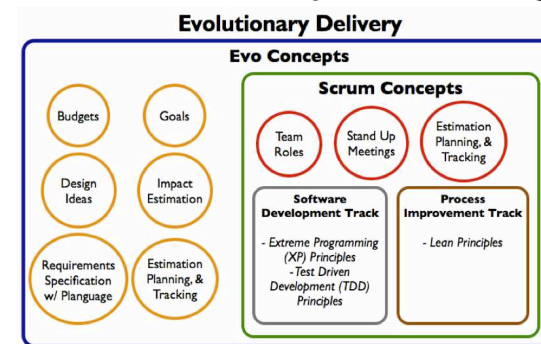


- "Conventional Agile methods (Scrum etc.) are fine for organising the programming tasks. !
- "But, they need to be supplemented by an Agile Envelope!



– "Evo' Method!

– "Which focuses on!

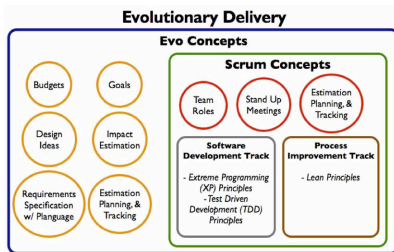


• "Delivery of useful results to stakeholders!

• "In both Norway  and USA  we have recent experience from this combination (Evo+Scrum)!

• "Are you ready for the next step of Agile Maturity?!



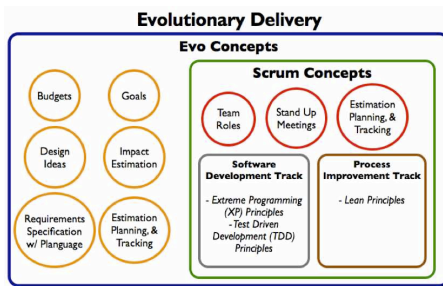


# Agile is an *improvement* but it's not *enough*



- " Yes they work –
  - " Agile methods (XP, Scrum) have proven themselves adept at delivering results quickly and agile is becoming more mature and accepted in the industry
- "But where's the **alignment with business value**?
  - Popular agile methods such as XP and Scrum **don't provide guidance**
    - "on ensuring the agile team is implementing solutions
    - " with the "biggest bang for the buck"
    - " and make sure that business is getting the best ***value for their money!***
- " **Alignment to Measurable Goals**
  - "In order for agile methods to transform, not only software projects, but also the way businesses **implement change** across their organization,
    - " teams using agile methods must align their work with **higher-level business goals** and
    - "measure their results, with **respect to helping organizations achieve their goals!**

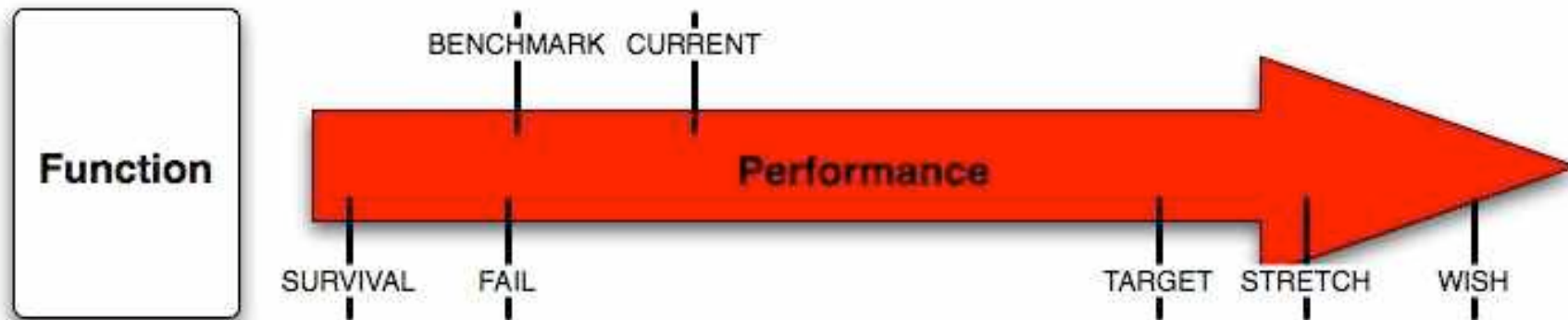
**"Just because you're Agile doesn't mean you're making Smart Decisions. Scrum and XP alone aren't enough!"**



## We need a **framework** to help us make Smart Decisions

- " **Measuring Progress towards Goals** - Defining measurable goals and recording before and after metrics to see if our solution really delivered value
- " **Judicious with our Budget** - With our resources and investments of time and money to ensure they're focused on the right projects. We're not funding projects that can't quantify how their solutions produce measurable progress towards the prioritized business goals (If you can't deliver results with 10% of the budget, what makes you think you can deliver results with 100%?)
- " **Analyze Frequent Feedback and Adapt** – Ensuring our investments are delivering measurable results using performance-to-cost ratios and percent-to-goals metrics. We're adapting to changing conditions on the ground using iterative planning and PDSA (Plan-Do-Study-Act)
- " **Utilizing People, Process and Technology** – Using the right balance of each to deliver well thought out solutions that maximize overall operational performance and don't simply "speed up the mess"
- " **Delivering value iteratively** - Utilizing popular agile methods (like Scrum and XP) to deliver the business value incrementally.

### 3 Requirements Examples DD Case: Specification with Planguage



#### **Decisioning Capability:**

**Ambition:** Develop the capability to rapidly build and deploy new decisioning rules !

**Scale:** Elapsed time in hours from idea to production upgrade of new decisioning rules that follow a pre-defined pattern!

**Goal [End Project] :** < 1 hour!

**Fail:** > 6 hours!

**Meter:** Wall clock time!

#### **Client Acquisition:**

**Ambition:** Acquire 2 new B2B clients and launch them on Release 2 of <Solution Name> !

**Scale:** New clients put into production with transactions flowing between parties!

**Goal [2008]:** 2!

**Fail [2008]:** 0!

**Meter:** Cognos report from analysis database!

#### **Update Capability:**

**Ambition:** Ability for a trained business analyst to update the offer decision rules directly !

**Scale:** Time in minutes for trained analyst to update offer rules and run test to validate change!

**Goal [End 2008] 5 minutes!**

**Fail:** > 15 mins!

**Meter:** Elapsed time as measured from user interface using wristwatch!



# Impact Estimation DD Case

## numeric evaluation of design



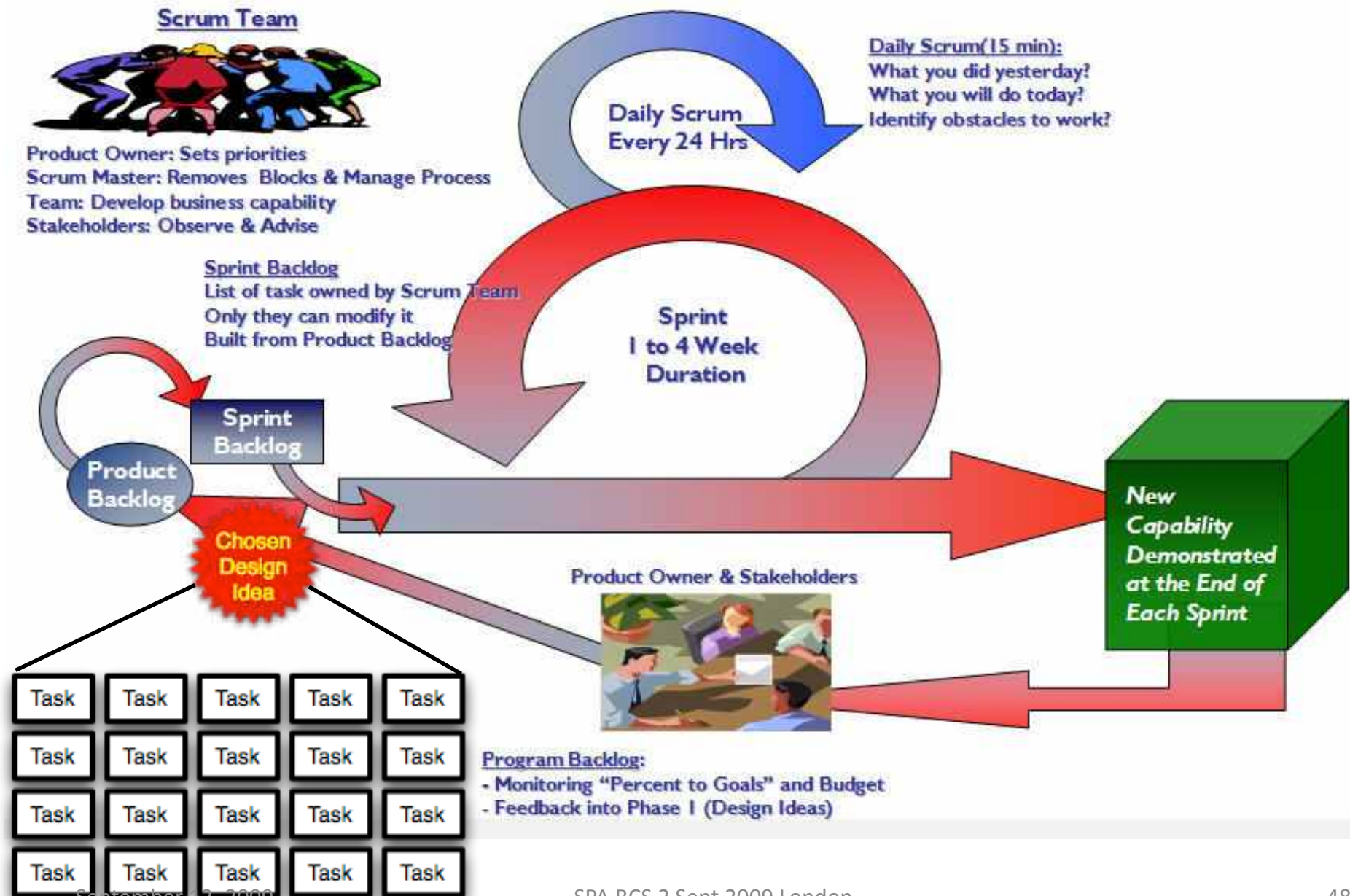
Goal: Increase Time to Sell (Individual hours devoted to direct sales activities) from 12 hrs/wk to 28 hrs/wk (30% to 70% of their time)		Design: Build New Accounts Wizard	Design: Electronically send data to SOR	Design	Totals
		Design Ideas			
Current Benchmark	12 hrs / wk	12	12	12	
Target Goal	28 hrs / wk	28	28	28	
Scale Impact	hrs / wk	1	2.5	0	3.5
Scale Uncertainty	+ / - hrs/wk	0.5	1	0	1.5
Percentage Impact	on design	6%	16%	0%	22%
Percentage Uncertainty	percentage	3%	6%	0%	9%
Evidence	based upon	Aneecdotal	High level estimate		
Source	person or doc	Ryan [06/18/07]	Ryan [06/20/07]		
Credibility	and 1	0.7	0.5		
Costs					
Solution Owner	effort hours	20	30	0	
Analysis	effort hours	70	200	0	270
Development	effort hours	100	300	0	400
Testing	effort hours	20	60	0	80
Total Resources	effort hours	210	590	0	800
Performance to Cost Ratio	of design	0.030	0.026	#VALUE!	
Credibility-adjusted					
Performance to Cost Ratio	factored in	0.021	0.013	#VALUE!	

September 12, 2009

SPA BCS 2 Sept 2009 London

47





**“Plan and Deliver” with Evolutionary Delivery**

Management Engineering

**Plan using Evo**

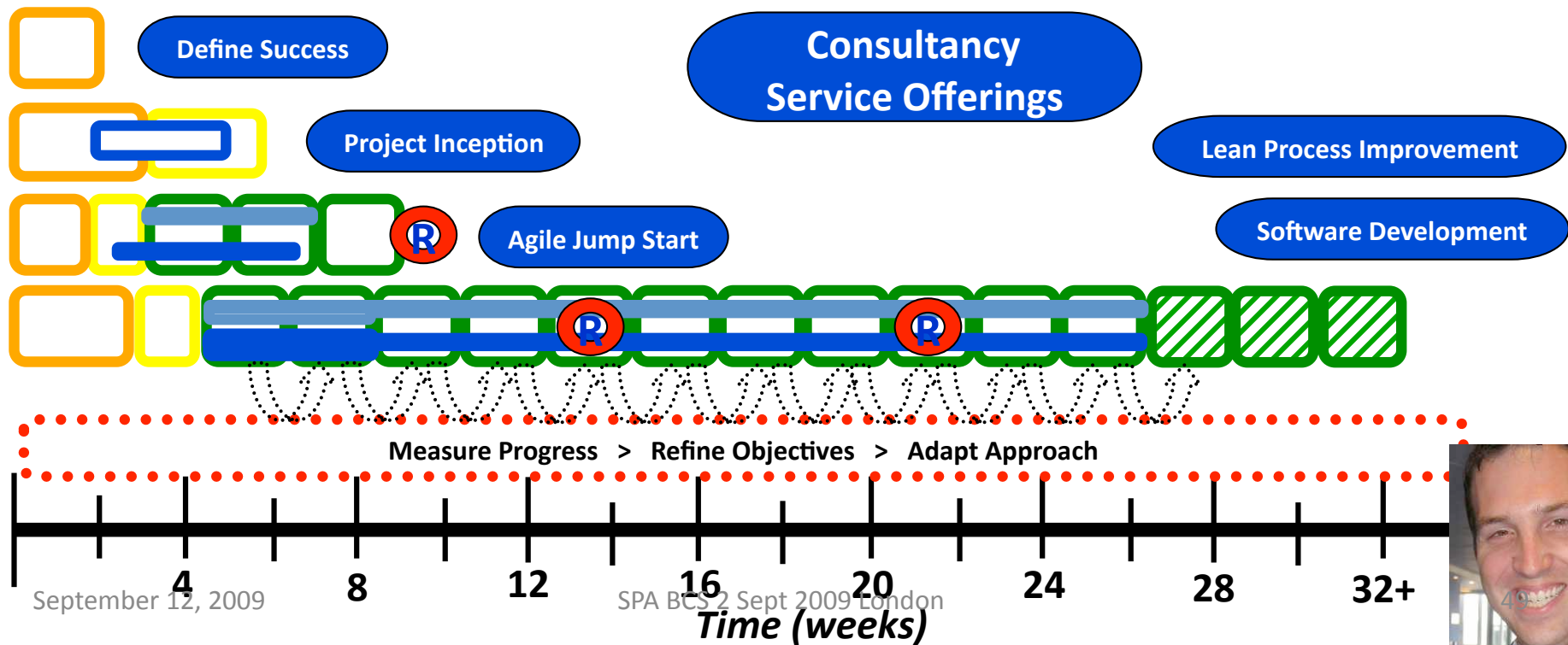
- Define Success
- Select Best Opportunity

**Deliver using Scrum**

- Requirements Engineering
- Design Engineering
- Test Engineering

Release

**COMPETITIVE ENGINEERING** by TOM GILBERT







## Evolutionary Delivery Components

### Define Success

Stakeholders

Values

Key Objectives

Resources

### Select Best Opportunity

Design Ideas

Design Criteria

Impact Estimation

Bang for the Buck

### Deliver Value

#### Requirements Engineering

Requirements Specification  
using Planguage

Inspection

Modeling

All Qualities are Quantified

#### Scrum

Sprints

Product Owner

ScrumMaster

Scrum Team

Release Planning

Sprint Planning

Stand Up Meetings

Retrospectives

Product Backlog

Sprint Backlog

#### Design Engineering

XP Principles  
& Practices

Systems Architecture

Test Driven Development

Lean  
Principles  
& Practices

Management Engineering

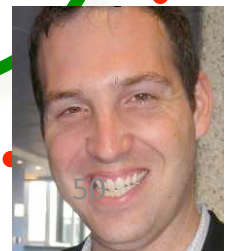
Evolve Goals

Plan, Estimate, Track to Goals

Delivery Quality Measurement

Test Engineering

Automation



## 4. The ConFIRMit Case study of a successful agile method.

# FIRM as Presentation

## Trond Johnsen

Tom Gilb Version

May 7 2005

Updated with 9.0 2006

Updated 19 June 06 \$\$ Results

Updated Sept 2006 Estimation, OCT INTUITIVENESS

Graphical Improvement Nov 8 2006, Oct 10 2007



September 12, 2009

2 Sept 2009 London

52

# Customer Successes in Corporate Sector

# FIRM R&D department

- " Developers (13)
- " Management/(CSO) (2)
- " Tech Support NY (1)
- " Microsoft .NET framework, SQL
- " SEPG group (3) with responsibility of process improvement and quality assurance (QA).
  - "Configuration Management, setup ++
  - "Testing
  - "Software Process Improvement (SPI)



# Requirements - 3, Real Example of Spec

## Usability.Productivity *(taken from Confirmit 8.5 development)*

Scale for quantification: Time in minutes to set up a typical specified Market Research-report

Past Level [Release 8.0]: 65 mins.,

Tolerable Limit [Release 8.5]: 35 mins.,

Goal [Release 8.5]: 25 mins.

Note: end result was actually 20 minutes ☺

Meter [Weekly Step]: Candidates with Reportal experience, and with knowledge of MR-specific reporting features, performed a set of predefined steps, to produce a standard MR Report.



Trond Johansen

- " Our new focus is on the **day-to-day operations** of our Market Research users,
  - " not a list of features that they might or might not like. 50% never used!
  - " We KNOW that increased efficiency, which leads to more profit, will please them.
  - " The '45 minutes actually saved x thousands of customer reports'
    - " = big \$\$\$ saved
- " After **one week** we had defined more or less all the requirements for the next version (8.5) of Confirmit.



September 12, 2009



SPA BCS 2 Sept 2009 London

55



Trond Johansen

- IET for MR Project – Confirmit (<-FIRM Product Brand) 8.5
- **Solution:** Recoding
  - Make it possible to recode variable on the fly from Reportal.
  - Estimated effort: 4 days
  - **Estimated** Productivity Improvement: 20 minutes (50% way to Goal)
  - actual result 38 minutes (95% progress towards Goal)

	A	B	C	D	E	F	G	BX	BY	BZ	CA
1											
2		Current Status	Improvements		Goals			Step9			
3								Recoding			
4								Estimated impact		Actual impact	
5		Units	Units	%	Past	Tolerable	Goal	Units	%	Units	%
6					Usability.Replacability (feature count)						
7		1,00	1,0	50,0	2	1	0				
8					Usability.Speed.NewFeaturesImpact (%)						
9		5,00	5,0	100,0	0	15	5				
10		10,00	10,0	66,7	0	15	5				
11		0,00	0,0	0,0	0	30	10				
12					Usability.Intuitiveness (%)						
13		0,00	0,0	0,0	0	60	80				
14					Usability.Productivity (minutes)						
15		20,00	45,0	112,5	65	35	25	20,00	50,00	38,00	95,00
20					Development resources						
21			101,0	91,8	0		110	4,00	3,64	4,00	3,64



Real client example: weekly design impact estimates, and same week measurement,  
 Weekly Feedback to the development team  
 about cumulative progress toward critical numeric performance and quality targets

	A	B	C	D	E	F	G	BX	BY	BZ	CA
1											
2		Current Status	Improvements		Goals			Step9			
3								Recoding			
4								Planned impact		Actual impact	
5		Units	Units	%	Past	Tolerable	Goal	Units	%	Units	%
6					Usability.Replacability (feature count)						
7		1,00	1,0	50,0	2	1	0				
8					Usability.Speed.NewFeaturesImpact (%)						
9		5,00	5,0	100,0	0	15	5				
10		10,00	10,0	200,0	0	15	5				
11		0,00	0,0	0,0	0	30	10				
12					Usability.Intuitiveness (%)						
13		0,00	0,0	0,0	0	60	80				
14					Usability.Productivity (minutes)						
15		20,00	45,0	112,5	65	35	25	20,00	50,00	38,00	95,00
20					Development resources						
21			101,0	91,8	0		110	4,00	3,64	4,00	3,64

Priority  
 Next week  
 Warning  
 metrics based

Cumulative  
 weekly  
 progress  
 metric

Development resources

Development resources

# Evo – IET


- " Product quality:
  - " Usability.**Intuitiveness**: Probability that <secret name of stakeholders> can intuitively, and without any help, figure out how to do a set of defined, common, simple tasks correctly (without any errors needing correction)

Current Status		Improvem ent	Goals			Step 1 (7.-18.Aug)		Step 2 (21.-1.sep)		Step 3 (4.-15.sep)	
	Units		Past	Tolerable	Goal	Estimated Impact	Actual Impact	Estimated Impact	Actual Impact	Estimated Impact	Actual Impact
Usability.Intuitiveness											
	9,0	9,0	18	12	8					8	9
	1,5	5,0	6,5	3,0	1,0					4,5	5,0

- " Meter1: The time it takes for “secret name of stakeholders” (First time users) to create a SimpleSet1 of pre-defined authoring tasks
- " Meter2: The number of times “secret name of stakeholders” (First time users) are uncertain of how to perform a step in SimpleSet1



## FIRM EVO-week cycle

	Development Team	Users (PMT, Pros, Doc writer, other)	CTO (Sys Arch, Process Mgr)	QA (Configuration Manager & Test Manager)
<b>Friday</b>	<ul style="list-style-type: none"> <li>✓ PM: Send Version N detail plan to CTO + prior to Project Mgmt meeting</li> <li>✓ PM: Attend Project Mgmt meeting: 12.00-15.00</li> <li>✓ Developers: Focus on genereal maintenance work, documentation.</li> </ul>		<ul style="list-style-type: none"> <li>✓ Approve/reject design &amp; Step N</li> <li>✓ Attend Project Mgmt meeting: 12-15</li> </ul>	<ul style="list-style-type: none"> <li>✓ Run final build and create setup for Version N-1.</li> <li>✓ Install setup on test servers (external and internal)</li> <li>✓ Perform initial crash test and then release Version N-1</li> </ul>
<b>Monday</b>	<ul style="list-style-type: none"> <li>✓ Develop test code &amp; code for Version N</li> </ul>	<ul style="list-style-type: none"> <li>✓ Use Version N-1</li> </ul>		<ul style="list-style-type: none"> <li>✓ Follow up CI</li> <li>✓ Review test plans, tests</li> </ul>
<b>Tuesday</b>	<ul style="list-style-type: none"> <li>✓ Develop Test Code &amp; Code for Version N</li> <li>✓ Meet with users to Discuss Action Taken Regarding Feedback From Version N-1</li> </ul>	<ul style="list-style-type: none"> <li>✓ Meet with developere rs to give Feedback and Discuss Action Taken from previous actions</li> </ul>	<ul style="list-style-type: none"> <li>✓ System Architect to review code and test code</li> </ul>	<ul style="list-style-type: none"> <li>✓ Follow up CI</li> <li>✓ Review test plans, tests</li> </ul>
<b>Wednesday</b>	<ul style="list-style-type: none"> <li>✓ Develop test code &amp; code for Version N</li> </ul>			<ul style="list-style-type: none"> <li>✓ Review test plans, tests</li> <li>✓ Follow up CI</li> </ul>
<b>Thursday</b>	<ul style="list-style-type: none"> <li>✓ Complete Test Code &amp; Code for Version N</li> <li>✓ Complete GUI tests for Version N-2</li> </ul>			<ul style="list-style-type: none"> <li>✓ Review test plans, tests</li> <li>✓ Follow up CI</li> </ul>



# Code quality – “green” week

- In these “green” weeks, some of the deliverables will be less visible for the end users, but more visible for our QA department.
- We manage code quality through an Impact Estimation table.

Current Status		Improvement	Goals			Step 6 (week 14)		Step 7
	Units		Past	Tolerable	Goal	Estimated Impact	Actual Impact	Estimated Imp
	100,0	100,0	0	80	100			
Speed								
	100,0	100,0	0	80	100	100	100	
Maintainability.Doc.Code								
	100,0	100,0	0	80	100	100	100	
InterviewerConsole								
	0,0	0,0	0	90	100			
NUnitTests								
	100,0	100,0	0	90	100			
PeerTests								
	0,0	10,0	10	0	0			
FxCop								
	100,0	100,0	0	90	100			
TestDirectorTests								
	2,0	2,0	0	1	2	2	2	
Robustness.Correctness								
	0,0	0,0	0	80	100			
Robustness.BoundaryConditions								
	0,0	0,0	0					
Speed								
	100,0	0,0	100					
ResourceUsage.CPU								
	100,0	100,0	0					
Maintainability.Doc.Code								
SynchronizationStatus								
NUnitTests								

POT-SHOTS — Brilliant Thoughts in 17 words or less



© Ashleigh Brilliant 1980

www.ashleighbrilliant.com

Speed

Maintainability

Nunit Tests

PeerTests

TestDirectorTests

Robustness.Correctness

Robustness.Boundary  
Conditions

ResourceUsage.CPU

Maintainability.DocCode

SynchronizationStatus

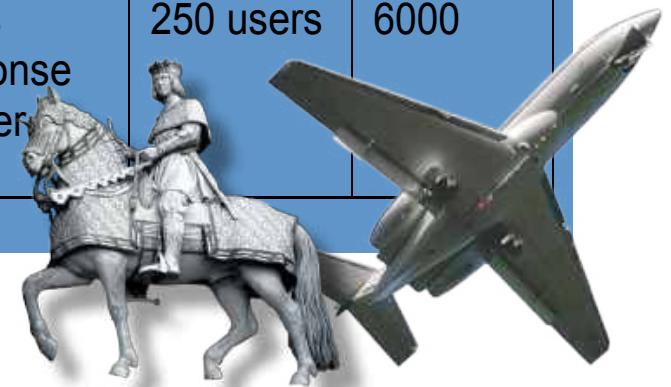
September 12, 2009

61

## EVO's impact on Confirmit product qualities - 2

- " Only highlights of the impacts are listed here

Description of requirement/work task	Past	Status
Usability.Productivity: Time for the system to generate a survey	7200 sec	15 sec
Usability.Productivity: Time to set up a typical specified Market Research-report (MR)	65 min	20 min
Usability.Productivity: Time to grant a set of End-users access to a Report set and distribute report login info.	80 min	5 min
Usability.Intuitiveness: The time in minutes it takes a medium experienced programmer to define a complete and correct data transfer definition with Confirmit Web Services without any user documentation or any other aid	15 min	5 min
Performance.Runtime.Concurrency: Maximum number of simultaneous respondents executing a survey with a click rate of 20 sec and an response time<500 ms, given a defined [Survey-Complexity] and a defined [Server Configuration, Typical]	250 users	6000





# Initial Experiences and conclusions

- " We launched our first major release based on Evo in May 2004 (Rel. 8.5)
  - " and we have already gotten feedback from users on some of the leaps in product qualities.
  - " E.g. the time for the system to generate a complex survey has gone **from 2 hours (=wait for the system to do work) to 15 seconds!**
- " EVO has resulted in
  - " increased **motivation and**
  - " **enthusiasm** amongst developers,
  - " it opens up for empowered creativity
- " Developers
  - " **embraced the method** and
  - " **saw the value of using it,**
  - " even though they found parts of Evo difficult to understand and execute
- " Project leaders feel:
  - " Defining good requirements can be hard.
  - " It was hard to find meters which were practical to use, and at the same time measure real product qualities.
  - " Sometimes we would like to spend more than a day on designs, but this was not right according to our understanding of Evo. (Concept of backroom activity was new to us)
  - " Sometimes it takes more than a week to deliver something of value to the client. (Concept of backroom activity was new to us)



# Experiences and conclusions – 2

## – "Team members (developers)

- "“Sometimes it felt like we were rushing to the next weekly step, before we had finished the current step”
- "Testing was sometimes ‘postponed’
  - "in order to start next step,
  - "some of these test delays were not compensated for, in later testing.



Trond Johansen



## Evo's impact on **confirmit** product qualities - 1

- " The impact described is based on:
  - " Internal usability test, productivity tests ++
  - " Performance tests carried out at Microsoft Windows ISV laboratory in Redmond USA
  - " Direct customer feedback
    - " *"I just wanted to let you know how appreciative we are of the new "entire report" export functionality you recently incorporated into the Reportal.*
    - " *It produces a fantastic looking report, and the table of contents is a wonderful feature.*
    - " *It is also a **HUGE time saver.**" <- Customer*
  - " "These leaps in product qualities would not have been achieved without Evo". <- TJ





Trond Johansen

## Conclusions - 1

- " The method's **positive impact** on Confrimit product qualities has convinced us that
  - " Evo is a better suited development process than our former waterfall process, and
  - " we will continue to use Evo in the future.
- " What **surprised** us the most was
  - " the method's **power of focusing on delivering value** for clients versus cost of implementation.
  - " Evo enables you to **re-prioritize** the next development-steps based on the weekly feedback.
  - " What seemed important
    - " at the start of the project
    - " may be replaced by other solutions
    - " based on knowledge gained from previous steps.
- " The method has
  - " high focus on **measurable product qualities**, and
    - " defining these clearly and testably, requires training and maturity.
  - " It is important to **believe that everything can be measured**,
    - " and to seek guidance if it seems impossible.





Trond Johansen

## Conclusions - 2

- " A pre-requisite related to the method for using Evo is an ***open architecture***.
- " Another pre-requisite is ***management support*** for changing the work process, and this is important in any software process improvement initiative.
- " The concept of **Continuous Integration (CI)/daily builds**
  - " was valuable
  - " with respect to delivering new versions of the software every week.
- " Evo,
  - " as most other software processes,
  - " requires continuous focus
  - " and learning about the methodology.





Trond Johansen

# The way ahead

- " Overall, the whole organization has embraced EVO.
- " We all think it has great potential,
  - " and we will work hard to utilize it to the full.
- " In June 2004
  - " we had Tom and Kai Gilb for a 4 days course for the whole R&D department and related resources
- " The next version of Conformat, Conformat 9.0, will prove whether we have matured in our understanding and execution of EVO
- " Conformat 9.0 is due to be released Q4 2004, here is a sneak preview...





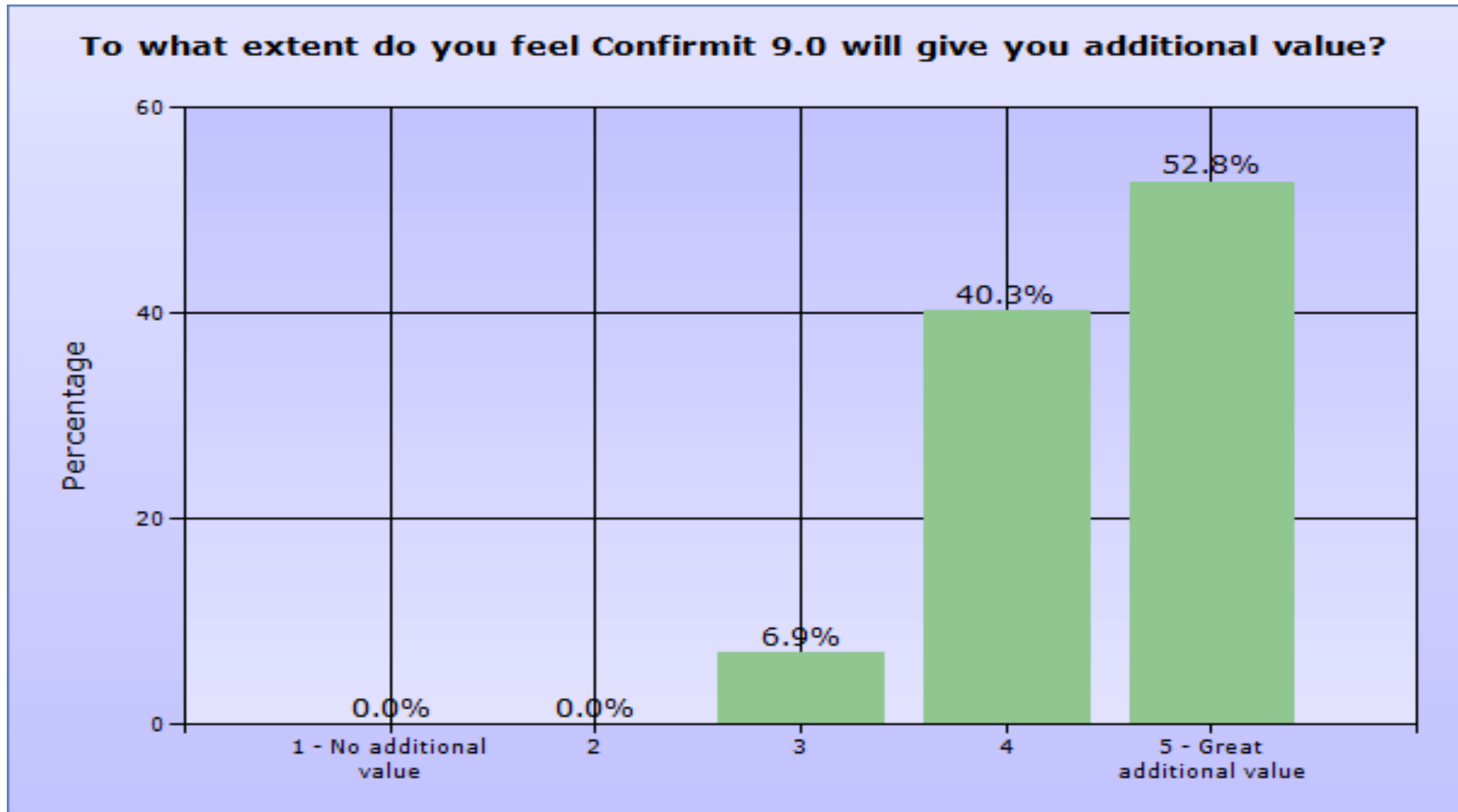
- Theme for 9.0:
  - Extend usage in large corporations,
    - hence focus on **usability, intuitiveness, easy to learn**
  - Market Research:
    - Support for large panels, up to 200 000 panellists.
    - Improve productivity in general for those who work with such large panels
  - Improve throughput
    - for users that receive reports with more than 1 000 000 responses
      - (important for large corporations; HP, Microsoft, Accenture etc)

Description of requirement/work task	Past	Status 11.09	Goal
<b>Usability.Intuitiveness:</b> Probability that a defined User can intuitively figure out how to do a defined Task correctly (without any errors needing correction)	30%	45%	80%
<b>Panel.Scalability:</b> Maximum number of panelists that the system can support within a timeframe of 120 seconds for creating a sample of 50 000, with all components of the panel system performing acceptably.	30,000	500,000	200,000
<b>Performance.DataVolume:</b> Numbers of survey responses that can be handled by Reportal. Tables should be generated within 5 seconds.	20,000	500,000	500,000

# Initial Customer Feedback on the new Conformat 9.0

November 24th, 2004

# Initial perceived value of the new release (Base 73 people)





# ACTUAL RESULTS IN SECOND 12 WEEKS OF USING THE NEW METHOD

Evo's impact on Conformat 9.0 product qualities

Product quality	Description	Customer value
Intuitiveness	Probability that an inexperienced user can intuitively figure out how to set up a defined Simple Survey correctly.	Probability increased by 175%
Productivity	Time in minutes for a defined advanced user, with full knowledge of 9.0 functionality, to set up a defined advanced survey correctly.	Time reduced by <b>38%</b>

Product quality	Description	Customer value
Productivity	Time (in minutes) to test a defined survey and identify 4 inserted script errors, starting from when the questionnaire is finished to the time testing is complete and is ready for production. (Defined Survey: Complex survey, 60 questions, comprehensive JScripting.)	Time reduced by <b>83%</b> and error tracking increased by 25%

# MORE ACTUAL RESULTS IN SECOND 12 WEEKS OF USING THE NEW METHOD

Evo's impact on Conformat 9.0 product qualities

Product quality	Description	Customer value
Performance	Max number of panelists that the system can support without exceeding a defined time for the defined task, with all components of the panel system performing acceptable.	Number of panelists increased by <b>1500%</b>
Scalability	Ability to accomplish a bulk-update of X panelists within a timeframe of Z sec.	Number of panelists increased by <b>700%</b>
Performance	Number of responses a database can contain if the generation of a defined table should be run in 5 seconds.	Number of responses increased by <b>1400%</b>

# Initial qualitative feedback on the new release

" ... keep up the good work."

"It looks like you have listened to the people that actually use the software daily and aimed to make it easier for them ... "

"I was very impressed with the version 9.0"

- " Seminar observations !

- "On several occasions, customers gave spontaneous "WOWs" and applause!
- "The training room in London was literally packed with people eager to test the new version.
- "Several clients asked if they could access the test server from home as well.
- "Great participation rate; 95% of all registered people showed up. !



## Press Release from FIRM

### New version of Confirmat increases user productivity up to 80 percent

NOVEMBER 29th, 2004

- " : FIRM, the world's leading provider of online survey & reporting software, today announced the release of a new version of Confirmat delivering substantial value to customers including increased user productivity of up to 83 percent.
- " FIRM is using Evolutionary (EVO) development to ensure the highest focus on customer value through early and continuous feedback from stakeholders.
- " A key component in EVO is measuring the effect new and improved product qualities have on customer value.
- " Increased customer value in Confirmat 9.0 includes: -
  - " Up to 175 percent more intuitive user interface\*.
  - " Up to 80 percent increased user productivity in questionnaire design and testing\*.
  - " Up to 1500 percent increased performance in Reportal and Panel Management\*.
- " Features delivering increased customer value include: -
  - " A completely new and state-of-the-art user interface.
  - " Random Data Generator enabling automated testing of questionnaires.
  - " Real-time Script Checker for on-the-fly script validation.
  - " Block Randomization of questions to avoid respondent bias.
  - " Reportal BitStream for fast online tabulation on high volume of responses.
  - " We are very pleased to see major improvements in Confirmat 9.0, including updates to both the user interface and survey engine. We plan to deploy this new version when it becomes available to server customers, stated Alex Grinberg, Greenfield Online's Chief Information Officer. -
    - " We believe the improvements in Confirmat 9.0 will benefit Greenfield Online's survey programming, data collection and data delivery capabilities, helping us to bring even more value to our clients.
  - " FIRM's VP of Marketing, Kjell Øksendal, comments; - FIRM, through evolutionary development, is able to substantially increase customer value by focusing on key product qualities important for clients and by continuously asking for their feedback throughout the development period.
  - " Confirmat is used by the leading market research agencies worldwide and Global 1000 companies, and together, we have defined the future of online surveying and reporting, represented with the Confirmat 9.0. Confirmat 9.0 was released onto FIRM's ASP environments in London and New York on November 27th. The new version will be available for server customers in January 2005. \* Measured in FIRM's TestLab by monitoring internal and external stakeholders executing predefined test scenarios.

# Initial qualitative feedback on the new release

" ... keep up the good work."

"It looks like you have listened to the people  
that actually use the software daily and  
aimed to make it easier for them ... "

"I was very impressed with the version 9.0"

- " Seminar observations !

- " On several occasions, customers gave spontaneous "WOWs" and  
applauses!
- " The training room in London was literally packed with people eager  
to test the new version.
- " Several clients asked if they could access the test server from  
home as well.
- " Great participation rate; 95% of all registered people showed up. !

# FIRM Results Since Evo Method

## ∴ Revenue growth



### Quarterly revenues - Y/Y growth



- ✓ Full year 2005 revenue growth: 33%
- ✓ YTD Q2 2005 revenue growth: 27%
- ✓ YTD Q2 2006 revenue growth: 27%



Has FIRM found their estimation values have become more accurate over time and experience

- " ***Has FIRM found their estimation values have become more accurate over time and experience?***
- " Yes, the estimation gets better. However, after working with Evo for some time now, we are not estimating each step as detailed as before, and we run fewer formal measurements.
- " Evo has gotten into our veins,
- " we feel more confident and hence can spend less of our time with formal project management tasks and still get the huge benefits from Evo.
- " We find that we normally can quite easily tell which solutions will bring us closest to the goal after practicing Evo for a while.
- "
- " ***"how closely have those estimations actually been realised" and "how valuable has estimation value accuracy been".***
- " The importance of estimation accuracy will depend a bit on your business (product/project related work), but we are now in a position where usually don't significantly miss with our estimates. This makes it easier to **roughly** plan at least 2-3 steps ahead (i.e. we can set up a monthly plan containing the heading of each step and some bullet points on each solution).
- " As new to Evo, I would suggest that you try to follow Evo as "strictly" as you can.
  - " This will give you the experience and confidence to later pay more attention to some parts of Evo, and less on other parts, while keeping the overall benefits.
- " I would say it's harder for us now to create killer-slides containing measurements than it was the first couple of releases, but the value for the clients (which is more important..) is still the same or better. (This is based on customer satisfaction measurements amongst our clients, using our own software and our gut-feeling.)
- " That's all I have time for, it's very busy these days. Best  
trond
  - " **Trond Johansen**  
Software Development Manager  
Trond.Johansen@confirmit.com | Phone +47 21 50 25 25 | Mobile +47 92234861 **Confirmit Deliver Actionable Insight Now,**  
**Future Information Research Management (FIRM)**  
Hoffsveien 48, NO-0377 Oslo, Norway  
[www.confirmit.com](http://www.confirmit.com) | Main +47 21 50 25 00 | Fax +47 21 50 25 01



5. Stakeholder analysis:  
the many existing  
requirements holders.  
Not limited to 'customer'  
or 'user'.

# Value Driven Planning: 10 Value Principles

# Value Driven Planning: Stakeholders, Value Focus, Quantified, Stepwise

- " Value Driven Planning focuses on
  - " the primary values of key *stakeholders*.
- " The *technology* used, and the project *processes* used are sub-ordinate.
- " The critical stakeholder values are quantified and trackable.
- " There is an assumption of
  - " step by step achievement,
  - " of *learning* at each step
  - " and consequent *action*
    - " to resolve problems of value achievement.



## **Gilb's 'Value Driven Planning' Principles:**

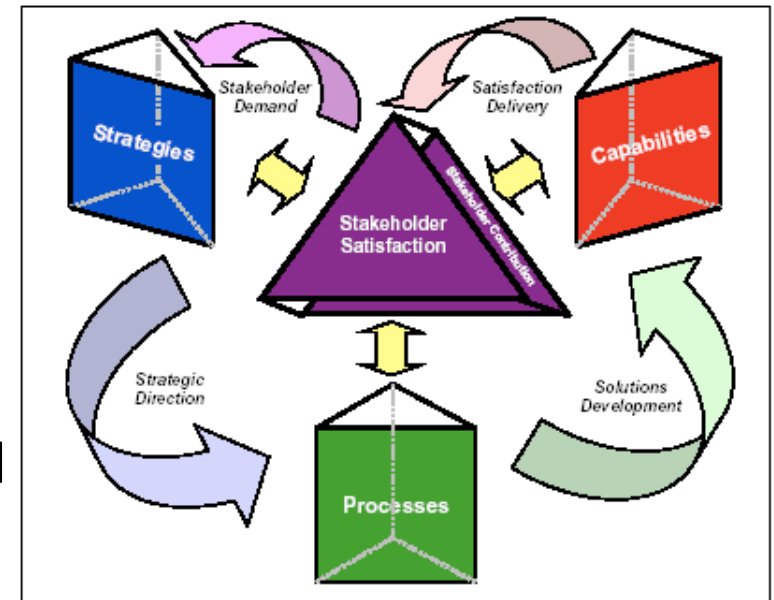
- 1. Critical Stakeholders determine the values**
- 2. Values can and must be quantified**
- 3. Values are supported by Value Architecture**
- 4. Value levels are determined by timing, architecture effect, and resources**
- 5. Value levels can differ for different scopes (where, who)**
- 6. Value can be delivered early**
- 7. Value can be locked in incrementally**
- 8. New Values can be discovered (external news, experience)**
- 9. Values can be evaluated as a function of architecture (Impact Estimation)**
- 10. Value delivery will attract resources.**

# Value Driven Planning Principles in Detail:

# 1. Critical Stakeholders determine the values

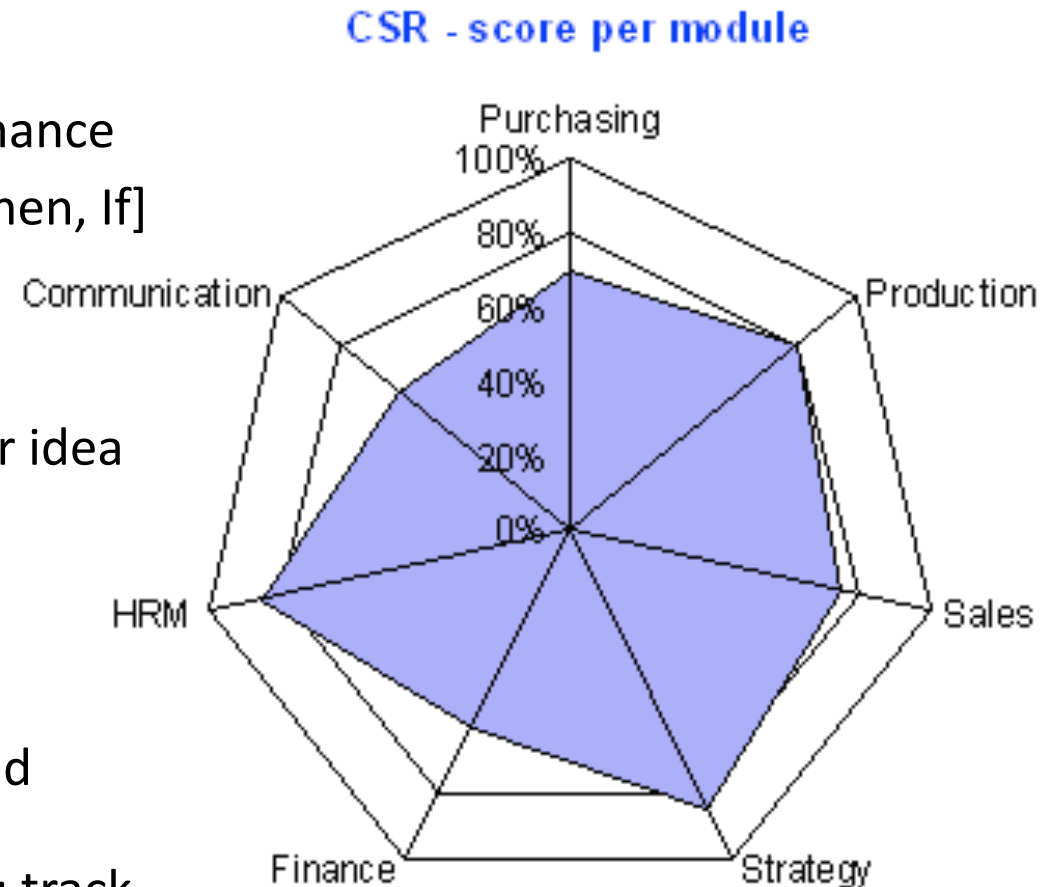
Critical: “having a decisive or crucial importance in the success or failure of something” <-Dictionary

- " The primary and prioritized values we need to deliver are determined by
  - " analysis of the needs and values of stakeholders
    - " stakeholders who can determine whether we *succeed or fail*.
- " We cannot afford to satisfy *other (less critical)* levels, at other times and places, yet.
  - " Because that might undermine our ability to satisfy the more critical stakeholders –
  - " and consequently threaten our overall project success.



## 2. 'Values' can and must be *quantified*

- " Values can, if you want, be expressed numerically.
  - " With a defined scale of measure
  - " with a deliverable level of performance
  - " and with qualifier info [Where, When, If]
- " Quantification is useful:
  - " to clarify your own thoughts
  - " to get real agreement to one clear idea
  - " to allow for varied targets and constraints
  - " to allow direct comparison with benchmarks
  - " to put in Request for bids, bids and contracts
  - " to manage project evolutionarily : track progress
  - " as a basis for measurement and testing
  - " to enable research on methods





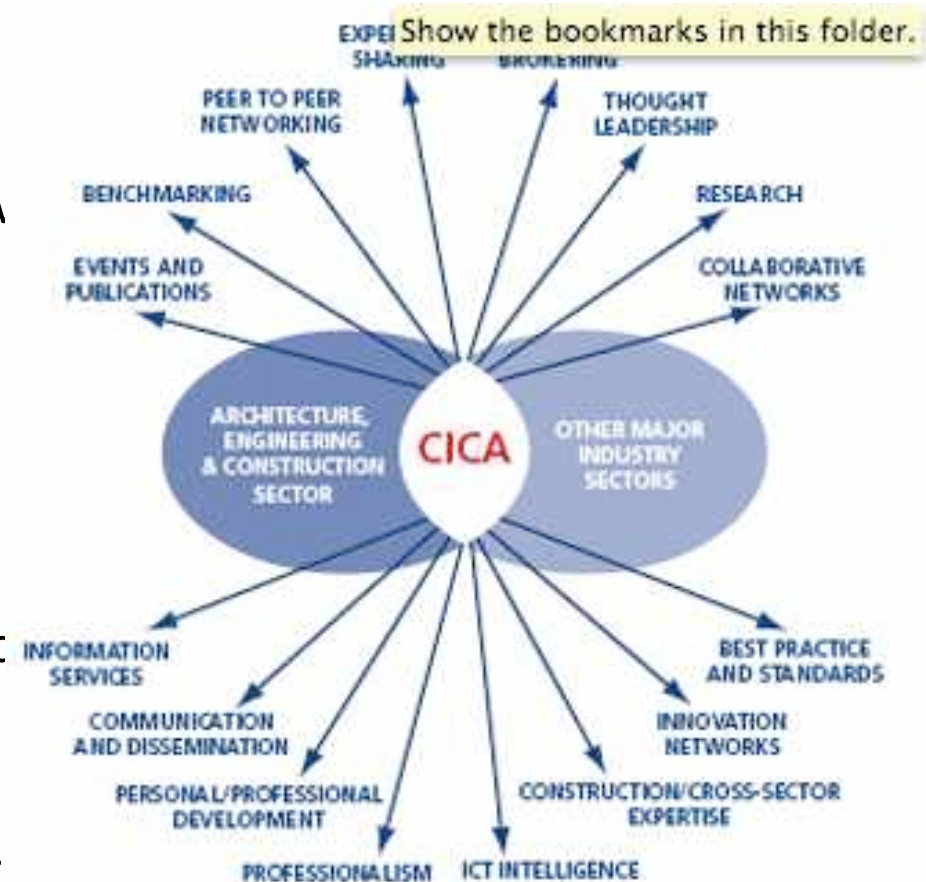
- Figure 1: Real (NON-CONFIDENTIAL version) example of an initial draft of setting the objectives that engineering processes must meet.

Business objective	Measure	Goal (200X)	Stretch goal ('0X)	Volume	Value	Profit	Cash
Time to market	Normal project time from GT to GT5	<9 mo.	<6 mo.	X	X	X	X
Mid-range	Min BoM for The Corp phone	<\$90	<\$30	X	X	X	X
Platformisation Technology	# of Technology 66 Lic. shipping > 3M/yr	4	6	X	X	X	X
Interface	Interface units	>11M	>13M	X	X	X	X
Operator preference	Top-3 operators issue RFQ spec The Corp	1	2	X	X	X	X
Productivity				X	X	X	X
Get Torden	Lyn goes for Technology 66 in Sep-04	yes		X	X	X	X
Fragmentation	Share of components modified	<10%	<5%	X	X	X	X
Commoditisation	Switching cost for a UI to another System	>1yr	>2yrs	X	X	X	X
Duplication	The Corp share of 'in scope' code in best-selling device	>90%	>95%	X	X	X	X
Competitiveness	Major feature comparison with MX	Same	Better	X	X	X	X
User experience	Key use cases superior vs. competition	5	10	X	X	X	X
Downstream cost saving	Project ROI for Licensees	>33%	>66%	X	X	X	X
Platformisation IFace	Number of shipping Lic.	33	55	X	X	X	X
Japan	Share of of XXX sales	>50%	>60%	X	X	X	X
Numbers are intentionally changed from real ones							

Business Values Quantified

### 3. Values are supported by Value Architecture

- " Value Architecture: defined as:
  - " anything you *implement* with a view to satisfying stakeholder values.
- " Value Architecture:
  - " includes product/system objectives
    - " Which are a 'design' for satisfying stakeholder values
  - " Has a multitude of performance and cost impacts
  - " can impact a given system differently, depending on what is in the system, or what gets put in later
  - " Needs to try to maximize value delivered for resources used.

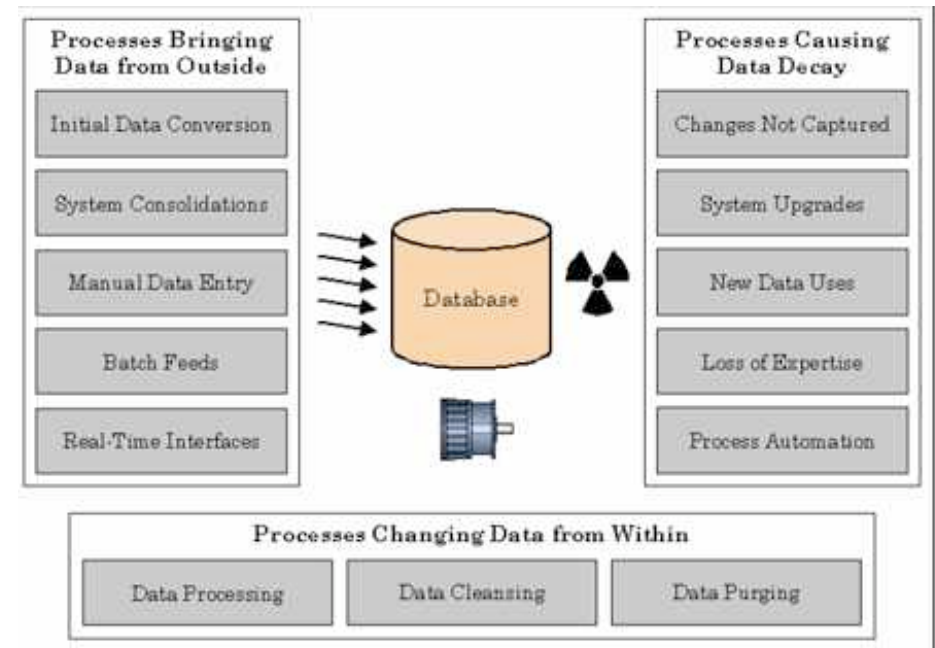


#### 4. Value levels are determined by *timing, architecture effect, and resources*

Value levels: defined as:  
the degree of satisfaction of value needs.

Value level:

- " depends on *when* you observe the level
  - " The environment, the people, other system performance characteristics (security, speed, usability)
- " depends on the *current incremental power of particular value architecture* components
- " depends on *resources available* both in development and operation

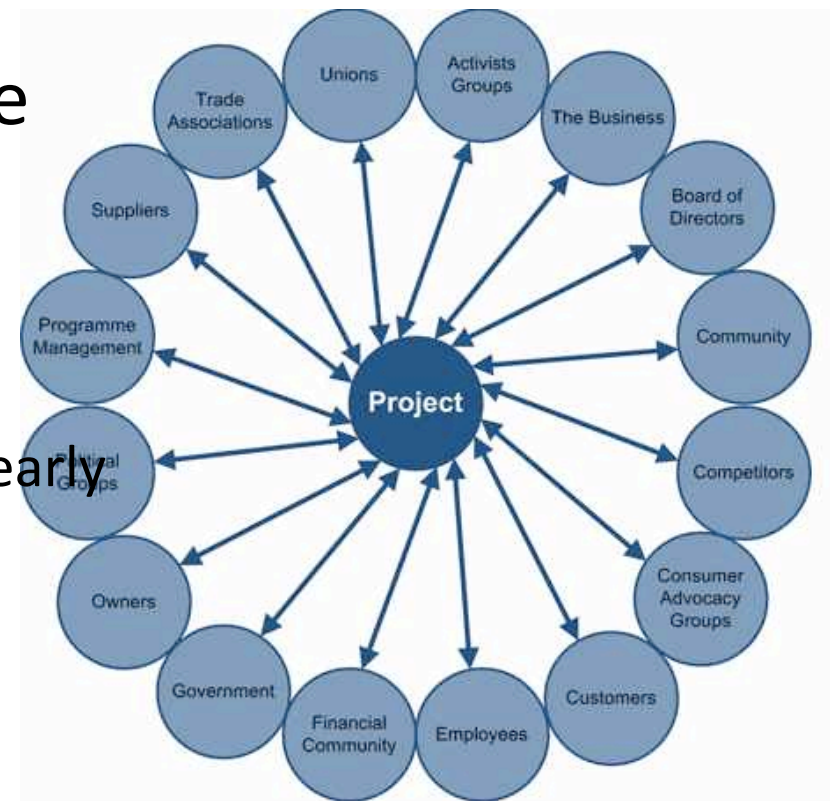


## 5. Required Value *levels* can differ for different scopes (where, who)

The level of value needed, and the level of value delivered - for a single attribute dimension (like Ease of Use) can vary for:

- " different stakeholders
- " at different times
  - " (peak, holiday, slack, emergency, early implementation)
- " for different 'locations'
  - " countries, companies, industries

There is nothing simple like 'one level for all'



- 6. Value can be delivered early

You do not have to wait until 'the project is done' to deliver useful stakeholder value satisfaction.

You can intentionally target the highest priority stakeholders, and their highest priority value area, and levels.

You can deliver them early and continuously

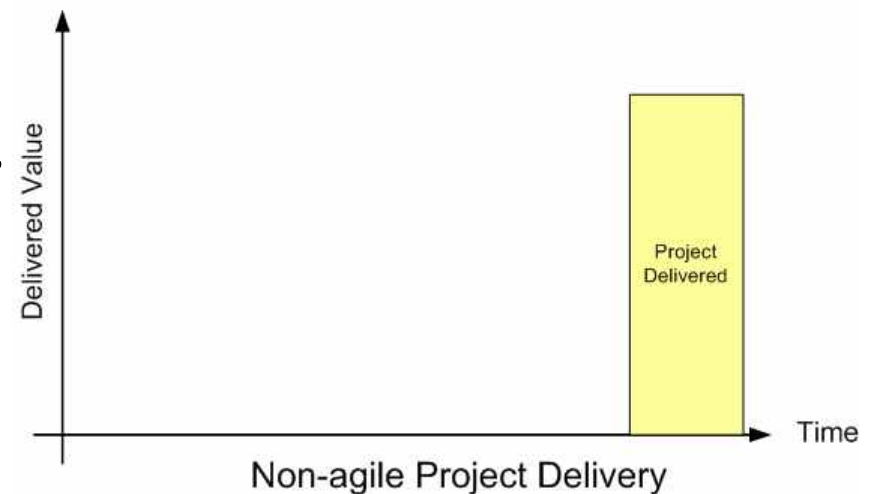
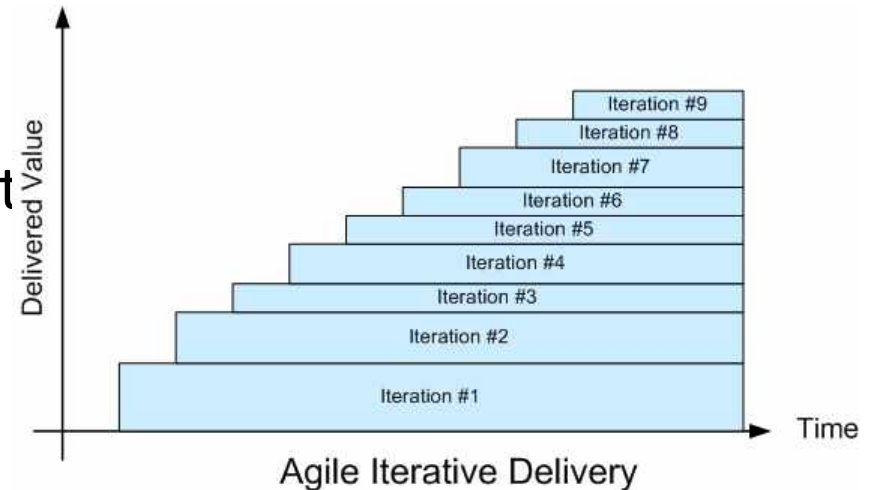
You can learn what is possible

And what stakeholders really value.

Discover new value ideas

Discover new stakeholders

Discover new levels of satisfaction



- 7. Value can be locked in incrementally

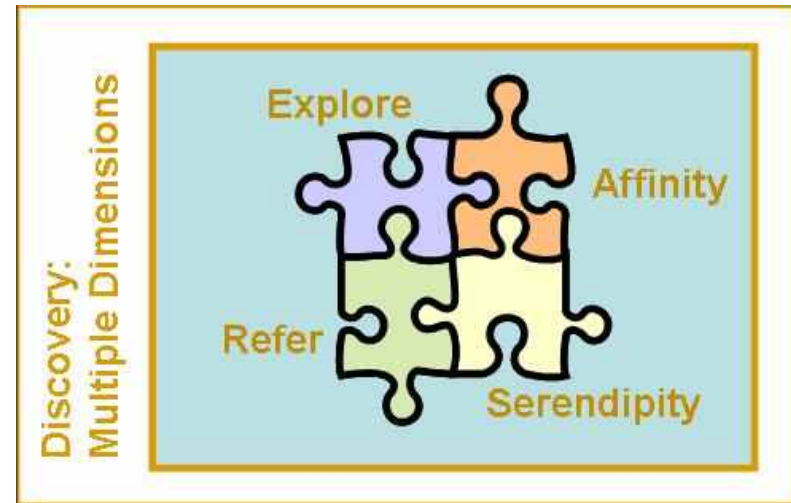
- " You can increment the value satisfaction
  - "*towards* longer term Goal levels
- " You can spread the value deliveries
  - " that are *proven* in *some* places,
  - "more widely in the next increments
- " This probably assumes that you have really handed over real results to real people.
  - "Not just developed systems without delivery





## 8. New Values can be discovered (external news, experience)

- " *Expect*, and try to discover,
  - "entirely new stakeholder values.
- " These will of course emerge *after you start delivering* some satisfaction, because:
  - "Stakeholders believe you can help
  - "Things *change*





## 9. Values can be *evaluated* as a function of *architecture* (using 'Impact Estimation')

- It is possible to get an **overview** of

- the totality of impacts
- that your **architecture**
- (all designs and strategies)
- **might** have
- on all your defined stakeholder needs

Business Objective	Weight	Viking Deliverables											
		hardware adaptation	Telephony	Reference designs	IFace	Modularity	Defend vs Technology 66	Tools	User Experience	GUI & Graphics	Security	Defend vs OCD	Enterprise
Time to market	20%	20%	10%	30%	5%	10%	5%	15%	0%	0%	0%	5%	5%
Mid-range	10%	15%	0%	15%	0%	30%	15%	5%	10%	5%	5%	0%	0%
Platformisation Technology	5%	25%	10%	30%	0%	0%	10%	0%	5%	0%	10%	0%	5%
Interface	5%	5%	15%	15%	0%	5%	0%	5%	0%	0%	10%	0%	10%
Operator preference	10%	0%	10%	0%	15%	5%	20%	5%	10%	10%	20%	5%	10%
Get Torden	10%	25%	10%	10%	-10%	0%	20%	0%	10%	-20%	10%	10%	5%
Commoditisation	5%	20%	10%	20%	10%	-20%	25%	15%	0%	0%	5%	10%	5%
Duplication	10%	15%	10%	10%	0%	0%	40%	0%	0%	0%	5%	20%	5%
Competitiveness	5%	10%	15%	20%	0%	10%	20%	10%	10%	20%	10%	10%	10%
User experience	5%	5%	0%	0%	0%	20%	0%	0%	30%	10%	0%	0%	0%
Downstream cost saving	5%	15%	5%	20%	0%	10%	20%	0%	10%	0%	0%	10%	5%
Platformisation IFace	5%	10%	10%	20%	40%	0%	20%	5%	0%	0%	0%	0%	5%
Japan	5%	10%	5%	20%	0%	10%	0%	0%	10%	5%	0%	0%	0%
Contribution to overall result		15%	9%	17%	4%	7%	15%	6%	6%	1%	6%	6%	5%
Cost (€M)		£ 2.85	£ 0.49	£ 3.21	£ 2.54	£ 1.92	£ 2.31	£ 0.81	£ 1.21	£ 2.68	£ 0.79	£ 0.62	£ 0.60
ROI Index (100=average)		106	358	109	33	78	137	148	107	10	152	202	174

- Use an Impact Estimation table

- and you will be able to spot *opportunities* for
  - high value and
  - low cost early deliveries
  - by analyzing the numbers on the table

See next slide  
For enlargement

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

# Technical Strategies

Objectives		Technical Strategies											
↓ Business Objective Defined In earlier slide		Viking Deferables											
		hardware adaptation	Telephony	Reference designs	IFace	Modularity	Defend vs Technology 66	Tools	User Expe'ce	GUI & Graphics	Security	Defend vs OCD	Enterprise
Time to market		20%	10%	30%	5%	10%	5%	15%	0%	0%	0%	5%	5%
Mid-range		15%	7%	7%	7%	7%	5%	5%	10%	5%	5%	0%	0%
Platformisation Technology		25%	10%	30%	0%		10%	0%	5%	0%	10%	0%	5%
Interface		5%	15%	15%	0%	5%	0%	5%	0%	0%	10%	0%	10%
Operator preference		0%	0%	0%	0%	0%	20%	5%	10%	10%	20%	5%	10%
Get Torden		25%	10%	10%	-10%	0%	20%	0%	10%	-20%	10%	10%	5%
Commoditisation		20%	10%	20%	10%	-20%	25%	15%	0%	0%	5%	10%	5%
Duplication		15%	0%	10%	0%	0%	40%	0%	0%	0%	5%	20%	5%
Competitiveness		10%	15%	20%	0%	10%	20%	10%	10%	20%	10%	10%	10%
User experience		5%	0%	0%	0%	20%	0%	0%	30%	10%	0%	0%	0%
Downstream cost saving		15%	0%	0%	0%	0%	20%	0%	10%	0%	0%	10%	5%
Platformisation IFace		10%	10%	20%	40%	0%	20%	5%	0%	0%	0%	0%	5%
Japan		10%	5%	20%	0%	10%	0%	0%	10%	5%	0%	0%	0%
Contribution to overall result		15%	9%	17%	4%	7%	15%	6%	6%	1%	6%	6%	5%
Cost (£M)		£ 2.85	£ 0.49	£ 3.21	£ 2.54	£ 1.92	£ 2.31	£ 0.81	£ 1.21	£ 2.68	£ 0.79	£ 0.62	£ 0.60
ROI Index (100=average)		106	358	109	33	78	137	148	107	10	152	202	174

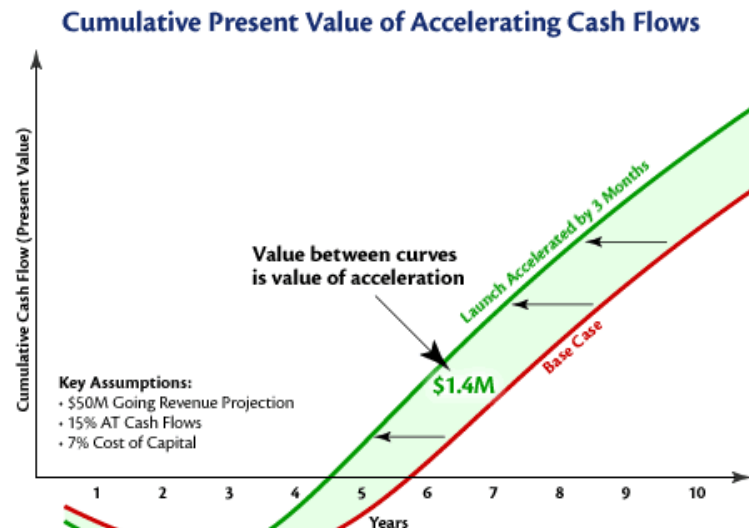
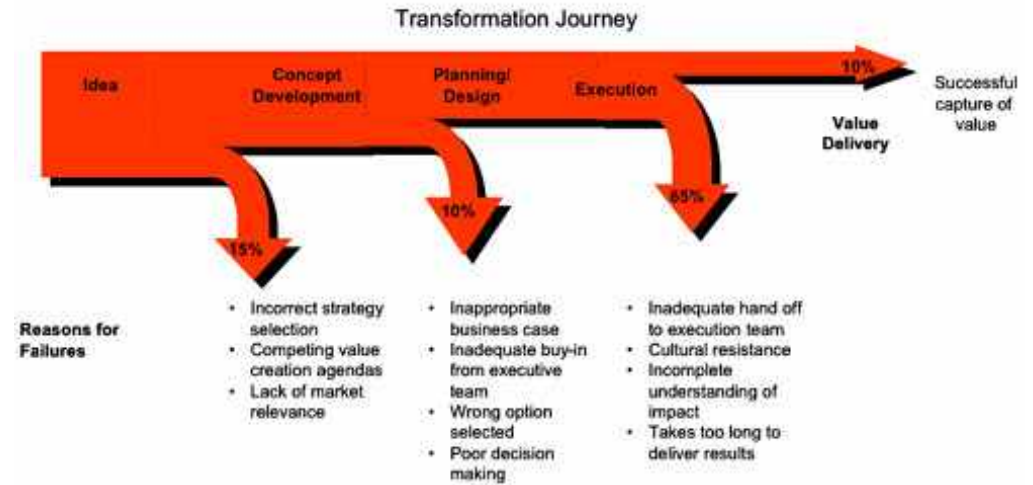
## 10. Value delivery will attract

- " If you are really good at delivering value
  - "You can expect to attract
    - "even more funding
  - "Managers like
    - "to be credited with success
  - " Money seeks
    - "best interest rates



# Gilb's Value Manifesto: A Management Policy?

- 1." Really useful value, for real stakeholders will be defined measurably.  
No nice-sounding emotive words please.
- 2." Value will be seen in light of total long term costs as a decent return on investment.
- 3." Powerful management devices, like motivation and follow-up, will make sure that the value for money is really delivered –  
or that the failure is punished, and the success is rewarded.
- 4." The value will be delivered evolutionarily –  
not all at the end.
- 5." That is, we will create a stream of prioritized value delivery to stakeholders, at the *beginning* of our value delivery projects;  
and continue as long as the real return on investment is suitably large.
- 6." The CEO is primarily responsible for making all this happen effectively.
  - 1." The CFO will be charged with tracking all value to cost progress.
  - 2." The CTO and CIO will be charged with formulating all their efforts in terms of measurable value for resources.



Source "Value Delivery in Systems Engineering" available at [www.gilb.com](http://www.gilb.com)  
 Unpublished paper [http://www.gilb.com/community/tiki-download\\_file.php?fileId=137](http://www.gilb.com/community/tiki-download_file.php?fileId=137)

# The Value Delivery Problem

- " Sponsors who order and pay for systems engineering projects, must justify their money spent based on the expected consequential effects (hereafter called 'value') of the systems.
- "
- " The value of the technical system is often expressed in presentation slides and requirements documents as a set of nice-sounding words, under various titles such as "System Objectives", and "Business Problem Definition"

# Some Assertions

Assertion 1. **When top management allows large projects to proceed, with such badly formulated primary objectives,** then

- " they are responsible as managers for the outcome (failure).
- " They cannot plead ignorance.

Assertion 2. **The failure of technical staff (project management) to react to the lack of primary objective formulation by top management is also a total failure** to do reasonable systems engineering.

- " Management might have a poor requirements culture, but we should routinely save them from themselves.

Assertion 3. **Both top managers and project personnel can be trained and motivated to clarify and quantify critical objectives routinely.**

- " But until the poor external culture of education and practice changes, it may take strong CEO action to make this happen in your corporation.
- " My experience is that no one else will fight for this.

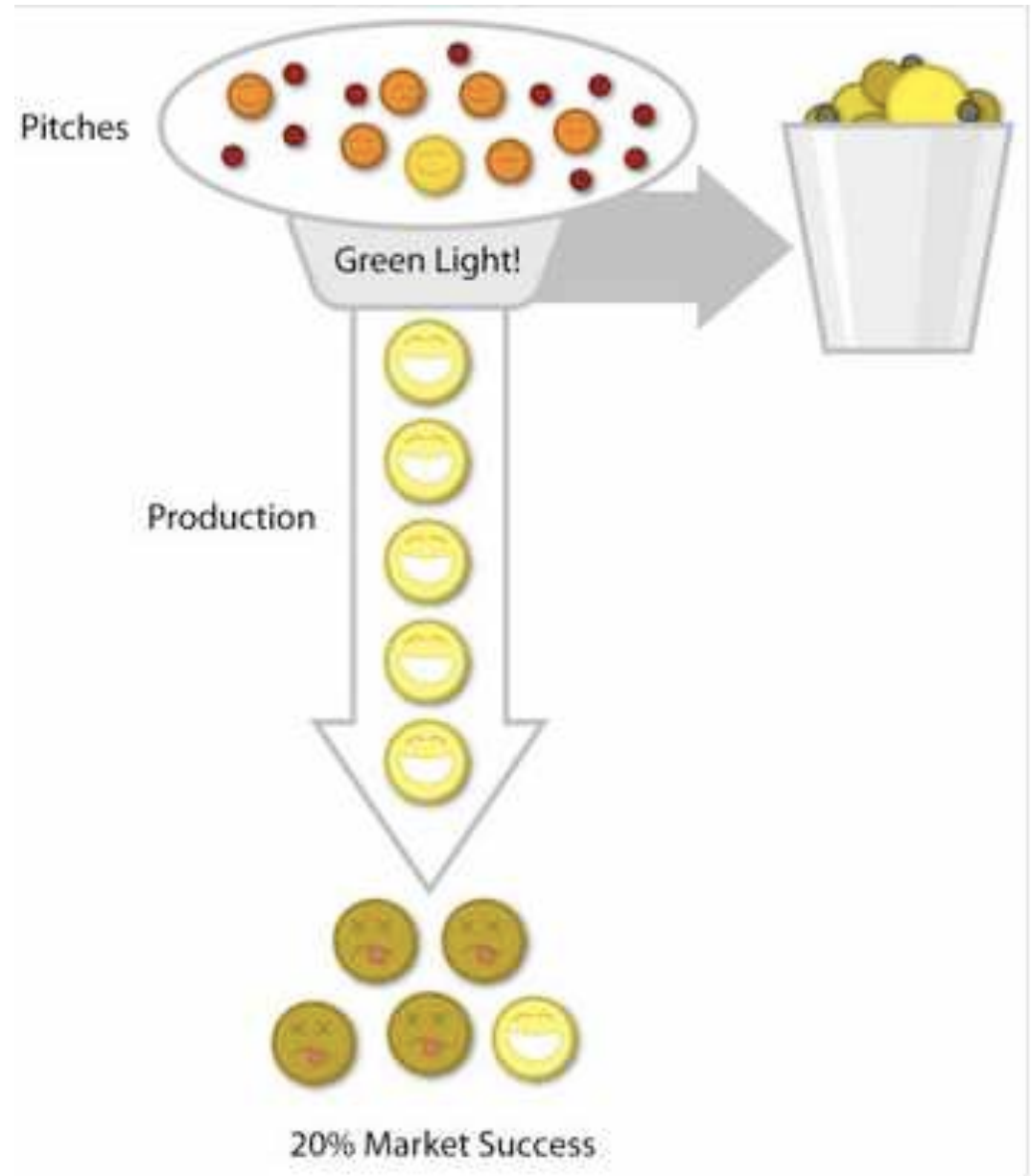
Assertion 4. **All top level system performance improvements, are by definition, variables.**

- " So, we can expect to define them quantitatively.
- " We can also expect to be able to measure or test the current level of performance.
- " Words like 'enhanced', 'reduced', 'improved' are not serious systems engineering requirements terms.

# 6. Stakeholder Needs: quantifying them



# Horror Project Requirements Case

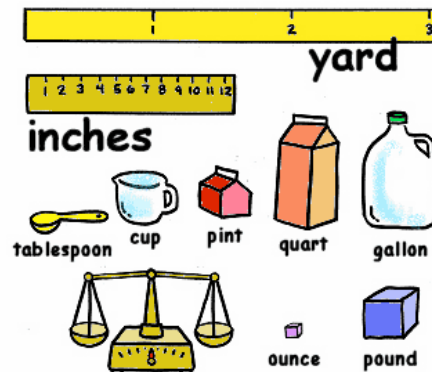


Based On Real Case 2006-8

# Summary of Top '8' Project Objectives

Real Example of **Lack** of Scales

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



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

# The Lesson



- "If management does not clarify the main reasons for a software development project, QUANTITATIVELY,
- "It can cost \$100,000,000+ and 8 years of wasted time

# What the Project Manager Wanted after \$160,000,000\* was spent

**“Able to add features *without fear***

**...**

**Able to improve code *without fear***

**...**

**Able to incorporate improved  
technology *without fear* ...**

**Able to rapidly adapt to changing  
requirements ...**

**Code that's easy to maintain ...**

**Code that's uniform, easy to  
understand ...**

**Code that's readily and thoroughly  
testable ...”**

\* The number was sometimes  
quoted at \$100 million, and by  
2008 it was certainly much  
higher, no deliveries had taken  
place by May 2008.



# What the CIO Director Told Me

*"In 1998 I voted to veto this project start because the requirements were insufficient.*

*But I was overruled by the other directors (including the current CEO)"*

©2002RickLondon/JohannWessels



Lemming rush hour



# Main Hypothesis by Gilb:

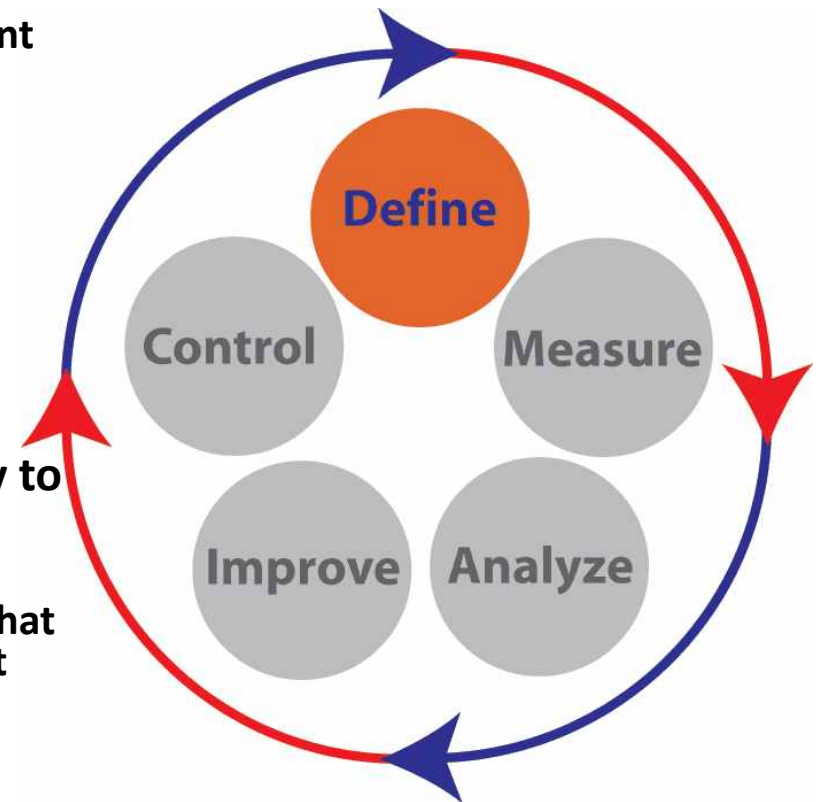
- 1.! The requirements are unacceptably unclear.
  - 1.! They are not defined to any objective level.
  - 2.! the key results are NOT intelligible or NOT testable
  - 3.! They contain far too much *specific design*, instead of the *actual results* that justify investment (results)
  - 4.! The project should never have been approved on such a flimsy basis at the outset.
  - 5.! The CORPORATION has to question its process for review and approval of such expenditure.
  - 6.! The CORPORATION has to question the competence of the highest levels of executives that have allowed this to persist.
  - 7.! You have to worry that many other projects have an equally bad problem of control of results.
2. The project has proceeded to throw masses of detail ('design') at the unacceptably unclear requirements.
3. There is no objective way to decide if any of the built or planned detail is necessary or sufficient to meet the unclear requirements.
4. There is no point whatsoever in continuing the project on this basis (the bad requirements).

Because there is no way to determine if the project is progressing towards any reasonable goals.



# Suggested Practical Actions for HORROR Project.

- 1.! Stop all HORROR Project Effort based on the old plans**
- 2.! Adopt a new 'policy' for running this project**
- 3.! Quickly (in a week or 2) rewrite the top level requirements.**
  - 1.! Review the current business and technical environment to see if new and different requirements are more appropriate than the current (3.13 2003 set)**
  - 2.! Quantify all the top few objectives**
  - 3.! Estimate the value of reaching the objectives**
  - 4.! Get the objectives approved by top management**
    - 1.! This is not the same as project funding approval.**
    - 2.! It just says we would value reaching these objectives**
    - 3.! And we don't know of any better ones.**
- 4.! Let a 'qualified' system architect decide the best way to deliver the results.**
  - 1.! The big question is how much, if any of the current HORROR project investment can be applied, and to what degree the results need to be evolved into the current customer product and environment.**
  - 2.! Approve the architecture**
- 5.! Don't ever pour money into the project unless real measurable improvements are promised and delivered in short cycles.!**





# 1. Seamless ROCKfield data and workflow

Central to THE CORPORATION's ROCKfield business strategy is to **be the world's premier INTEGRATED** ROCKfield service provider. Software is a key enabling technology towards providing this integration. As an active contributor to this overall strategy, Horror will provide the following:

**Broad** MINESITE data coverage.

Horror will be able to tap a **broad variety** of data about the well and its environment. Each of the Horror products will be able to store and exchange all of the following data types, e.g. wireline will be able to access MINING data, etc. These data types include:

• *GILB COMMENT: There is no attempt to define **seamless** quantitatively so that we can **measure** and **track** the final result.*

• *The content of the rest of the requirement is an equally vague set of functional requirements (like "will support standard Windows OLE compound document functionality").*

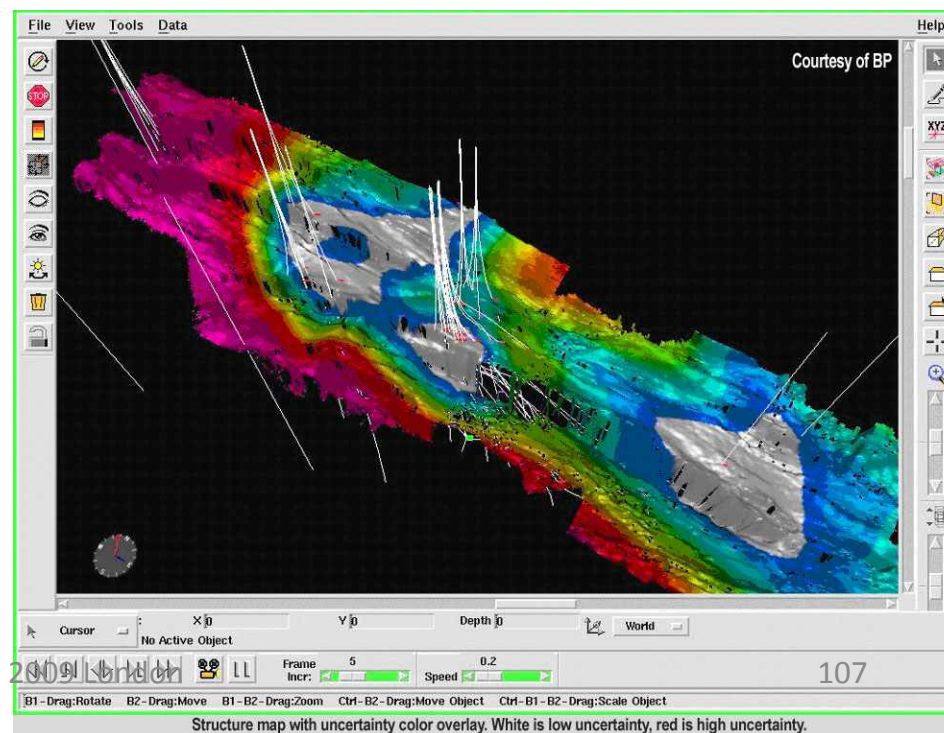
• *It is not at all clear how well these things will be done (no performance or quality requirements for these are mentioned).*

• *The result is likely to be that the function is there but has substandard user quality and performance.*

• *We need to define the user experience – how fast, how easy.*

• *We need to define the end state that would make us the worlds premier provider.*

• *We have not even got close to it.*



## 2. Dramatic boost in operational efficiency

- HORROR will provide a
  - much **more efficient user experience**
  - **by**
  - **automating a number of routine activities**
  - **and by removing restrictions on when or how a number of activities may be performed.**
- **These improvements include:**
- **As-you-go product generation** HORROR will provide the following features
  - to **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
  - **by**
  - semi-automating and/or performing these activities as the data comes in.



### GILB ANALYSIS:

- ❖ *There is no unambiguous definition of 'operational efficiency' (no defined Scale or Scales of measure).*
- ❖ *There is no defined level on that (undefined) scale that tells us what is Dramatic ( and when it is dramatic ( short term levels, longer term levels, competitor levels). Goal, Stretch, Trend levels to use Planguage terms.*
- ❖ *The 'efficient user experience' is not at all defined in terms quantified*
- ❖ *In short this requirement completely fails, where it could have easily succeeded (in 1998)*  
*to specify the level of operational efficiency that the product would measurably achieve.*
- ❖ *The rest of the specification with features like*  
*'Automated depth adjustment for data acquired since last deviation survey'*  
*are merely suggested design elements,*  
*that will only contribute to the operational efficiency if they are well designed and implemented to a defined level of impact on*  
*the (yet undefined quantified definition of operational efficiency).*  
*These design ideas do not belong here at all*  
*(this applies to all the requirements at this level).*  
*They should be in a separate architecture or design specification, that suggested appropriate designs for*

### 3. Much easier to understand and use

**A critical requirement for HORROR's success is to make the software much easier to understand and use than has been the case for previous CORPORATION MINE software.**

Benefits of this requirement include  
    **reduced training time, better utilization of system features**  
    **and fewer operational errors.**

*As an aid in achieving this objective, HORROR has adopted a new use-case centric development process,*

*which makes the users and their use of the system a focal point of the development*

*The intent is to design for and evaluate usability continually during the development process rather than fixing it at the end.*

**(And it goes on about processes and designs)**



•**Gilb Comment:** essentially same criticism as above. This concept could be defined quantitatively (See Usability, Gilb CE Chapter 5, [www.gilb.com](http://www.gilb.com) download).

•" '**To understand**' needs definition (scale) and '**much easier**' needs specification of numeric points on the scale for various users and tasks.

•" The rest of the requirement makes the systemic mistake of diving into **specific design detail ("Minimized panes., Docked and undocked panes, Product generation console"** for example).

•"These are badly defined, and badly justified designs for an undefined problem.

•We would end up building them into the system and there is no guarantee that we would end up getting the 'operational efficiency' we need ( since we have not even decided what we want!).

## ○ 4. Greater software development productivity

- "A primary goal of HORROR is to provide a much more productive software development environment than was previously the case.
- "In addition to traditional software development by professional software personnel,
  - this goal is aimed at facilitating the development of exploratory or custom software or reports by personnel such as tool or interpretation algorithm developers whose software expertise is more modest.
- "A related aspect of this goal is that the software development difficulty should scale,
  - "i.e. simple applications should be easy to develop.

7

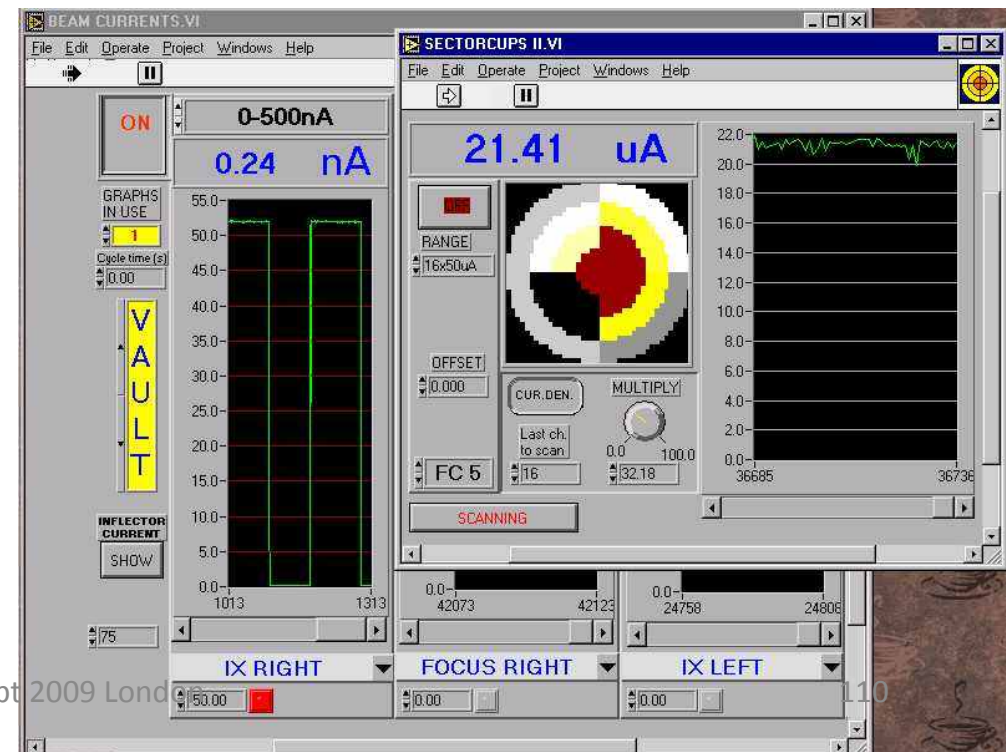
### **GILB COMMENT:**

◆ **SAME COMMENTS AS ABOVE**

◆ ***The Major concept (Productivity) is NOT defined.***

***No level of productivity is numerically and testably set.***

***It could easily be (ask me how!)***





## 5. Rich support for next-generation tools and applications

“HORROR will provide

– “a richer set of functionality

– “for **supporting**

• next-generation logging tools

• and applications.

**Provided features include:**

***Richer equipment model***

**HORROR will**

• provide a

– “richer equipment model that

– “better fits modern hardware configurations.

• *GILB COMMENT:*

– “ Total lack of quantified definition of what this “Supportability” is.

• It could easily be defined as a clear quantified requirement.

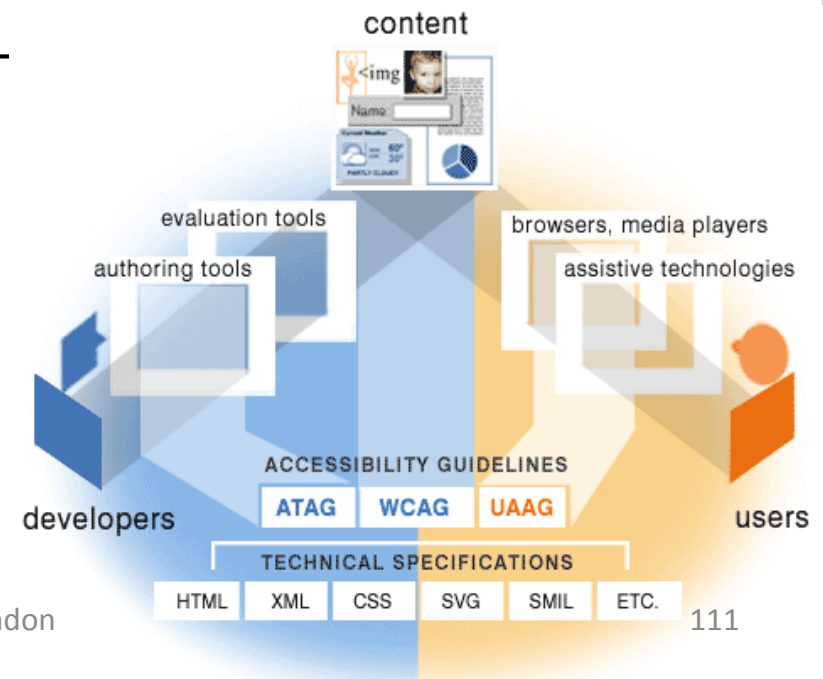
– “ Masses of *nice sounding gratuitous design ideas*

– “unjustified in relation to the (undefined) requirement.

– “ A license to keep on implementing all these things endlessly

– “with no end in sight

– “and no **responsibility** for costs or effects.



# 6. Rock solid robustness

•" While **robustness** is an **essential** HORROR requirement in all its uses, it is especially critical in MINING applications where the much longer job durations afford software defects (e.g. memory leaks) a greatly expanded opportunity to surface.

•" In this regard,  
•HORROR will provide the following features or attributes:

## **Minimal down-time**

•" A critical HORROR objective is to have **minimal downtime due to software failures**.

•This objective includes:

## **Mean time between forced restarts > 14 days**

•" HORROR's goal for mean time between forced restarts **is greater than 14 days**.

•" *Comment: This figure does not include restarts caused by hardware problems, e.g. poorly seated cards or communication hardware that locks up the system. MTBF for these items falls under the domain of the hardware groups.*

## **Restore system state < 10 minutes**

•" Log scripts and test scripts, subsystem tests

## **Built-in testability**

•" HORROR will provide the following features and attributes to facilitate testing.

## **Fool simulators**

•" **GILB COMMENT:**

–" For once a reasonable attempt was made to quantify the meaning of the requirement!

–" But it could be done much better

–"

–" As usual the **set of designs to meet the requirement** do not belong here.

–"And none of them make any **assertion** about how well (to what degree) they will meet the defined numeric requirements.

–" And as usual another guarantee of eternal costs on pursuit of a poorly defined requirements is most of the content.



# Rock Solid Robustness



## **Rock Solid Robustness:**

**Type: *Complex* Product Quality Requirement.**

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





# Software Downtime:

## Software Downtime:

**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 with no 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 able to restore the system to a previously saved state in less than 10 minutes.

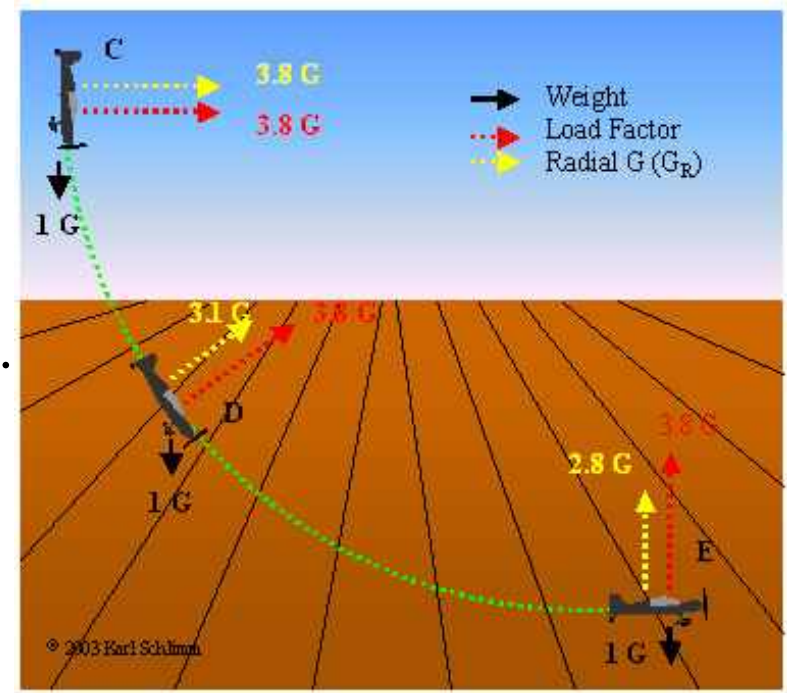
**Scale:** Duration from Initiation of Restore to Complete and verified state of a defined [Previous: Default = Immediately Previous] saved state.

**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.



## 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.

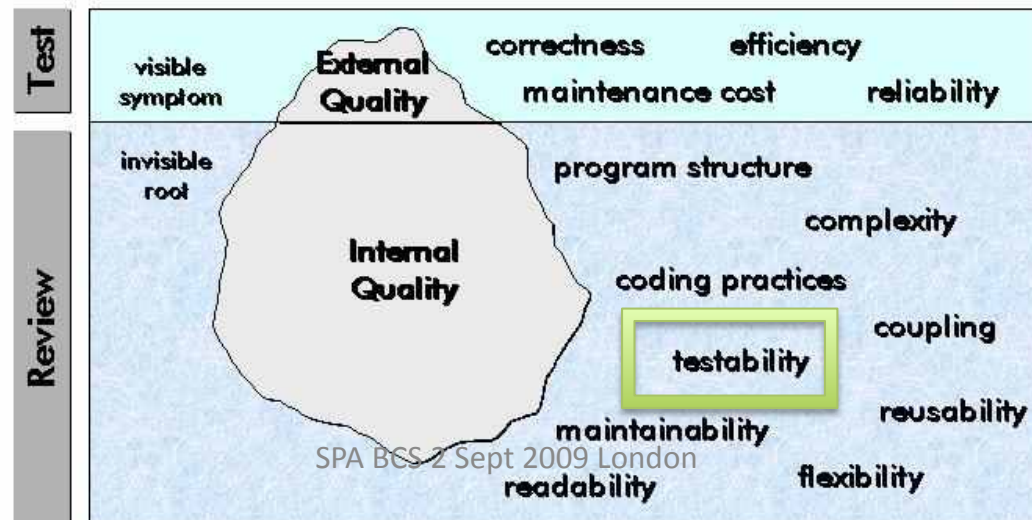
**Scale: the duration of a defined [Volume] of testing, or a defined [Type], by a defined [Skill Level] of system operator, under defined [Operating Conditions].**

**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.

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

## Testability:

### The Software Quality Iceberg



September 12, 2009

SPA BCS 2 Sept 2009 London



# 7. Improved data quality

## *“Quality improvements from job planning*

The inclusion of job planners (see section xxx) as part of the HORROR mandate will provide **major improvements in data quality over current practices** wherein the job planning process is much more haphazard. These improvements include:

- **Client requirements capturing...** “HORROR’s largest step towards improving data quality is freeing the user from many of the mundane system and data management tasks and thereby providing more time to monitor and improve data quality.
- In addition, HORROR will provide the following features and attributes aimed at this goal. (See also section xxx )

• Same critical remarks as other requirements earlier.

• This is **not** clearly defined, not quantified.

• Of course it should and could have been

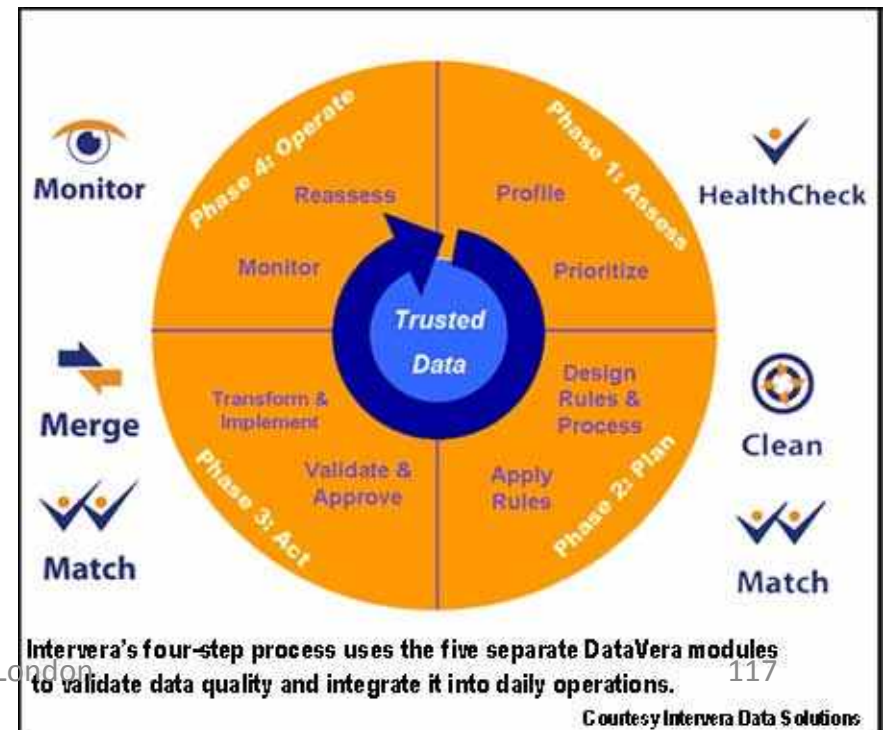
• What is the measure of data quality?

• How much improvement by when are we thinking of.

• (one poster session AL DATA QUALITY, IS an good example of deeper thought on this vital subject)

• The usual detailed designs (“More flexible measure point support”) themselves need quantified definition to be clear and powerful.

• So again masses of things to spend money on for badly defined purposes.





# Project Manager Says

Hi Tom, I did receive your analysis and had the proverbial good intentions to reply but did not, so do apologize.

And I further apologize for taking so long to reply to this -- has been a hectic week on top of a busy two months...

Given the scope of your recommendations, I am not terribly surprised that you did not receive a response from upper management -- am certain that they intend to "fix" the project in their own way.

We are, at our level, trying to improve our development processes, and

**I am advocating that we understand and incorporate your principles in our working standards from here on out**

I do appreciate the starting point you given us.

Thanks again, and I hope you have a good holiday season

Sxxxx

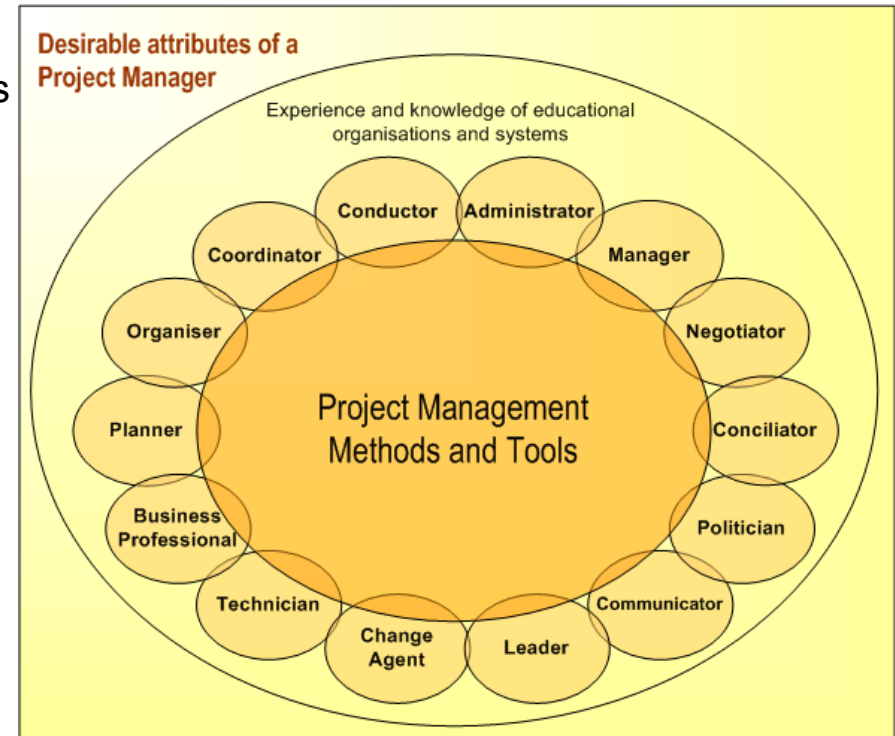
About December 2006



# 22 April 2008

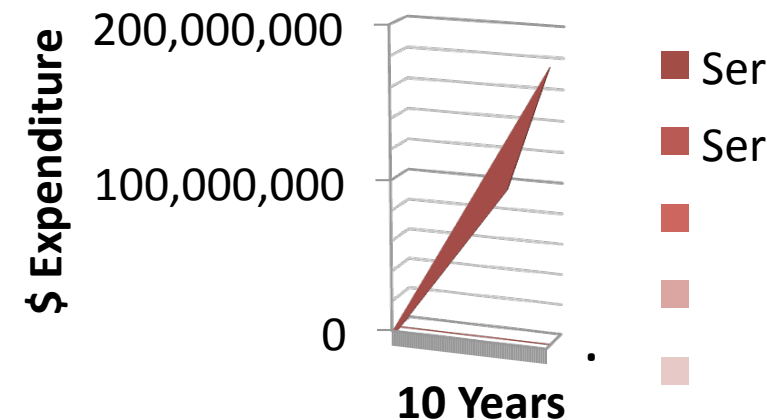
## Project Manager Looks Back

- " Hi Tom, Sorry that I didn't pick this up last night so hope you check your email this morning.
- " **Our project is on sound track.**
- " Requirements aside, when you visited, our code base was "unstable" due to too much development with too little qc.
- " We stopped,
  - " stabilized the code,
  - " emphasized inspections,
  - " and quite significantly, the "powers that be"
  - " replaced the PM with another,
  - " quite well respected,
  - " and with considerably more immediate domain experience.
- " We also **focused** and **shortened our delivery cycle.**
- " To that end I see that we essentially have done much,
- " but not all, of what you suggested in your report
  - " (but certainly not with the **requirements rigor that you advocate – still an issue**).
- " I would like to think that your advice had an influence on the outcome however much of the directive came from levels on high to which I'm not privy.
- " Fyi, M is very familiar with our project. Best regards,S



# \$100-180 million+ Wasted

- " The above example was the basis in 1999 for a project that had
  - " in 2006 spent over \$100 million,
  - " for 8 years
  - " and had never delivered any value whatsoever to the corporation.
- " There was never any quantified or testable definition of the requirements.
- " There was never any direct link
  - " from the project activity, requirements, or architecture,
  - " to these primary top management
    - " (CEO and next level directors) objectives.
- " The project was doomed from the start.

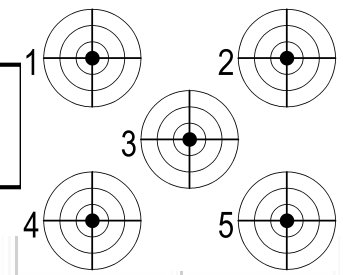
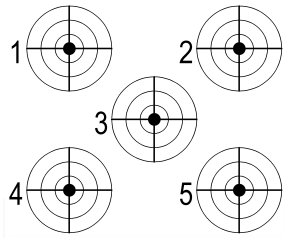




# Top Manager Objectives

- " Here is an example of a CEO Level Plan to get £50 million from outside owners, in order to invest in organizational productivity and quality improvement for 800 software engineers producing a telecoms product.





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

Business objective	Measure	Goal (200X)	Stretch goal ('0X)	Volume	Value	Profit	Cash
Time to market	Normal project time from GT to GT5	<9 mo	<6 mo	X		X	X
Mid-range	Min BoM for The Corp phone	<\$9	<3	X			X
Platformisation Technology	# of Technology 66 Lic. shipping > 3M/yr	4	6	X		X	X
Interface	Interface units	>11M	>13M	X		X	X
Operator preference	Top-3 operators issue RFQ spec The Corp	7	5	X		X	X
Productivity				X		X	X
Get Torden	Lyn goes for Technology 66 in Sep-04	Yes		X		X	X
Fragmentation	Share of components modified	<10%	<5%		X	X	X
Commoditisation	Switching cost for a UI to another System	>1y	>6 m			X	X
Duplication	The Corp share of 'in scope' code in best-selling device	>90%	>95%		X	X	X
Competitiveness	Major feature comparison with MX	Same	Better	X		X	X
User experience	Key use cases superior vs. competition	5	10	X	X	X	X
Downstream cost saving	Project ROI for Licensees	>33%	>66%	X	X	X	X
Platformisation IFace	Number of shipping Lic.	33	55	X		X	X
Japan	Share of of XXX sales	>50%	>60%	X		X	X

Numbers are intentionally changed from real ones

**Business Objectives Quantified**

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

# Technical Strategies



Objectives		Technical Strategies											
Business Objective		Viking Deliverables											
		hardware adaptation	Telephony	Reference designs	IFace	Modularity	Defend vs Technology 66	Tools	User Exper'ce	GUI & Graphics	Security	Defend vs OCD	Enterprise
Time to market	1	20%	10%	30%	5%	10%	5%	15%	0%	0%	0%	5%	5%
Mid-range	2	15%	10%	30%	5%	10%	5%	5%	10%	5%	5%	0%	0%
Platformisation Technology	3	25%	10%	30%	0%	5%	10%	0%	5%	0%	10%	0%	5%
Interface	4	5%	15%	15%	0%	5%	0%	5%	0%	0%	10%	0%	10%
Operator preference	5	0%	10%	10%	0%	0%	20%	5%	10%	10%	20%	5%	10%
Get Torden		25%	10%	10%	-10%	0%	20%	0%	10%	-20%	10%	10%	5%
Commoditisation		20%	10%	20%	10%	-20%	25%	15%	0%	0%	5%	10%	5%
Duplication		15%	10%	10%	0%	0%	40%	0%	0%	0%	5%	20%	5%
Competitiveness		10%	15%	20%	0%	10%	20%	10%	10%	20%	10%	10%	10%
User experience		5%	10%	0%	0%	0%	0%	0%	30%	10%	0%	0%	0%
Downstream cost saving		15%	10%	20%	0%	0%	20%	5%	10%	0%	0%	10%	5%
Platformisation IFace		10%	10%	20%	40%	0%	20%	5%	0%	0%	0%	0%	5%
Japan		10%	5%	20%	0%	0%	0%	0%	10%	5%	0%	0%	0%
Contribution to overall result		15%	9%	17%	4%								5%
Cost (£M)		£ 2.85	£ 0.49	£ 3.21	£ 2.54	£ 1.92	£ 2.31	£ 0.81	£ 1.21	£ 2.68	£ 0.79	£ 0.62	£ 0.60
ROI Index (100=average)		106	358	109	33	78	127	148	107	10	152	202	174

# The CEO Got His Money

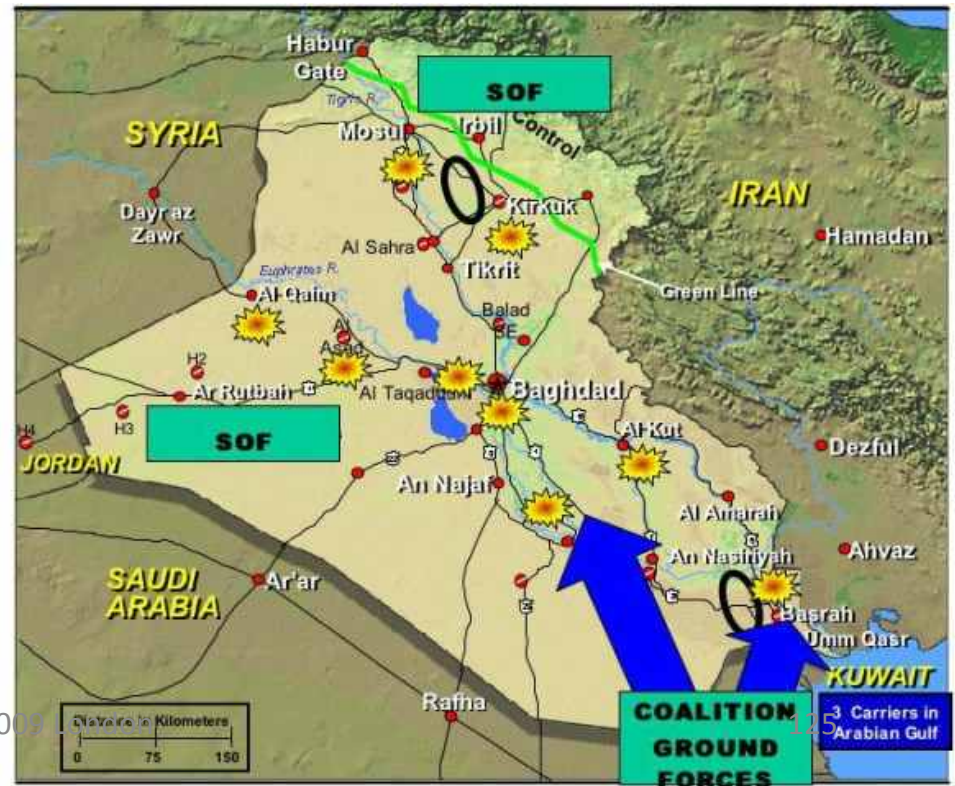


- " Showing Financial People
  - "Exactly what you will do for their money
  - "Is a powerful way to sell complex technology
  - "Sell them the results THEY are interested in
  - "Show them Value for money
    - " Not Techie Expenditure
  - "Be prepared to be responsible for delivering the numbers you claim you can deliver
    - " Then maybe you will get funded next time too!

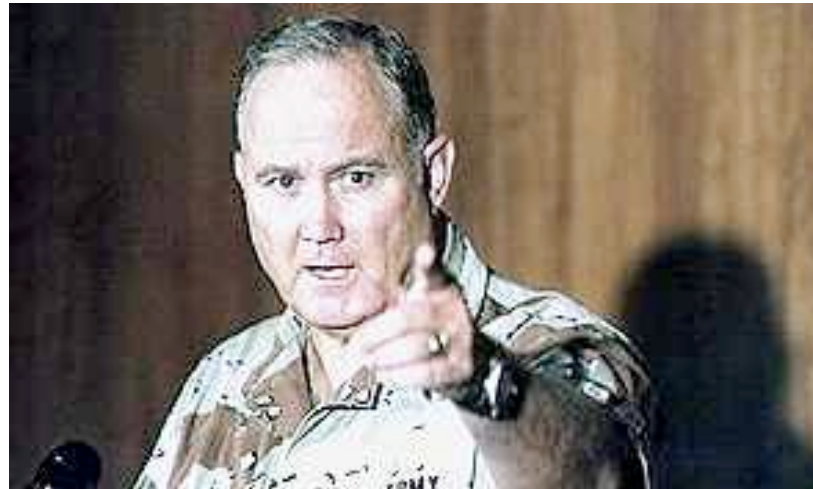


# And Now A True War Story (and an Agile Evo Case)

- "About Why Bad IT Requirements
  - "Can lose a war in Iraq
  - "Or at least make it drag on for years



# The Persinscom IT System Case



He who does not learn from history  
September 12, 2009  
Is doomed to repeat it



SPA BCS 2 Sept 2009 London

A Man Who understood that  
"a bird in the hand is worth two in the Bush" <-tsg

126

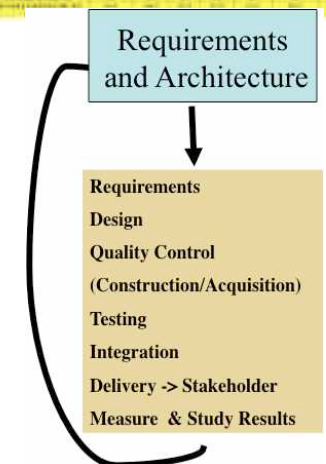




- 

US Army Example: PERSINSCOM

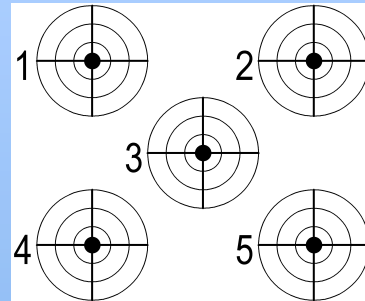
Year 1990	Year 1991	Year 1992	Year 1993	Year 1994	Year 1995	Year 1996	Year 1997	Year 1998	Year 1999	Year 2000	Year 2001	Year 2002	Year 2003	Year 2004	Year 2005	Year 2006	Year 2007	Year 2008	Year 2009	Year 2010	Year 2011	Year 2012	Year 2013	Year 2014	Year 2015	Year 2016	Year 2017	Year 2018	Year 2019	Year 2020	Year 2021	Year 2022	Year 2023	Year 2024	Year 2025	Year 2026	Year 2027	Year 2028	Year 2029	Year 2030	Year 2031	Year 2032	Year 2033	Year 2034	Year 2035	Year 2036	Year 2037	Year 2038	Year 2039	Year 2040	Year 2041	Year 2042	Year 2043	Year 2044	Year 2045	Year 2046	Year 2047	Year 2048	Year 2049	Year 2050	Year 2051	Year 2052	Year 2053	Year 2054	Year 2055	Year 2056	Year 2057	Year 2058	Year 2059	Year 2060	Year 2061	Year 2062	Year 2063	Year 2064	Year 2065	Year 2066	Year 2067	Year 2068	Year 2069	Year 2070	Year 2071	Year 2072	Year 2073	Year 2074	Year 2075	Year 2076	Year 2077	Year 2078	Year 2079	Year 2080	Year 2081	Year 2082	Year 2083	Year 2084	Year 2085	Year 2086	Year 2087	Year 2088	Year 2089	Year 2090	Year 2091	Year 2092	Year 2093	Year 2094	Year 2095	Year 2096	Year 2097	Year 2098	Year 2099	Year 2100
1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100



# US Army Example: PERSINSCOM: Personnel System



STRATEGIES →
OBJECTIVES
Customer Service ? → 0 Violation of agreement
Availability 90% → 99.5% Up time
Usability 200 → 60 Requests by Users
Responsiveness 70% → ECP's on time
Productivity 3:1 Return on Investment
Morale 72 → 60 per mo. Sick Leave
Data Integrity 88% → 97% Data Error %
Technology Adaptability 75% Adapt Technology
Requirement Adaptability ? → 2.6% Adapt to Change
Resource Adaptability 2.1M → ? Resource Change
Cost Reduction FADS → 30% Total Funding



Monday  
← The Top Ten  
Critical Objectives  
Were decided

# Sample of Objectives/Strategy definitions

## US Army Example: PERSINCOM: Personnel System



- " *Example of one of the Objectives:*

### Customer Service:

**Type:** Critical Top level Systems Objective

**Gist:** Improve customer perception of quality of service provided.

**Scale:** Violations of Customer Agreement per Month.

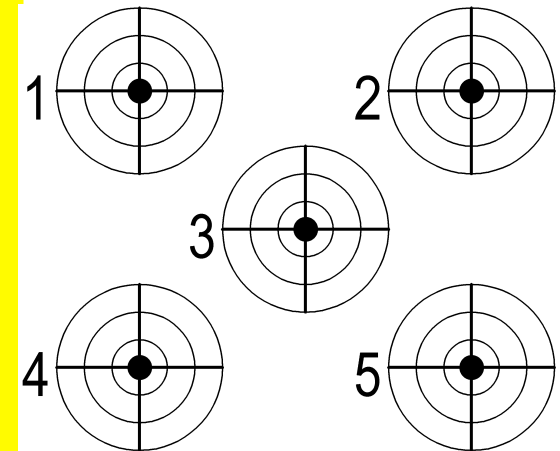
**Meter:** Log of Violations.

**Past** [Last Year] Unknown Number ← State of PERSCOM Management Review

**Record** [NARDAC] 0 ? ← NARDAC Reports Last Year



**Fail** : <must be better than Past, Unknown number> ← CG

**Goal** [This Year, PERSINCOM] 0 "Go for the Record" ← Group SWAG



# US Army Example: PERSINSCOM: Personnel System



STRATEGIES →  OBJECTIVES	Technology Investment	Business Practices	People	Empow- erment	Principles of IMA Management	Business Process Re- engineering	SUM
Customer Service ? → 0 Violation of agreement	<div> <p>Tuesday</p> <p>The Top Ten</p> <p>Critical Strategies</p> <p>For reaching the</p> <p>← objectives</p> <p>Were decided</p> </div>  						
Availability 90% → 99.5% Up time							
Usability 200 → 60 Requests by Users							
Responsiveness 70% → ECP's on time							
Productivity 3:1 Return on Investment							
Morale 72 → 60 per mo. Sick Leave							
Data Integrity 88% → 97% Data Error %							
Technology Adaptability 75% Adapt Technology							
Requirement Adaptability ? → 2.6% Adapt to Change							
Resource Adaptability 2.1M → ? Resource Change							
Cost Reduction FADS → 30% Total Funding							

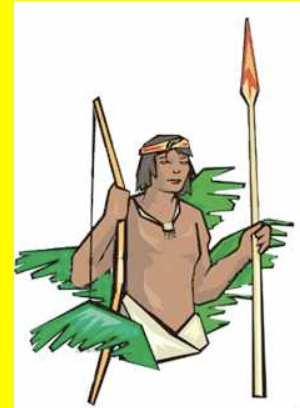


## *A Strategy (Top Level of Detail)*

### **Technology Investment:**

**Gist:** Exploit investment in high return technology.

**Impacts:** productivity, customer service and conserves resources.





# Wednesday:

## Day 3 of 5 of 'Feasibility Study

- " We made a rough evaluation
  - " of how powerful our strategies might be
  - " in relation to our objectives
- " Impact Estimation Table
  - " 0% Neutral, no  $\pm$  impact
  - " 100% Gets us to Goal level on time
  - " 50% Gets us half way to Goal at deadline
  - " -10% has 10% negative side effect

STRATEGIES → OBJECTIVES	Technology Investment	Business Practices	People	Empowerment	Principles of IMA Management	Business Process Re-engineering	SUM
Customer Service ? → 0 Violation of agreement	50%	10%	5%	5%	5%	60%	185%
Availability 90% → 99.5% Up time	50%	5%	5-10%	0	0	200%	265%
Usability 200 → 60 Requests by Users	50%	5-10%	5-10%	50%	0	10%	130%
Responsiveness 70% → ECP's on time	50%	10%	90%	25%	5%	50%	180%
Productivity 3:1 Return on Investment	45%	60%	10%	35%	100%	53%	303%
Morale 72 → 60 per mo. Sick Leave	50%	5%	75%	45%	15%	61%	251%
Data Integrity 88% → 97% Data Error %	42%	10%	25%	5%	70%	25%	177%
Technology Adaptability 75% Adapt Technology	5%	30%	5%	60%	0	60%	160%
Requirement Adaptability ? → 2.6% Adapt to Change	80%	20%	60%	75%	20%	5%	260%
Resource Adaptability 2.1M → ? Resource Change	10%	80%	5%	50%	50%	75%	270%
Cost Reduction FADS → 30% Total Funding	50%	40%	10%	40%	50%	50%	240%
<b>SUM IMPACT FOR EACH SOLUTION</b>	<b>482%</b>	<b>280%</b>	<b>305%</b>	<b>390%</b>	<b>315%</b>	<b>649%</b>	
Money % of total budget	15%	4%	3%	4%	6%	4%	
Time % total work months/year	15%	15%	20%	10%	20%	18%	
<b>SUM RESOURCES</b>	<b>30</b>	<b>19</b>	<b>23</b>	<b>14</b>	<b>26</b>	<b>22</b>	
<b>BENEFIT/RESOURCES RATIO</b>	<b>16:1</b>	<b>14:7</b>	<b>13:3</b>	<b>27:9</b>	<b>12:1</b>	<b>29:5</b>	



MEASURING HAND FOR GLOVE SIZE



# DoDef. Persinscom Impact Estimation Table:

## Designs

<i>Design Ideas -&gt;</i>	<i>Technology Investment</i>	<i>Business Practices</i>	<i>People</i>	<i>Empowerment</i>	<i>Principles of IMA Management</i>	<i>Business Process Re-engineering</i>	<i>Sum Requirements</i>
<b>Requirements</b>	50%	10%	5%	5%	5%	60%	185%
Availability 90% <-> 99.5% Up time	50%	5%	5-10%	0%	0%	200%	265%
Usability 200 <-> 60 Requests by Users	50%	5-10%	5-10%	50%	0%	10%	130%
Responsiveness 70% <-> ECP's on time	50%	10%	90%	25%	5%	50%	180%
Productivity 3:1 Return on Investment	45%	<b>R → D Impacts</b>			100%	53%	303%
Morale 72 <-> 60 per month on Sick Leave	50%				15%	61%	251%
Data Integrity 88% <-> 97% Data Error %	42%	10%	25%	5%	70%	25%	177%
Technology Adaptability 75% Adapt Technology	5%	30%	5%	60%	0%	60%	160%
Requirement Adaptability ? <-> 2.6% Adapt to Change	80%	20%	60%	75%	20%	5%	260%
Resource Adaptability 2.1M <-> ? Resource Change	10%	80%	5%	50%	50%	75%	270%
Cost Reduction FADS <-> 30% Total Funding	50%	40%	10%	40%	50%	50%	240%
<i>Sum of Performance</i>	<i>482%</i>	<i>280%</i>	<i>305%</i>	<i>390%</i>	<i>315%</i>	<i>649%</i>	
Money % of total budget	15%	4%	3%	4%	6%	4%	36%
Time % total work months/year	15%	15%	20%	10%	20%	18%	98%
<i>Sum of Costs</i>	<i>30</i>	<i>19</i>	<i>23</i>	<i>14</i>	<i>26</i>	<i>22</i>	
<i>Performance to Cost Ratio</i>	<i>16:1</i>	<i>14:7</i>	<i>13:3</i>	<i>27:9</i>	<i>12:1</i>	<i>29:5</i>	

# US Army Example: PERSINSCOM: Personnel System



<b>STRATEGIES → OBJECTIVES</b>	Technology Investment	Business Practices	People	Empow- erment	<i>Principles of IMA Management</i>	Business Process Re- engineering	SUM
Customer Service ? → 0 Violation of agreement	50%	10%	5%	5%	5%	60%	185%
Availability 90% → 99.5% Up time	50%	5%	5-10%	0	0	200%	265%
Usability 200 → 60 Requests by Users	50%	5-10%	5-10%	50%	0	10%	130%
Responsiveness 70% → ECP's on time	50%	10%	90%	25%	5%	50%	180%
Productivity 3:1 Return on Investment	45%	60%	10%	35%	100%	53%	303%
Morale 72 → 60 per mo. Sick Leave	50%	5%	75%	45%	15%	61%	251%
Data Integrity 88% → 97% Data Error %	42%	10%	25%	5%	70%	25%	177%
Technology Adaptability 75% Adapt Technology	5%	30%	5%	60%	0	60%	160%
Requirement Adaptability ? → 2.6% Adapt to Change	80%	20%	60%	75%	20%	5%	260%
Resource Adaptability 2.1M → ? Resource Change	10%	80%	5%	50%	50%	75%	270%
Cost Reduction FADS → 30% Total Funding	50%	40%	10%	40%	50%	50%	240%
<b><i>SUM IMPACT FOR EACH SOLUTION</i></b>	<b><i>482%</i></b>	<b><i>280%</i></b>	<b><i>305%</i></b>	<b><i>390%</i></b>	<b><i>315%</i></b>	<b><i>649%</i></b>	
Money % of total budget	15%	4%	3%	4%	6%	4%	
Time % total work months/year	15%	15%	20%	10%	20%	18%	
<b><i>SUM RESOURCES</i></b>	<b><i>30</i></b>	<b><i>19</i></b>	<b><i>23</i></b>	<b><i>14</i></b>	<b><i>26</i></b>	<b><i>22</i></b>	
<b>BENEFIT/RESOURCES RATIO</b>	<b><i>16:1</i></b>	<b><i>14:7</i></b>	<b><i>13:3</i></b>	<b><i>27:9</i></b>	<b><i>12:1</i></b>	<b><i>29:5</i></b>	

# Thursday:

## Day 4 of 5 of 'Feasibility Study

- " We looked for a way to deliver some stakeholder results, next week
- " 1 1 1 1
  - "1 increase from 0%
  - "1 stakeholder
  - "1 quality
  - "1 week

<b>STRATEGIES → OBJECTIVES</b>	Technology Investment	Business Practices	People	Empow- erment	<i>Principles of IMA Management</i>	Business Process Re- engineering	SUM
Customer Service ? → 0 Violation of agreement	50%	10%	5%	5%	5%	60%	185%
Availability 90% → 99.5% Up time	50%	5%	5-10%	0	0	200%	265%
Usability 200 → 60 Requests by Users	50%	5-10%	5-10%	50%	0	10%	130%
Responsiveness 70% → ECP's on time	50%	10%	90%	25%	5%	50%	180%
Productivity 3:1 Return on Investment	45%	60%	10%	35%	100%	53%	303%
Morale 72 → 60 per mo. Sick Leave	50%	5%	75%	45%	15%	61%	251%
Data Integrity 88% → 97% Data Error %	42%	10%	25%	5%	70%	25%	177%
Technology Adaptability 75% Adapt Technology	5%	30%	5%	60%	0	60%	160%
Requirement Adaptability ? → 2.6% Adapt to Change	80%	20%	60%	75%	20%	5%	260%
Resource Adaptability 2.1M → ? Resource Change	10%	80%	5%	50%	50%	75%	270%
Cost Reduction FADS → 30% Total Funding	50%	40%	10%	40%	50%	50%	240%
<b>SUM IMPACT FOR EACH SOLUTION</b>	<b>482%</b>	<b>280%</b>	<b>305%</b>	<b>390%</b>	<b>315%</b>	<b>649%</b>	
Money % of total budget	15%	4%	3%	4%	6%	4%	
Time % total work months/year	15%	15%	20%	10%	20%	18%	
<b>SUM RESOURCES</b>	<b>30</b>	<b>19</b>	<b>23</b>	<b>14</b>	<b>26</b>	<b>22</b>	
<b>BENEFIT/RESOURCES RATIO</b>	<b>16:1</b>	<b>14:7</b>	<b>13:3</b>	<b>27:9</b>	<b>12:1</b>	<b>29:5</b>	



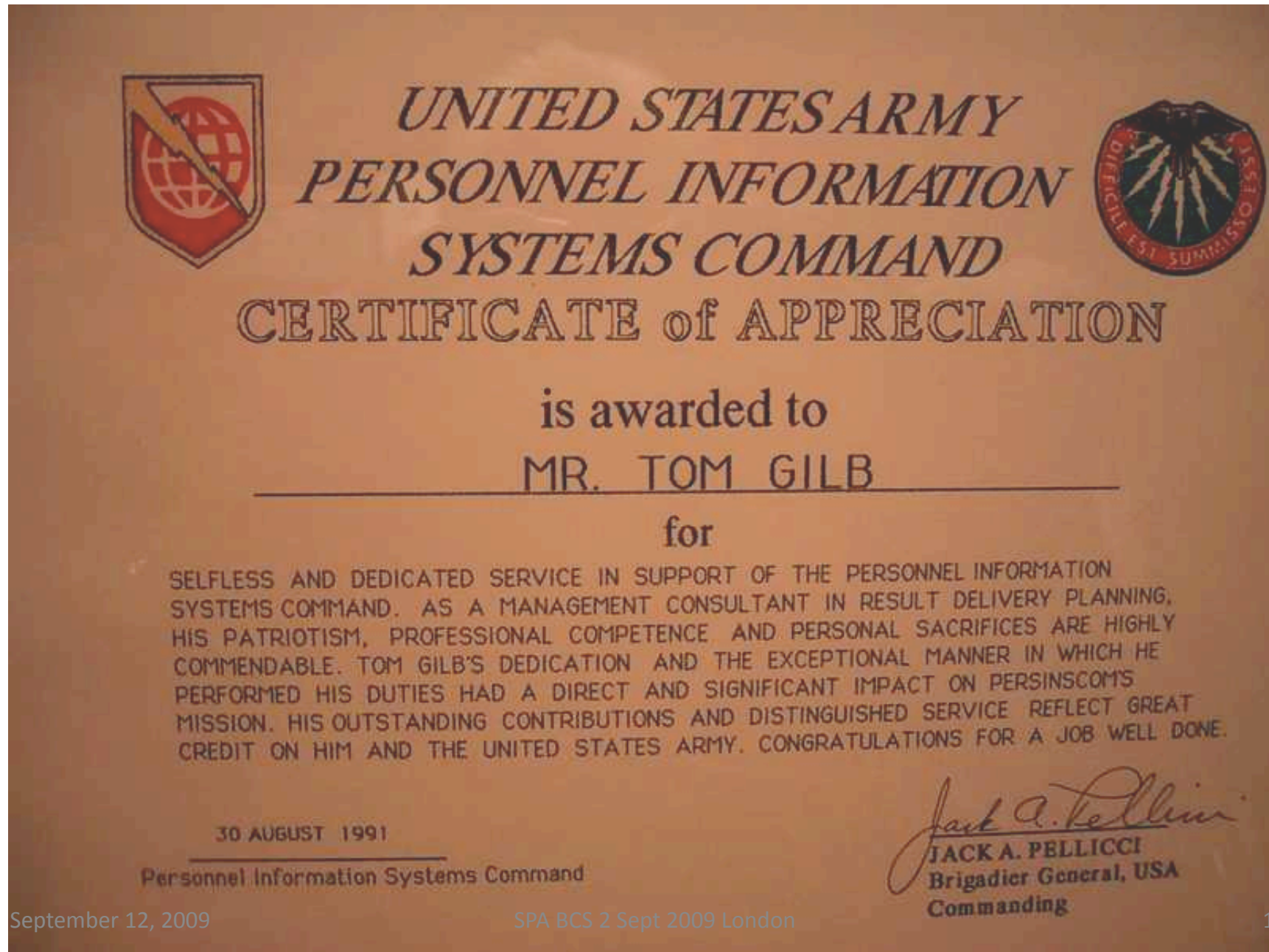
# Next weeks Evo Step??

- " "You won't believe we never thought of this, Tom!"
- " The step:
  - " When the Top General Signs in
  - " Move him to the head of the queue
    - " Of all people inquiring on the system.





# Thanks!



# Software Engineering Productivity Study

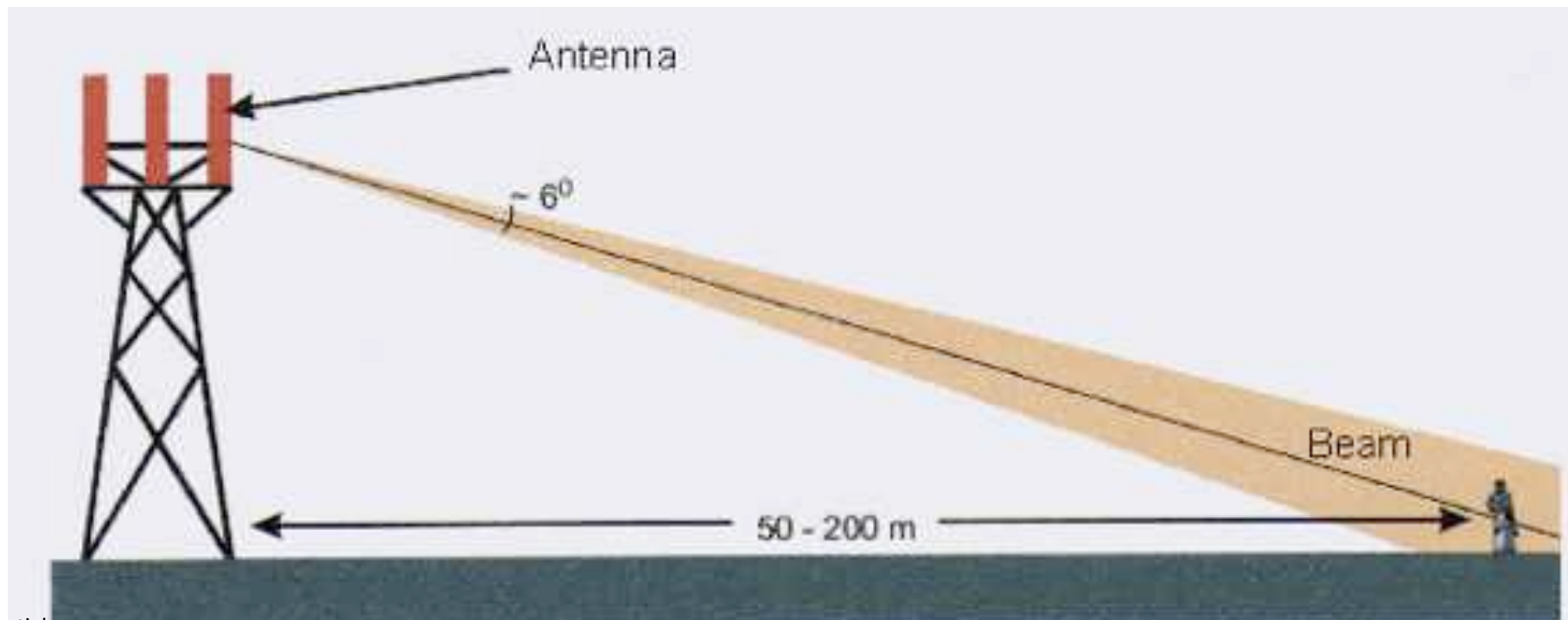
# ERICSSON



An example of setting objectives for process improvement

For 1997 with 70% software labor development content in products

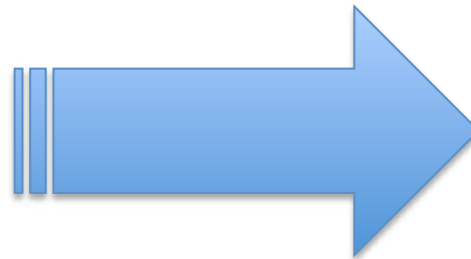
**THIS CASE SHOWS EVO USING SMALL IMMEDIATE INCREMENTS**





# The problem

- " Great Market Growth Opportunities
- " Too Few Software Engineers
- " Solution:
  - "Increase productivity of existing engineers



# **The One Page Top Management Summary (after 2 weeks planning) The Dominant Goal**

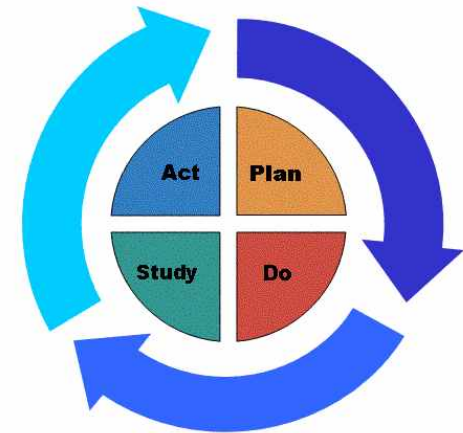
Improve Software Productivity in R PROJECT by 2X by year 2000

## **Dominant (META) Strategies**

Continual Improvement (PDSA Cycles)

.DPP: Defect Prevention Process

.EVO: Evolutionary Project Management



## **Long Term Goal [1997-2000+]**

DPP/EVO, Master them and Spread them on priority basis.

## **Short Term Goal [Next Weeks]**

DPP [ RS?]

EVO [Package C ?]

**Decision: {Go, Fund, Support}**



# The Ericsson Quality Policy:



"every company shall define performance indicators (which) ..

–"reflect **customer satisfaction**,

–"internal **efficiency**

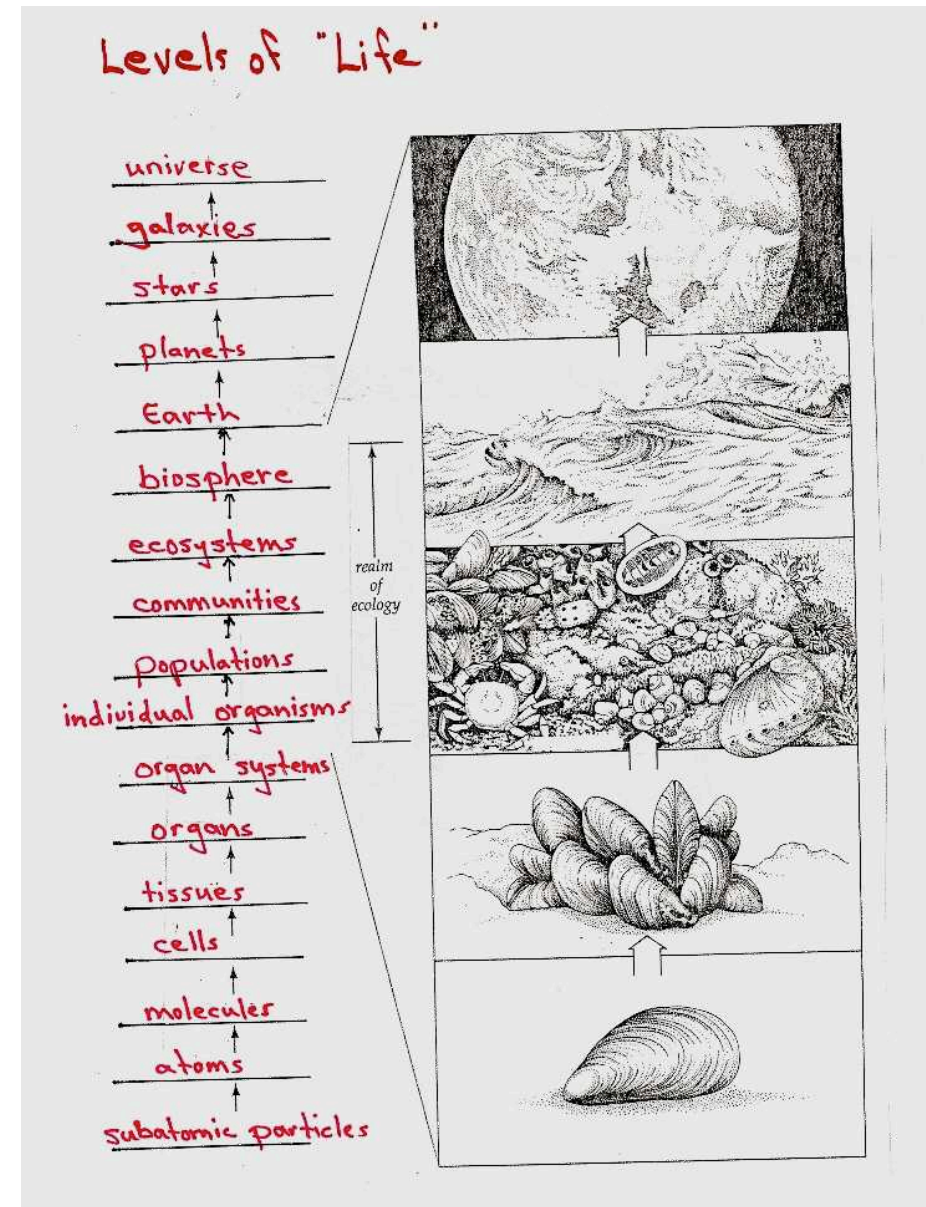
–"and business **results**.

•"The performance indicators are used in **controlling** the operation."

•"Quality Policy [4.1.3]

# Levels of Objectives.

- " **Fundamental Objectives**
- " **Strategic Objectives**
- " **Means Objectives:**
- "
- " **Organizational Activity Areas.**
  - " Pre-study.
  - " Feasibility Study.
  - " Execution.
  - " Conclusion.
- " **Generic Constraints**
  - " Political Practical
  - " Design Strategy Formulation Constraints
  - " Quality of Organization Constraints
  - " Cost/Time/Resource Constraints







# Keeney's: Levels of objectives.

## —" 1. Fundamental Objectives

- " (above us)

## —" 2. Generic Constraints

- " (our given framework)
- " Political Practical
- " Design Strategy Formulation Constraints
- " Quality of Organization Constraints
- " Cost/Time/Resource Constraints

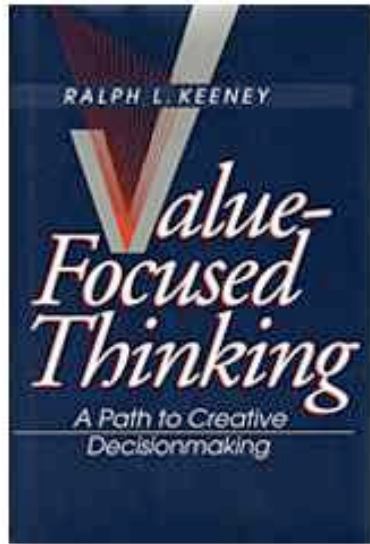
## —" 3. Strategic Objectives

- " (objectives at our level)

## —" 4. Means Objectives:

- " (*supporting* our objectives)

**Constraints**



# The Strategic Objectives (CTO level)

## —"Support

- "the **Fundamental Objectives**  
(Profit, survival)
- "**Software Productivity:**
  - "Lines of Code Generation  
Ability
- "**Lead-Time:**
- "**Predictability.**
- "**TTMP: Predictability of  
Time To Market:**
- "**Product Attributes:**
- "**Customer Satisfaction:**
- "**Profitability:**





# 'Means' Objectives:

—"Support the **Strategic** Objectives

- *"Complaints:*
- *"Feature Production:*
- *"Rework Costs:*
- *"Installation Ability:*
- *"Service Costs:*
- *"Training Costs:*
- *"Specification Defectiveness:*
- *"Specification Quality:*
- *"Improvement ROI:*



*"Let no man turn aside,  
ever so slightly,  
from the broad path of honour,  
on the plausible pretence  
that he is justified by the goodness  
of his end.*

*All good ends can be worked out  
by good means."*

*Charles Dickens*

# Strategies: (total brainstormed list)

## 'Ends for delivering Strategic Objectives'

- Evo [Product development]:
- DPP [Product Development Process]:  
Defect Prevention Process.
- Inspection?
- Motivation.Stress-Management-AOL
- Motivation.Carrot
- DBS
- Automated Code Generation
- Requirement -Tracability
- Competence Management
- Delete-Unnecessary -Documents
- Manager Reward:?
- Team Ownership:?
- Manager Ownership:?



- Training:?
- Clear Common Objectives:?
- Application Engineering area:
- Brainstormed List (not evaluated or prioritized yet)?
- Requirements Engineering:
- Brainstormed Suggestions?
- Engineering Planning:
- Process Best Practices:
- Brainstormed Suggestions?
- Push Button Deployment:
- Architecture Best Practices:
- Stabilization:
- World-wide Co-operation?

# Principles for Prioritizing Strategies

- " They are well-defined
  - " Not vague
- " They have some relevant predictable numeric experience
  - " On main effects
  - " Side effects
  - " Costs
  - " Risks - Uncertainty
- " Not huge spread of experience



"Software Productivity" =

# Lines of Code Generation Ability

—"Software Engineering net production in relation to corresponding costs."

—"Ambition: Net lines of code successfully produced per total working hours needed to produce them. A measure of the

—"efficiency" ('effective production/cost of production') of the organization in using its software staff

•Scale: [Defined Volume, kNCSS or kPlex] per

•Software Development: Defined:

•Productivity calculations include Work-Hours

•"Meter : <PQT Database and EPOS, CPAC>

—"Comment: we know that real software production is available in our current culture. AB, PK, T

—"P1: Past [ 1997, ERA/AR ] < to be calculated v

•" Past-R PROJECT: Past [ 1997, R PROJECT ] < to be calculated when data available, available Volume/Work Hours >

•"Past-EEI: Past [1997, Ireland, Plex] \_\_\_\_??\_\_ kPLEX / Work-Hour.

•"add more like LuleÅ"

•Fail [end 1998, R PROJECT, Same Reliability] 1.5 x Past-R PROJECT  
<- R PROJECT AS 3 c " by 50%".

—"50% better useful code productivity in 1.5 years overall"

•Same Reliability: State: The Software Fault Density is not worse than with comparable productivity. Use official The Company Software Fault Density measures <- 1997 R PROJECT Balanced Scorecard (PA3).

•Goal [Year=2000, R PROJECT, Same Reliability] 2 x Past-R PROJECT,

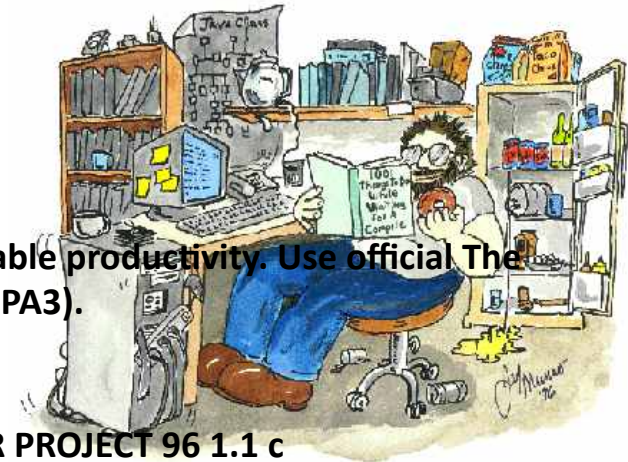
—"Year=2005, RPL, Same Reliability] 10?? x Past-R PROJECT

•Wish [Long term, vs. D pack.] 10 x Past-R PROJECT "times higher productivity" <- R PROJECT 96 1.1 c

•Wish [undefined time frame] 1.5 x Past-R PROJECT <- R PROJECT AS 3 c " by 50%"

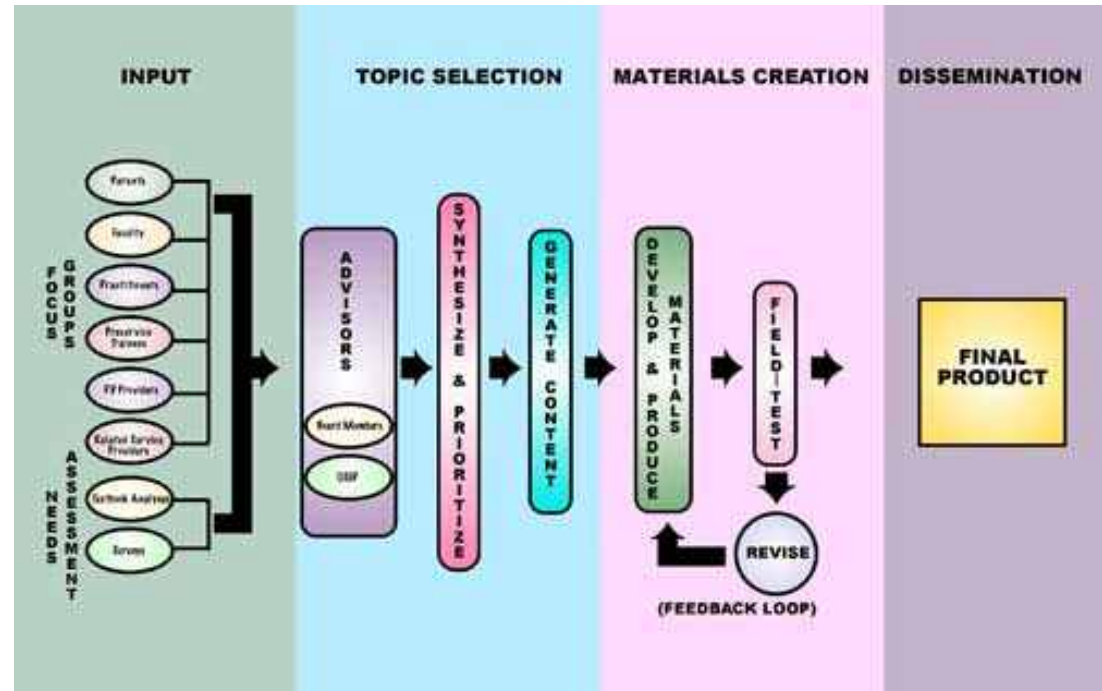
—"Comment: May 13 1997 1600, We have worked a lot on the Software Productivity objectives (all day) and are happy that it is in pretty good shape. But we recognize that it needs more exposure to other people.

**Scale: [Defined Volume, kNCSS or kPlex] per Software Development Work-Hour.**



- " Lead-Time:
  - " *"Months for major Packages"*
- " *Ambition: decrease months duration between major Base Station package release.*
- " **Scale: Months from TG0, to successful first use for**
  - " **major work station package.**
  - " *Note: let us make a better definition. TG*
- " Past [C Package, 1996?] 20? Months?? <-guess tg
- " Goal [D-package] 18 months <- guess tg
- " Goal [E-package and later] 10.8 Months <- R PROJECT 96 1.1 a "40% > D"
- " Goal [Generally] ??? <- R PROJECT AS 3a
  - " *"10% Lead-Time reduction compared to any benchmark".*

# Lead-Time:





# Predictability of Time To Market:

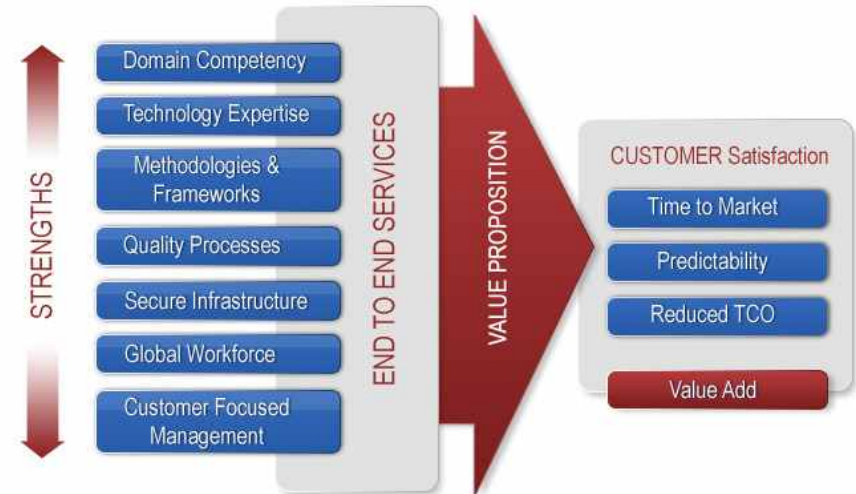
- " TTMP: Predictability of Time To Market:

- " *Ambition: From Ideas created to customers can use it. Our ability to meet agreed specified customer and self-determined targets.*

- " **Scale: % overrun of actual Project Time compared to planned Project Time**

- " **Project Time: Defined:** time from the date of Toll-Gate 0 passed, or other Defined Start Event, to, the Planned- or Actually- delivered Date of All [Specified Requirements], and any set of agreed requirements.
    - " **Specified Requirements: Defined:** written approved Quality requirements for products with respect to Planned levels and qualifiers [when, where, conditions]. And, other requirements such as function, constraints and costs.

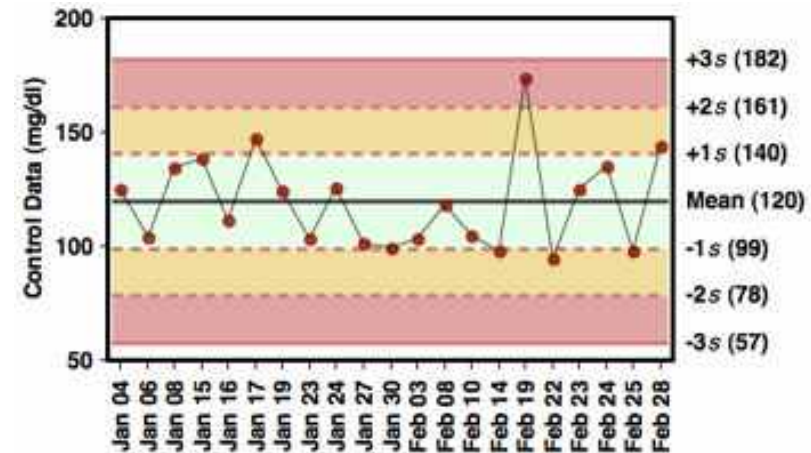
- " **Meter: Productivity Project or Process Owner will collect data from all projects, or make estimates and put them in the Productivity Database for reporting this number.**
  - " Past [1994, A-package] < 50% to 100%> <- Palli K. guess. [1994, B-package] 80% ?? <- Urban Fagerstedt and Palli K. guess
  - " Record [IBM Federal Systems Division, 1976-80] 0% <- RDM 9.0 quoting Harlan Mills in IBM SJ 4-80
  - " *"all projects on time and under budget"*
  - " [Raytheon Defense Electronics, 1992-5] 0% <- RDE SEI Report 1995 Predictability.
  - " Fail [All future projects, from 1999] 5% or less <- discussion level TG
  - " Goal [All future projects, from 1999] 0% or less <- discussion level TG





# Product Attributes:

- **Product Attributes:**
  - “*Keeping Product Promises.*”
  - *Ambition: Ability to meet or beat agreed targets, both cost, time and quality. (except TTMP itself, see above)*
- **Scale: % +/- deviation from [defined agreed attributes with projects].**
- **Past [1990 to 1997, OUR DIVISION] at least 100% ???**
  - “<- Guess. Not all clearly defined and differences not
    - tracked. TSG
- **Goal [Year=2000, R PROJECT] near 0% negative deviation <- TsG for discussion.**



Westgard Procedure Warning Rules	
Run Accepted	151

# Customer Satisfaction

## Customer Satisfaction:

*“Customer Opinion of Us”*

**Scale: average survey  
result on scale  
of 1 to 6 (best)**

Meter: The Company  
Customer

Satisfaction Survey

Past [1997] **4**

Goal [1998-9?] **5** <- R  
PROJECT 96 1.1 b



# Profitability

- "Profitability:
  - "Return on Investment."
  - "Ambition: Degree of *saleable product ready for installation.*
  - "Scale: Money Value of Gross Income derived by
    - "[All R PROJECT Production OR
    - " defined products] for
    - " [Product Lifetime OR
    - "a defined time period]
  - "Goal: <we did not complete this>



‘Means Objectives’ Samples  
They use the  
same *definition* process  
as we use for the higher level objectives



# Means Objectives

- " *“support Strategic Objectives”*
- " Summary:
  - " **Means Objectives'** are
    - " not our major Strategic Objectives (above),
    - " but each one represents areas which if improved
      - " will normally help us achieve our Strategic Objectives.
  - " Means Objectives have a lower priority than Strategic Objectives.
  - " They must never be 'worked towards'
    - " to the point where they reduce our ability to meet Strategic Objectives.





# Complaints

## Complaints:

*"Customer complaint rate to us"*

## Ambition:

Means Goal: for Customer Satisfaction  
(Strategic).

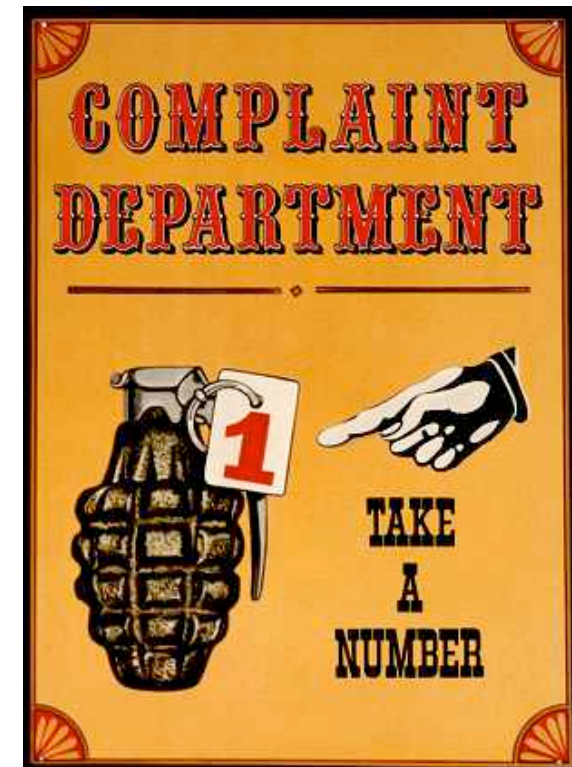
**Scale:** number of complaints per customer in  
[defined time into <operation>]

**Past** [Syracuse Project , 1997] ?? <bad> <- ML

**Goal** [Long term, software component, in first  
6 months in Operation] **zero complaints** <-  
R PROJECT 96 1.1 b

"zero complaints on software features"

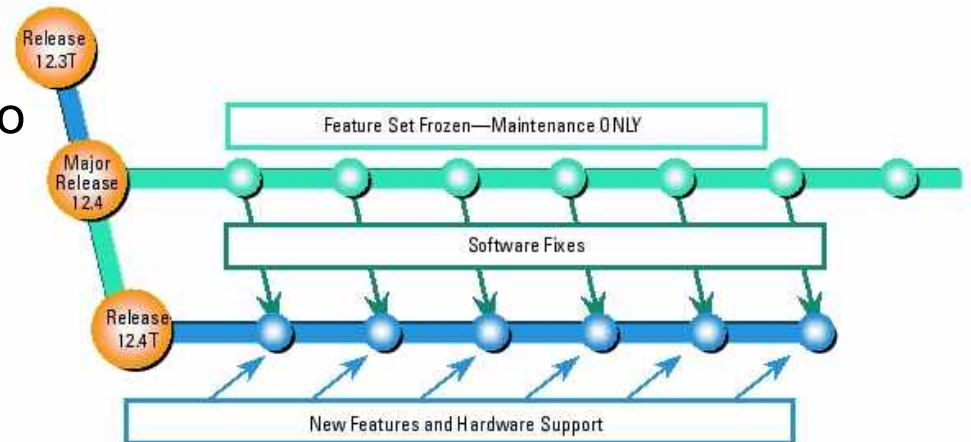
Impacts: <one or more strategic objectives>





# Feature Production:

- " **Feature Production:**
  - "ability to deliver new features to customers"
  - " **Ambition:** reverse our decreasing ability to deliver new features <- R PROJECT AS 1.1
  - " **Scale:** Number of new prioritized <Features> delivered successfully to customer per year per software development engineer.
  - " Too Little: Past [1997] ?? "estimate needed, maybe even definition of feature"
  - " **Goal** [1998-onwards] **Too Little + 30% annually??** <-For discussion purposes TsG.
  - " "we need to drastically change our ability to effectively develop SW" <- R PROJECT AS 1.1



**Note:** Technology releases are those Cisco IOS Software releases that introduce new features, functionality, and hardware support.

# Improvement ROI:

## Improvement ROI:

*"Engineering Process Improvement Profitability"*

*Ambition: Order of magnitude return on investment in process improvement.*

## Scale:

**The average [annual OR defined time term] Return on Investment in Continuous Improvement as a ratio of [Engineering Hours OR Money]**

*Note: The point of having this objective is to remind us to think in terms of real results for our process improvement effort, and to remind us to prioritize efforts which give high ROI. Finally, to compare our results to others. <-TsG*

## Record

\_\_\_\_ [Shell NL, Texas Instruments , Inspections] 30:1 <- Independently published papers TsG

## Past

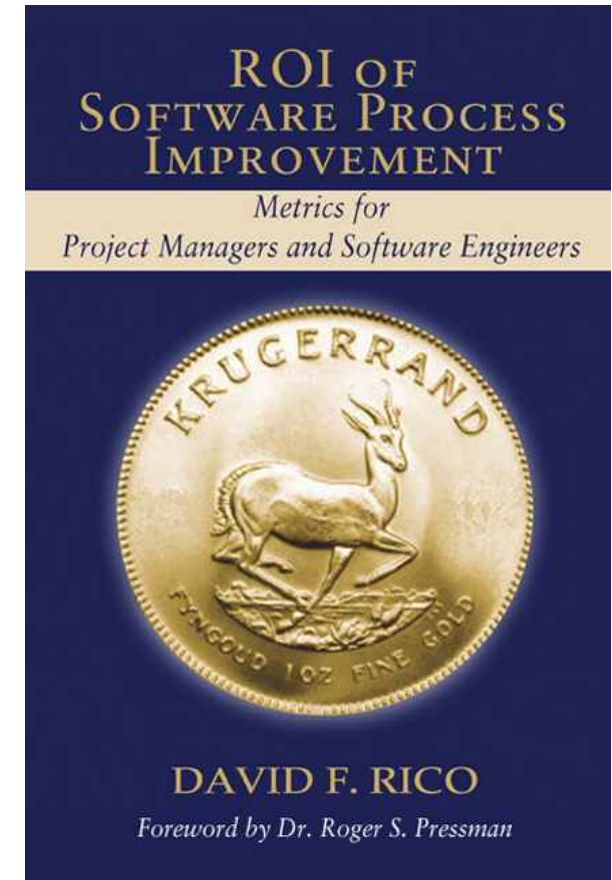
\_\_\_\_ [IBM RTP, 1995, DPP Process] **13:1** <- Robert Mays, Wash DC test conference slides TsG

[Raytheon, 1993-5, Inspection & DPP] **\$7.70:1** <- RDE Report page 51 (\$4.48 M/\$0.58M) Includes detail on how calculated. PK has copy.

[IBM STL, early 1990's] Average 1100% ROI (**11:1**) <- IBM Secrets pp32. PK has copy. NB Conservative estimate. See Note IBM ROI below.

September 12, 2009

SPA BCS 2 Sept 2009 London



2004

# Financial IT Examples of Top Management Planning

# The 'Official' Forgotten CIO Objectives: \$60 Million in 1 Year

The business problem and opportunities to be addressed are:

**Business Problem**



- Achieve "One Bank" vision through globally integrated IT Portfolio Management, by implementation of a single toolset supporting existing (and consistent) processes across [redacted] IT.
- Perform accurate measurement and tracking of project and non-project related IT expenses.
- Track and allocate human resources based on skills, level of work commitment and timing.
- Enable Business alignment through the ability to manage critical initiatives on a portfolio basis and support faster time to market, providing the potential for increase in revenues.
- Enable the business and SMT to make sound management decisions around the portfolio and optimize IT spend so as to effectively prioritize IT spend and maximize business value.
- Replace resource intensive and disparate Portfolio Management tools with industry "best in breed" capabilities.
- Improvement in the time it takes IT to respond to business changes.
- Reduction in costs through eliminating redundant projects.
- Better planning and tracking capabilities so as to reduce project cost and time overruns.

# Initial CIO Objectives

## **Benefits:**

**Reduce the costs associated with managing redundant / regionally disparate systems.  
Single global portfolio management system.**

**Reduce overall spending with a reduction in redundant initiatives.  
Governance structures - system agnostic.  
All projects in IT Portfolio system.**

**Reduce IT spend on low priority work with better alignment between IT and business demand.  
IT Portfolio Framework, Business Value metrics for prioritization.**

**Reduction in cost over runs.  
Definition criteria for project success.  
Metrics and exception reporting for cost management.  
Linkage of actual costs to forecast.**

**Increase revenue with a faster time to market.  
Knowledge management, project ramp up templates.  
Provide quantitative & qualitative benefits. State the consequences of project cancellation.  
These need quantification, and then a plan for delivery and delivery measurement focus – on results not the process.**

# Notes PM: The Objectives

1. COO wanted us to write up the objective he gave on the fly, and that's what he will present to CIO.

## **EXTRACT OF COO 4 OBJECTIVES:**

CIO has shifted from One IT , to 'don't let my view on that stand in the way of <getting results>. <- COO.

1 of 3 billion of new demand.

1. Make sure it is for key business goals,
2. avoid duplication,
3. not re-inventing the wheel
4. I am interested in the MIS. Id like some good metrics about what's coming off the 1 billion production line, (are we delivering on time, under budget, are customer satisfied, and are we delivering the value).<- COO My View

If we were using Evo delivery, for most of the billion, and if I am wasting 40% 400 million/year) Id like to know and deploy it better.

What is the cost of failure of processes used today. Where do they come from (Requirements or what). <Root cause>  
I do not feel comfortable (am flying blind) we have the metrics to manage the 1 billion. Where is my compliance for processes ( have requirements been inspected). I might use The Tool for this. <- COO

my process; work on COO 4 goals, then check with previous The Tool objectives.



## Reminder of COOs Initial 4 main objectives for Single IT, text 22 Sept meeting

- " 1. "Make sure it is for key business goals." <- COO,
- " 2. "avoid duplication" <- COO,
- " 3. "not re-inventing the wheel" <- COO
- " 4. "I am interested in the MIS. I'd like some good metrics about what's coming off the 1 billion production line,
- " (are we delivering on time, under budget, are customer satisfied, and are we delivering the value)." <- COO My View

# Draft in Planguage of Objectives

- "Scope: the 1/3 of IT spend for New Demand <- COO
- "Top Objectives for RESULTS Projects
- "
- "
- "

# SPEC TEMPLATE:

<Tag>:

Ambition:

----- *Measurement* -----

Scale:

Past:

Goal:

Meter:

----- *Relationships* -----

Type:

Supports:

Supported By:

----- *Objective Admin* -----

Version:

Owner:

Status:

Scope:

----- *Definitions* -----

# Results MIS:

- " ***Ambition: deliver high-significance real-time metrics, on critical aspects, of project results and resources.***
- " Scale: % of defined [Key Project Data] available to management in real time.
- " Key Project Data: default: {% of Goal Delivered to date, Stakeholder Satisfaction level, Value for Money}
- " Past [Corp., 2007]: 0%
- " Goal [Corp., 2010]: > 90%

## Results MIS:

- " Ambition: ***deliver high-significance real-time metriCorp., on critical aspects, of project results and resources.***
- " ----- Measurement -----
- " Scale: % of defined [Key Project Data] available to management in real time.
- " Key Project Data: default: {% of Goal Delivered to date, Stakeholder Satisfaction level, Value for Money}
- " Past [CORP., 2007]: 0%
- " Goal [CORP., 2010]: > 90%
- " Meter: < manual evaluation of projects not feeding a defined as useful set of data to The Tool, or another useful system for management>.
- " ----- Relationships -----
- " Type: IT COO Level Project Objective
- " Supports:
  - " 1. Portfolio Management Strategic Initiative {Management Framework, Change Drivers, Driving Issues, Results}. Not Quantified.
  - " 2. Business problem statement (PID 2.00. 9 areas. Not Quantified.
  - " 3. High Level Business Requirements: OMSC1 (One IT), OMSC3 (Aligning the Business), OMSC4 (Financial Transparency), OMSC5 (IT Risk Control), OMSC6 (Resource Allocation), OMSC7 (Change Alignment). All quantified!
- " Supported By:
- " ----- Objective Admin -----
- " Version: 23 Sept 2007
- " Sponsor: - CIO
- " Owner: - IT COO
- " Status: draft tg for COO? -> TS
- " Scope: : the 1/3 of IT spend for New Demand <- COO
- " ----- Definitions -----
- " Goal Delivered:
- " defined as: *The Goal refers to a formally defined and approved quantified level of performance that a project is committed to delivering. Goal satisfaction is the primary priority of the project team. The Goal level is needed to enable or drive business performance. 100% of a goal means that the numeric goal is reached measurably in practice. 0% means that no progress from a benchmark level has been made.*
- " Value for Money:
- " defined as:
  - " Project Value is defined as the estimated (or measured) stakeholder consequence from the delivery of the main project objectives. This can be expressed in money terms. It will be for a defined set of assumptions and for a defined time period and scope. Money is the current real cost of getting that Value in place (investment and operational costs).
- " Stakeholder Satisfaction Level:
- " Defined as: a survey set of measures from defined stakeholders about satisfaction with a set of questions about current operational situation, and results of new technology implementation.

Ambition: **Maximize delivery speed, and satisfaction level, of currently prioritized business improvements, for 'key business goals'**

----- Measurement -----

Scale: % of Planned Value actually Delivered to the Business by defined [Time].

Past [Corp., Time = Deadline, 2007]: X% (guess X < 30%??) <- tg

Goal CS, Time = Deadline, 2009: < 50%, maybe much more?

Meter: <The Tool?>

Issue: can The Tool be exploited to track Value?

----- Relationships -----

Type: IT COO Level Project Objective

Supports:

1. Portfolio Management Strategic Initiative {Management Framework, Change Drivers, Driving Issues, Results}. Not Quantified.
2. Business problem statement (PID 2.00. 9 areas. Not Quantified.
3. High Level Business Requirements: OMSC3 (Align Business Needs), OMSC6 (Resource Allocation), OMSC7 (Change Alignment). All quantified!

Supported By: <The Tool>, Planguage, Evo

----- Objective Admin -----

Version: 23 Sept 2007

Sponsor: CIO

Owner: , IT COO

Status: draft tg for COO? -> TS

Scope: : the 1/3 of IT spend for New Demand <- COO

----- Definitions -----

Planned Value:

The monetary benefit estimated for a given scope and duration, that we have formally estimated the organization would get as a result of meeting defined project requirements, at defined levels.

*For example if a project had a requirement to save 1 hour per employee of learning to use a new IT application, and that hour was measurably saved, then the value would be the cost of employee time and overheads saved for a defined period, for a set of employees that needed to learn to use the system. For example for 1,000 employees learning the system in one year, the value would be the cost saving of their 1,000 hours save that year.*

Delivered:

'Delivered' means actually put into place; so that there are no restraints on obtaining the benefits (savings, productivity, and consequent value) that was formally planned in the project.

Business:

'Business' means a real defined set of stakeholders, that we need to give the improved systems to in order to derive benefits and consequent value, when they access or apply the improved system. These stakeholders can be any set of employees, contractors, or customers.

Planguage:

a Corp.Tailored planning language, for projects, that demands formal planning of Planned Value for all critical project performance (Improvement) requirements. *Planguage has been used in Corp.Swiss, and is judged to a be a necessary supplement to Corp.requirements to deal with non-use case requirements.*

Evo:

a project management discipline that focuses on delivering measurable critical requirements and consequent value, to stakeholders, in practice, early and continuously. Evo is about value maximization for the business. The frequent measured delivery of projects Business improvement, can be reported in terms of value delivery. It will keep projects and managers focussed on value delivery to the business.

## **Business Result Alignment: BRA:**



# Business Result Alignment: BRA:

- " ***Ambition: Maximize delivery speed, and satisfaction level, of the Change the Bank Book of Work to achieve 'key business goals'***
- " Scale: % of Planned Value actually Delivered to the Business by defined [Time].
- " Past [Corp., Time = Deadline, 2007]: X% (guess
- "  $X < 30\%??$ ) <- tg
- " Goal [Corp., Time = Deadline, 2009]: < 50%, maybe much more?
- " Issue: can The Tool be exploited to track Value?
- "

# Avoid Duplication:

Ambition: ***eliminate corporate efforts that duplicate other corporate efforts.***

----- *Measurement* -----

Scale: % of project investment that is Duplicated

Past [2007]: > 30%?? Wild guess

Goal [ 2010 ] < 5% hope

Meter: <manual estimate of all projects.>

----- *Relationships* -----

Type: IT COO Level Project Objective

Supports:

1. Portfolio Management Strategic Initiative {Management Framework, Change Drivers, Driving Issues, Results}. Not Quantified.
2. Business problem statement (PID 2.00. 9 areas. Not Quantified.
3. High Level Business Requirements: OMSC1 (One IT), OMSC2 (Top Down), OMSC4 (Common Methods), OMSC6 (Resource Allocation). All quantified!

Supported By: <strategy not identified yet>. <-tg

----- *Objective Admin* -----

Version: 23 Sept 2007

Sponsor: CIO

Owner: -, IT COO

Status: draft tg for COO? -> TS

Scope: : the 1/3 of IT spend for New Demand <- COO

----- *Definitions* -----

Duplicated:

Work that could to a substantial degree (30% or more) be avoided and saved, by making use of another similar effort or investment – is 'duplicated'.

# Avoid Duplication:

- ***"Ambition: eliminate corporate efforts that duplicate other corporate efforts."***
- "Scale: % of project investment that is Duplicated"
- "Past [2007]: > 30%?? Wild guess"
- "Goal [ 2010 ] < 5% hope"
- "

# Exploiting Existing Tools:

- *"Ambition: make use of existing tools, avoid reinventing the wheel.*
- "Scale: % by Total Investment Value that Arguably could be avoided by Profitably making use of Existing Tools
- "Past: 30%±30% ?? wild initial guess to start discussion tg
- "Goal [2012?, Corp. Wide]: ~ 100%
- "

# Exploiting Existing Tools:

**Ambition:** *make use of existing tools, avoid reinventing the wheel.*

----- *Measurement* -----

**Scale:** % by Total Investment Value that Arguably could be avoided by Profitably making use of Existing Tools

**Past:** 30%±30% ?? wild initial guess to start discussion tg

**Goal [2012?, Corp.Wide] :** ~ 100%

**Meter:** <human evaluation of case by case basis, possibly a sample>.

----- *Relationships* -----

**Type:** IT COO Level Project Objective

**Supports:**

1. Portfolio Management Strategic Initiative {Management Framework, Change Drivers, Driving Issues, Results}. Not Quantified.
2. Business problem statement (PID 2.00. 9 areas. Not Quantified.
3. High Level Business Requirements: OMSC4 (Common Financial Mgt Methods). All quantified!

**Supported By:** <strategies not identified yet> <-tg

----- *Objective Admin* -----

**Version:** 23 Sept 2007

**Sponsor:** - CIO

**Owner:** COO, IT COO

**Status:** draft tg for COO? -> CIO

**Scope :** the 1/3 of IT spend for New Demand <- COO

----- *Definitions* -----

**Total Investment Value:**

Entire IT budget, both new investments, and Run the Business costs.

**Arguably:**

A CORP. appointed human expert would argue that the cost could profitably be avoided if we reused some Existing Tool.

**Existing Tools:**

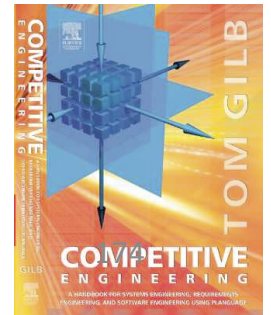
Tools {software, databases, hardware, contracts, development projects, methods, processes, and any other tool} for delivering/operating/maintaining an IT system for the business.

# Some Literature

- "The 'Priority Management' book manuscript, by Tom Gilb: aimed at management Planning —"

[http://www.gilb.com/community/tiki-download\\_file.php?fileId=76](http://www.gilb.com/community/tiki-download_file.php?fileId=76)

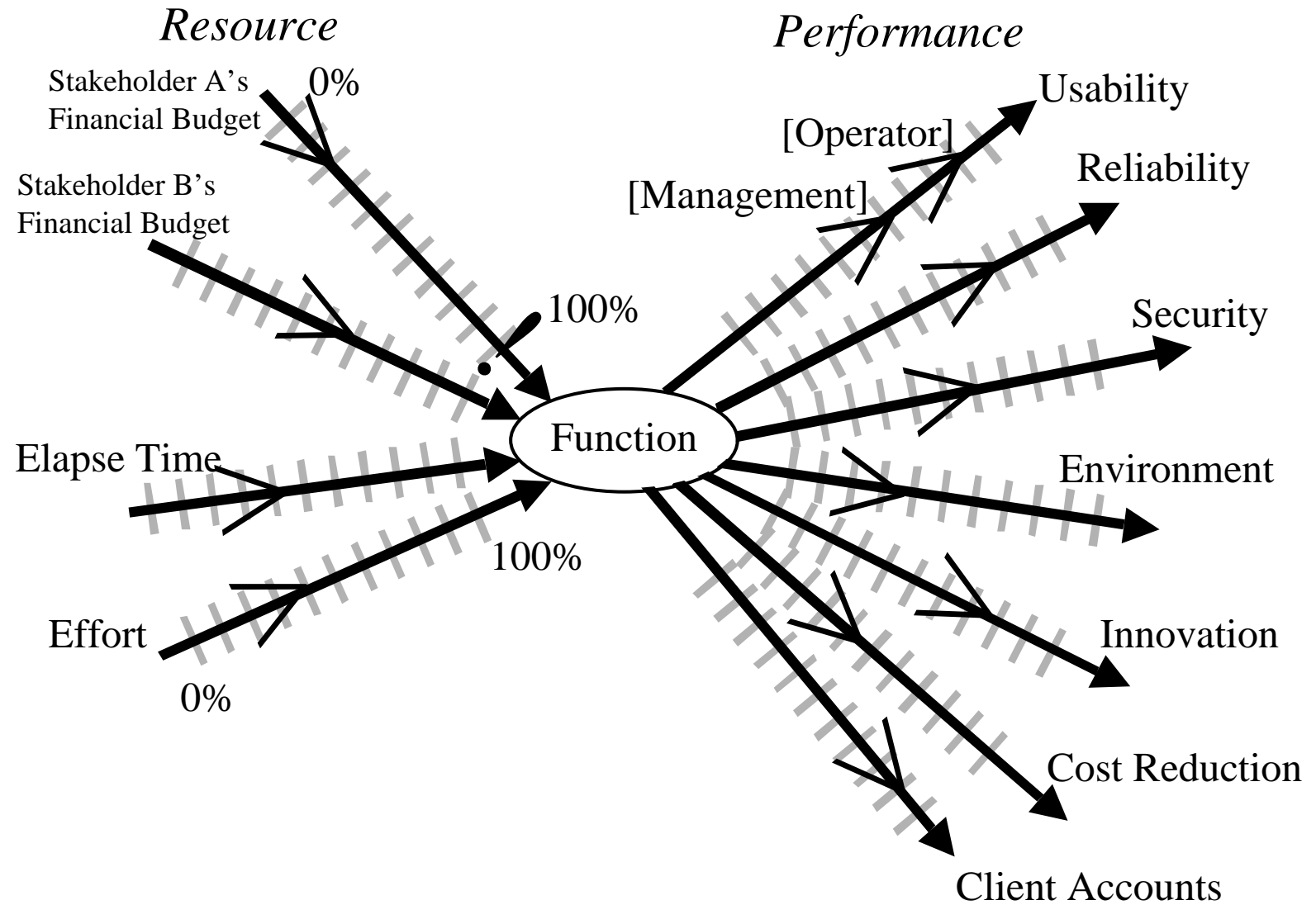
- "Competitive Engineering: the Handbook on the Planguage Method
  - "http://homepage.mac.com/tomgilb/filechute/Gilb Competitive Engineering Book copy.pdf



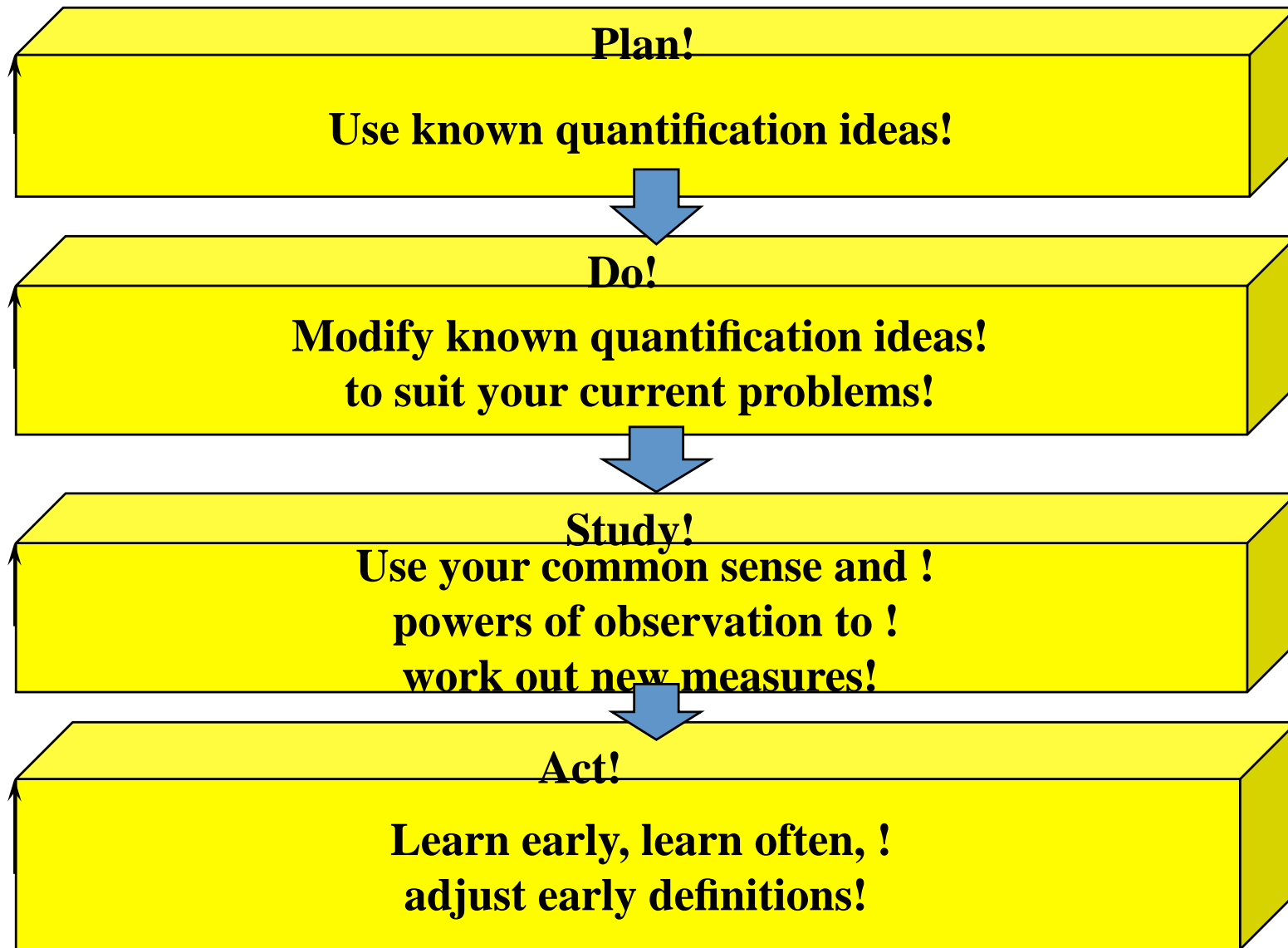


# 7. Generic IT Product Performance (Quality) requirements specification: simple quantification: Usability, Security

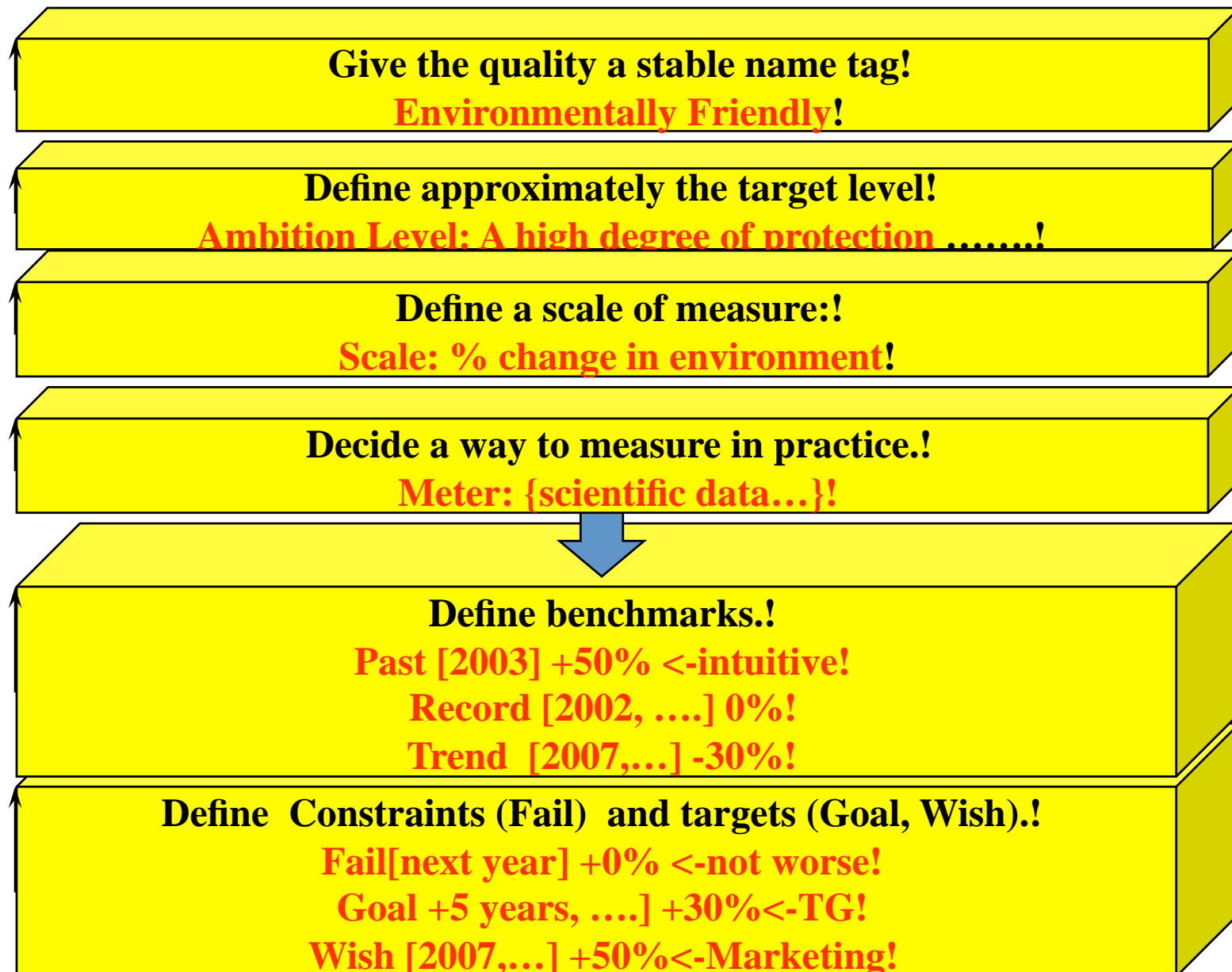
Multiple Required Performance and Cost Attributes  
are the basis for architecture selection and evaluation



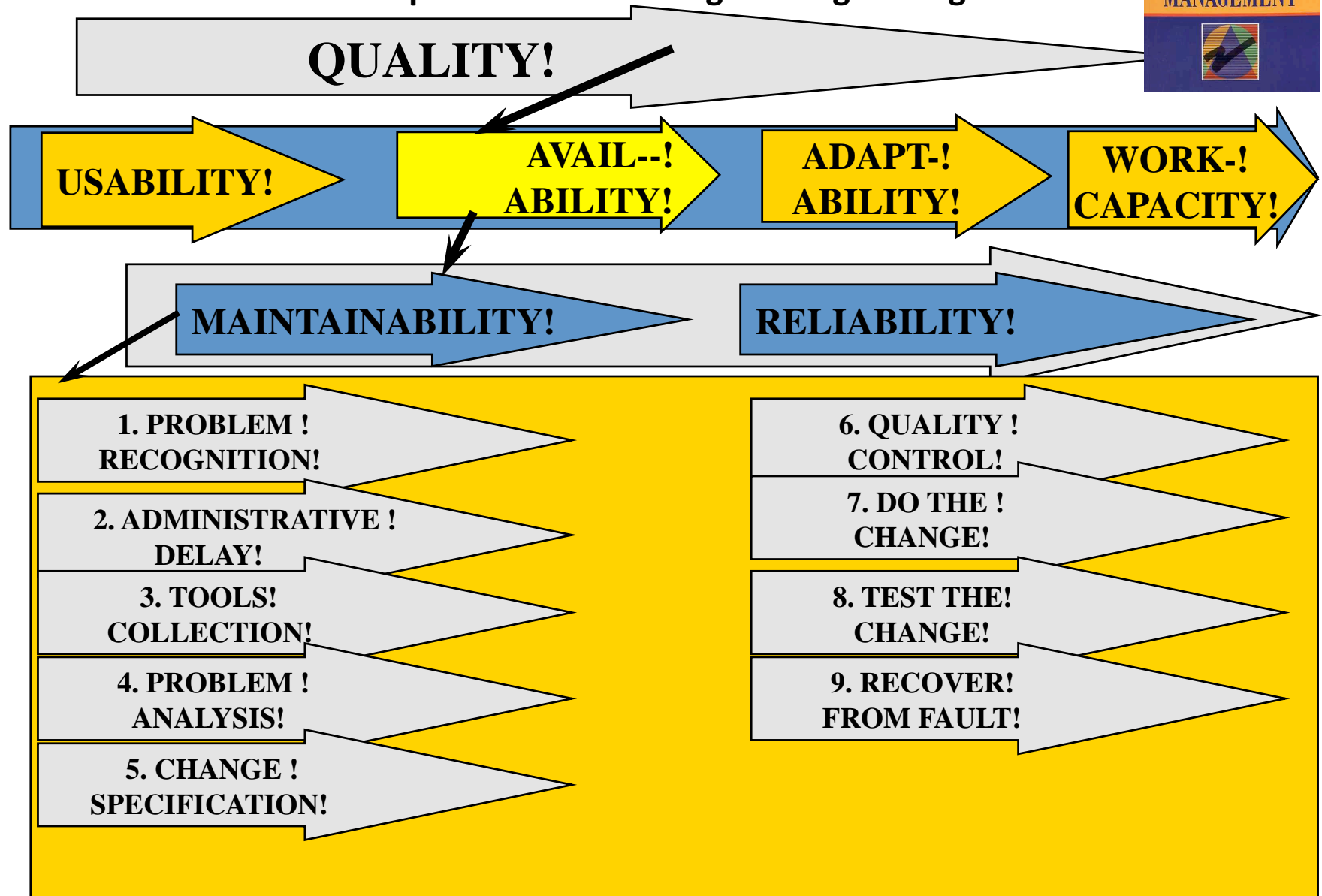
# How to Quantify Quality



## 'Environmentally Friendly' Quantification Example



Devices to help quantify quality ideas:  
Standard Hierarchy of Concepts from  
Gilb: Principles of Software Engineering Management.



# Rewrite of a real Defective 'Requirement at (Norway, 2004 )

- " 1.1.3 MS-Windows concepts
- " The system will make full use of the MS-Windows user-interface concepts such as Wizards to lead the user through user-defined parameters.

'Means'  
not  
'Ends'



False  
Requirement  
(a solution)



September 12, 2009

## Solutions (Designs):

The system will make full use of the MS-Windows user-interface concepts.  
Examples: such as Wizards to lead the user through user-defined parameters.

Why? Lots of users ask for it. (MS-Windows)  
Why? Easy to use. / Intuitive

Usability {intuitiveness, learn, training, mistakes}

## Usability.Intuitive

Ambition: after initial training, (one week course, two week field) the user shall not have to refer to the user manual.

Scale: % of defined [Elements] done Correctly, by defined [User], within <5> seconds.

Correctly: defined as: the System responded in a way the user thought the system should do.

System: Defined as: xxx

Record [ISX Sierra, 1994] 95%±5% <- Boss "as perceived by E"

Record [Product = 408] ??%

Past [Elements = Finding a menu option, User = Beginner, 2004] 40%±20% <- will Tolerable

Goal [Elements = Finding a menu option, User = Beginner, March 15<sup>th</sup> 2007] 70%±10% <- the team

Goal [Elements = Finding a menu option, User = Beginner, March 15<sup>th</sup> 2008, at Commercialization] 90%±5 <- the team

Analysis

The 'Real'  
Requirement  
in Planguage



# Can you Quantify Security?

- " Can you define a Scale of measure for Security?
- "Security:
- " Type: Quality Requirement.
- " Scale: \_\_\_\_\_?
- " Goal [Next Release, Our Software] \_\_\_\_\_ ?

# ISA (Information Security Assurance) security sub-team of IEEE development Standard for Developing Software Life Cycle Processes, P1074 concluded:

- **"Efforts that do not treat security**
  - **"as an integral part of systems engineering**
  - **"and architecture**
  - **"fail to provide security.**
- **"It no longer makes any business sense**
  - **"to spend any money,**
  - **" apply any resources and**
  - **"proceed with any Software Development project**
  - **"unless corporate assets and private customer data will be sufficiently secure."**
  - **"[Barbara Biszick-Lockwood]**
  - **"<http://www.qualityit.net/>**

**Example: “VERY TOP LEVEL PROJECT GOALS**  
**Security Administration Compliance:**

**Security Administration Compliance:**

**Ambition:** *to become compliant and to remain continuously compliant with all current officially binding security administration requirements both from CORP X and Regulatory Authorities.*

**Scope:** *Account Opening and Entitlement Reporting.*

**Scale:** % compliant with CORP X Information Security Standards (CISS) [CORP X Information Security Office (CISO)] on a defined System or Process.

**Note:** *CISS is an officially binding security administration requirement with which we must become compliant.*

# **“VERY TOP LEVEL PROJECT GOALS**

## **Security Administration Compliance:**

### **Security Administration Compliance:**

**Ambition:** to become compliant and to remain continuously compliant with all current officially binding security administration requirements both from CORP X and Regulatory Authorities.

**Scope:** Account Opening and Entitlement Reporting.

**Scale:** % compliant with CORP X Information Security Standards (CISS) [CORP X Information Security Office (CISO)] on a defined System or Process.

*Note: CISS is an officially binding security administration requirement with which we must become compliant.*

### **===== Benchmarks=====**

**Past** [CISS = RSA and IT DIVISION ISAG Compliance Matrix  
[Regional Security Administration and IT DIVISION  
Independent Security Administration Group, October  
2003] **25%** <- JC, Nov-03

*Note: The RSA/IT DIVISION Compliance Matrix originates  
from Otto CXXX and is based on CISS.*

# **“VERY TOP LEVEL PROJECT GOALS**

## **Security Administration Compliance:**

### **Security Administration Compliance:**

**Ambition:** to become compliant and to remain continuously compliant with all current officially binding security administration requirements both from CORP X and Regulatory Authorities.

**Scope:** Account Opening and Entitlement Reporting.

**Scale:** % compliant with CORP X Information Security Standards (CISS) [CORP X Information Security Office (CISO)] on a defined System or Process.

*Note: CISS is an officially binding security administration requirement with which we must become compliant.*

### **===== Targets =====**

**Wish** [Deadline = March 2004, Systems = High Criticality Systems] **100%**

**Wish** [Deadline = June 2004, Systems = {Medium & Low} Criticality Systems] **100%**

*Note: Wishes are stakeholder valued levels that we are not yet sure we can deliver in practice, on time, so we are not promising anything yet, just acknowledging the desire.*

**Goal** [Deadline = March 2004, Systems = High Criticality Systems] 90%±5%

**Goal** [Deadline = June 2004, Systems = {Medium & Low} Criticality Systems] 90%±5%

**Goal** [Midline = February 2004] **50%±10%** “intermediary goal short of 100%”

*Note: Goal levels are what we think we can really promise and focus on. These types of goals push us into thinking about possible Evolutionary result delivery steps.*

**Stretch** [Deadline = March 2004, Systems = High Criticality Systems] 95%±5%

**Stretch** [Deadline = June 2004, Systems = {Medium & Low} Criticality Systems] 95%±5%

*Note: Stretch levels are something that we might be able to achieve if we have sufficient resources, focus and technology available, but we are not sure of that yet. We are NOT promising it now! So this is a way to hold the ideals up in case those things become available.”*

# **“VERY TOP LEVEL PROJECT GOALS**

## **Security Administration Compliance:**

### **Security Administration Compliance:**

**Ambition:** to become compliant and to remain continuously compliant with all current officially binding security administration requirements both from CORP X and Regulatory Authorities.

**Scope:** Account Opening and Entitlement Reporting.

**Scale:** % compliant with CORP X Information Security Standards (CISS) [CORP X Information Security Office (CISO)] on a defined System or Process.

*Note: CISS is an officially binding security administration requirement with which we must become compliant.*

### **===== Targets =====**

**Wish** [Deadline = March 2004, Systems = High Criticality Systems] 100%

**Wish** [Deadline = June 2004, Systems = {Medium & Low} Criticality Systems] 100%

*Note: Wishes are stakeholder valued levels that we are not yet sure we can deliver in practice, on time, so we are not promising anything yet, just acknowledging the desire.*

**Goal** [Deadline = March 2004, Systems = High Criticality Systems] 90%±5%

**Goal** [Deadline = June 2004, Systems = {Medium & Low} Criticality Systems] 90%±5%

**Goal** [Midline = February 2004] **50%±10%** “intermediary goal short of 100%”

*Note: Goal levels are what we think we can really promise and focus on. These types of goals push us into thinking about possible Evolutionary result delivery steps.*

**Stretch** [Deadline = March 2004, Systems = High Criticality Systems] 95%±5%

**Stretch** [Deadline = June 2004, Systems = {Medium & Low} Criticality Systems] 95%±5%

*Note: Stretch levels are something that we might be able to achieve if we have sufficient resources, focus and technology available, but we are not sure of that yet. We are NOT promising it now! So this is a way to hold the ideals up in case those things become available.”*



# “VERY TOP LEVEL PROJECT GOALS

## Security Administration Compliance:

### Security Administration Compliance:

**Ambition:** to become compliant and to remain continuously compliant with all current officially binding security administration requirements both from CORP X and Regulatory Authorities.

**Scope:** Account Opening and Entitlement Reporting.

**Scale:** % compliant with CORP X Information Security Standards (CISS) [CORP X Information Security Office (CISO)] on a defined System or Process.

*Note: CISS is an officially binding security administration requirement with which we must become compliant.*

### ===== Targets =====

**Wish** [Deadline = March 2004, Systems = High Criticality Systems] 100%

**Wish** [Deadline = June 2004, Systems = {Medium & Low} Criticality Systems] 100%

*Note: Wishes are stakeholder valued levels that we are not yet sure we can deliver in practice, on time, so we are not promising anything yet, just acknowledging the desire.*

**Goal** [Deadline = March 2004, Systems = High Criticality Systems] 90%±5%

**Goal** [Deadline = June 2004, Systems = {Medium & Low} Criticality Systems] 90%±5%

**Goal** [Midline = February 2004] **50%±10%** “intermediary goal short of 100%”

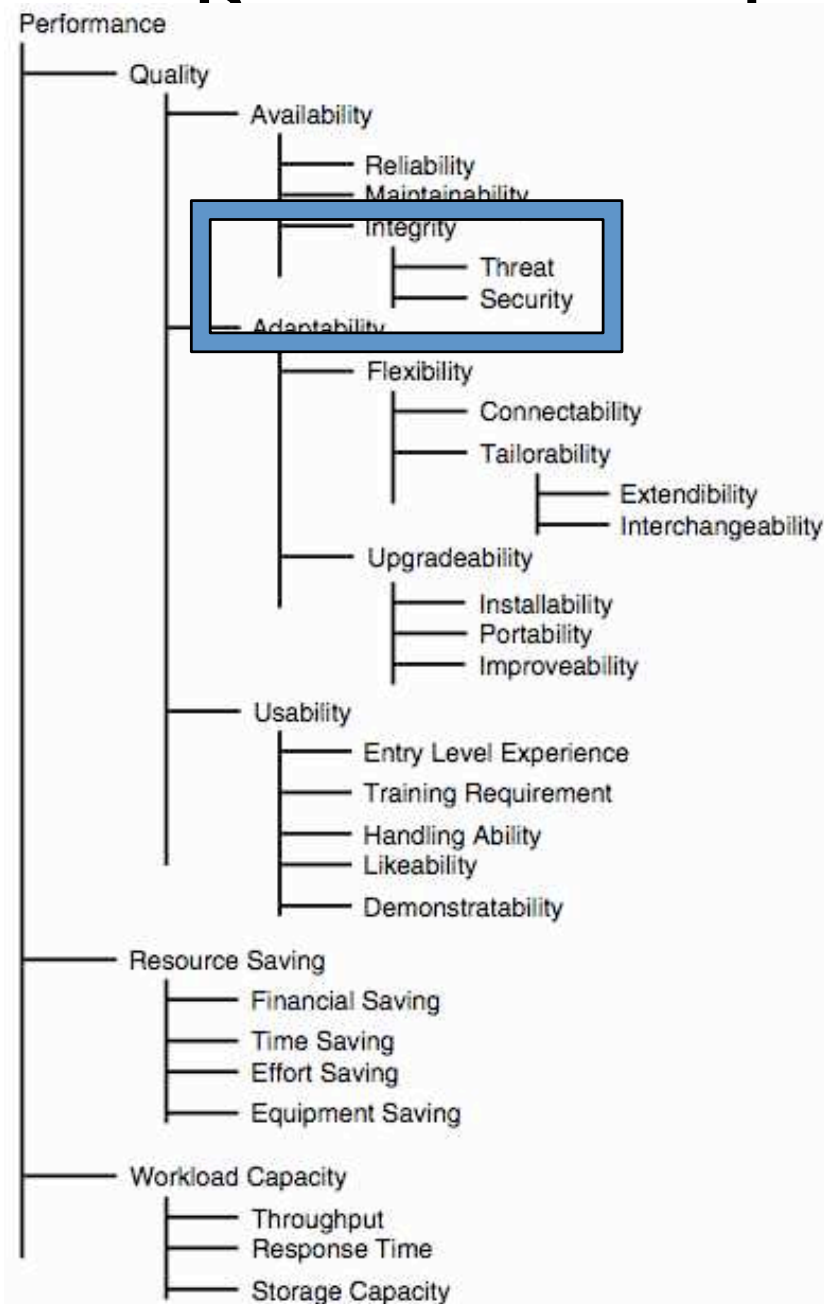
*Note: Goal levels are what we think we can really promise and focus on. These types of goals push us into thinking about possible Evolutionary result delivery steps.*

**Stretch** [Deadline = March 2004, Systems = High Criticality Systems] 95%±5%

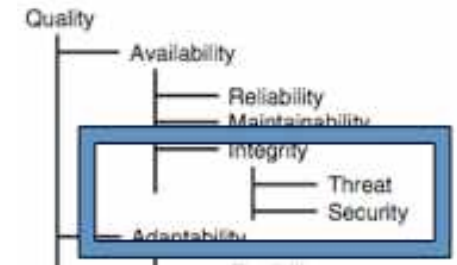
**Stretch** [Deadline = June 2004, Systems = {Medium & Low} Criticality Systems]  
95%±5%

*Note: Stretch levels are something that we might be able to achieve if we have sufficient resources, focus and technology available, but we are not sure of that yet. We are NOT promising it now! So this is a way to hold the ideals up in case those things become available.”*

# Security in Performance

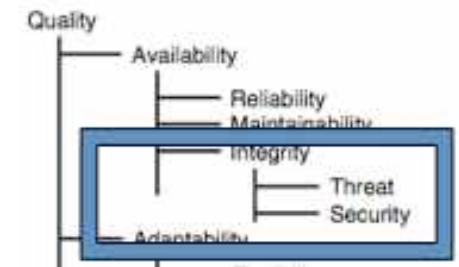


- A generic model of security  
(Integrity, Security and Attack)  
in the form of a Planguage specification.



- **Integrity:** *'The ability of the system to survive attack'*
- **Gist:** Integrity is a measure of the confidence that the system has suffered no harm: its security has not been breached and, its use has resulted in no 'corruption' or impairment to it.
- *Note: An attack on the Integrity of a system can be accidental or intentional.*
- *Note: The Integrity of a system depends on the frequency of threat to it and the effectiveness of its security.*
- Type: Elementary Quality Requirement.
- Scale: Probability for a defined [System] to achieve defined [Coping Action] when confronted with a defined [Attack] using defined [Security] measures, under defined [Conditions].
- Coping Action: defined as: {Detect, Prevent, Capture, Thwart, Recover}.
- *Note: here is an example of specifying a requirement using the defined scale above.*
- Goal [System = Our Product, Coping Action = Detect Attack, Attack = In House Amateur Hacker, Security = Microsoft Package, Conditions = Firewall Breached] 99%.

- A generic mode (**Pattern**) of security (Integrity, Security and Attack) in the form of a Language specification.



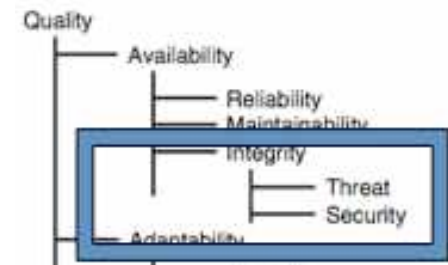
- " **Integrity:** *'The ability of the system to survive attack'*
- " Gist: Integrity is a measure of the confidence that the system has suffered no harm: its security has not been breached and, its use has resulted in no 'corruption' or impairment to it.
- " *Note: An attack on the Integrity of a system can be accidental or intentional.*
- " *Note: The Integrity of a system depends on the frequency of threat to it and the effectiveness of its security.*
- " Type: Elementary Quality Requirement.
- " **Scale: Probability for a defined [System] to achieve defined [Coping Action] when confronted with a defined [Attack] using defined [Security] measures, under defined [Conditions].**
- " **Coping Action:** defined as: {Detect, Prevent, Capture, Thwart, Recover}.
- " *Note: here is an example of specifying a requirement using the defined scale above.*
- " **Goal** [System = Our Product, Coping Action = Detect Attack, Attack = In House Amateur Hacker, Security = Microsoft Package, Conditions = Firewall Breached] 99%.

The Integrity formula: if you know or assume 2 factors, you can calculate the third!

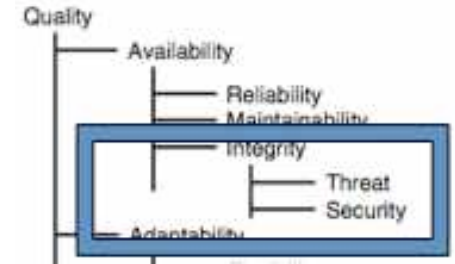
**Integrity =**

**Sum of all instances of  $[1 - \text{Threat} \times (1 - \text{Security})]$ .**

- " Or more simply:
- " **The Integrity level of a system**
  - "depends on the degree of threat
  - "and the security design's ability
    - " to cope with that class of threat.



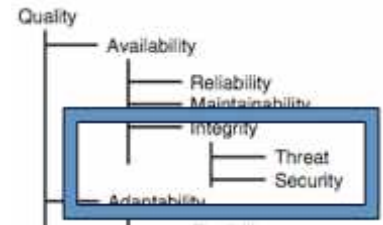
So, for example,



- "if planned Integrity is maximum one failure per time period,
  - "and there are 100 expected or assumed attacks on the system in a given timeframe,
  - "then the effectiveness of the security device must be at least 99%.



# Here is an example



## Integrity:

Type: Elementary Quality Requirement.

Scale: **Probability for a defined [System] to achieve defined [Coping Action] when confronted with a defined [Attack] using defined [Security] measures, under defined [Conditions].**

**Meter:** test one or more Security measure designs for all defined Coping Actions, and all defined Attack(s), under all defined Conditions.

**Goal** [System = Survey Database using Conformat software,  
Coping Action = Detect,

Attack = Professional Top Class Hacker, Security = Complete Security Architecture [Version 1.0],

Conditions = {No Advance Warning, Inside Mainframe Building, All Electronic Specs Available to Hacker}] **50%**

# Another example

**Security:**

**Stakeholders:** NSM

**Scale:** % probability the a defined [Assailant] does NOT succeed in a defined [Compromise] for defined [Data] under defined [Conditions].

**Meter** [for Supplier of Security System payment] Use a professional Norwegian hacker. Give them up to 100 break-in attempts.

*Note [Meter] If 1 or more of these is successful, then payment is not due the security suppliers, since the assumption is that it cannot be a better than 99.00% system. If great accuracy is desired increase number of hacks, and make sure they are representative of the best, by using at least 10 per 1000 attempts by professional hackers.*

**Goal** [Assailant = Professional Norwegian Hacker, Compromise = Detailed Knowledge, Data = Norwegian Government Budget, Conditions = Before Secrecy Lifted] 99.90 %

## Example: with 'Relationships' background specified

### **Integrity:**

Type: Elementary Quality Requirement.

Scale: Probability for a .... *as above examples in detail*

Goal [... *as above examples in detail*] 50% <- TG

Source: NASA Security Procedures 2004

**Rationale:** Deterrence of Professional Hackers

**Authority:** Congressional Budget for NASA

### **Issues:**

I1: will the guideline level change in this years unpublished budget?

I2: does this impact NASA business outside the USA?

### **Dependencies**

D1: Federal Penalties for Hacking.

### **Risks**

R1: the proposed security technology does not work at the levels estimated

R2: improved hacking paradigms, beyond currently know state of the art.

# Various Numeric level Specifications

## Integrity:

Type: Elementary Quality Requirement.

Scale: Probability for a .... *as above example in detail*

Meter: test one or .... *as above example in detail*

**Benchmarks** ----- reference levels

Past [2004, .....] : 15%

Record [Lab Tests]: 99%

Trend [Next Year]: 60% +

**Constraints** ----- minimum levels

Fail 30%

Survival 20%

**Targets** ----- levels to aim at

Wish 80% +

*Goal [... as above example in detail]* 50%

*Stretch* 55%

**Impacts** ----(if we reach the Goal level, what happens?)

Primary Impact: Legal Certification

Secondary Impact: Insurance Costs

# Impact Estimation Table for Security (Real Example)

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

# 8. Application Domain requirement Tailoring.

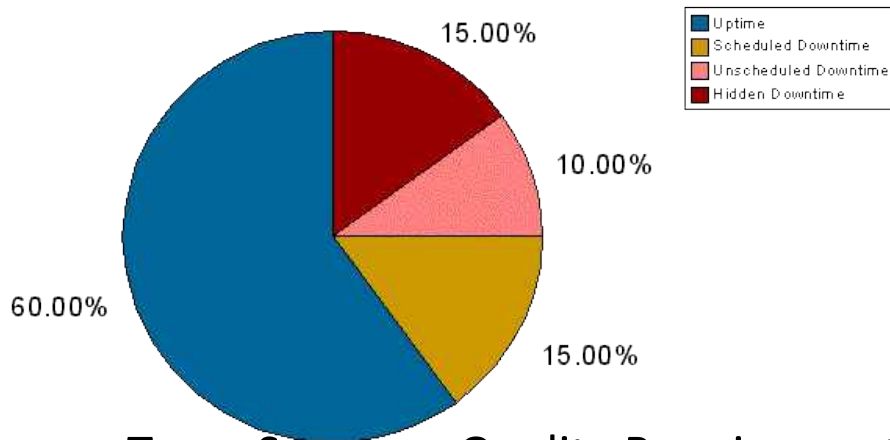
# Rock Solid Robustness: *many splendored*

- " **Type:** *Complex* Product Quality Requirement.
- " **Includes:**
  - " { *Software Downtime,*
  - " *Restore Speed,*
  - " *Testability,*
  - " *Fault Prevention Capability,*
  - " *Fault Isolation Capability,*
  - " *Fault Analysis Capability,*
  - " *Hardware Debugging Capability* }.



• "





# Software Downtime:

**Type:** Software Quality Requirement. **Version:** 25 October 2007.

**Part of:** Rock Solid Robustness.

**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:

**Type:** Software Quality Requirement. **Version:** 25 October 2007.

**Part of:** Rock Solid Robustness

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

3

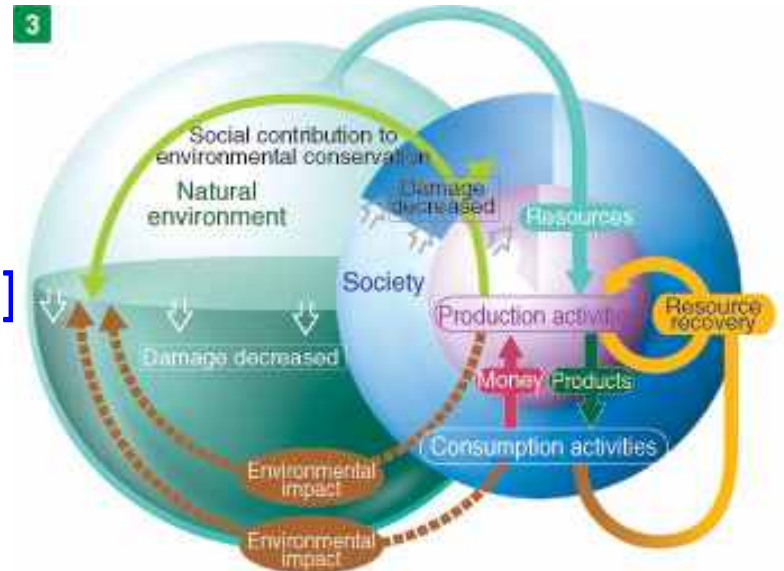
**Scale:** Duration from Initiation of Restore to Complete and verified state of a defined [Previous: Default = Immediately Previous] saved state.

**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.



# Testability:

**Type:** Software Quality Requirement.

**Part of:** Rock Solid Robustness

**Initial Version:** 20 Oct 2006

**Version:** 25 October 2007.

**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.

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



# Example: Operator Usability

## 4.2. Usability

### 4.2.1. Learn-ability

### 4.2.2. Like-ability

### 4.2.3. User Productivity

<b>ID</b>	7	<b>Title</b>	Faster spread layout handling			
<b>Priority</b>	1	<b>Status</b>	Open	<b>Version</b>	0.5	
<b>Category</b>	Usability/User Productivity		<b>Type</b>	Quality Requirement		
<b>Date submitted</b>	28.09.2004		<b>Last Update</b>	3 Feb 2005		
<b>Reporter</b>	S		<b>Assigned to</b>			
<b>Stakeholders</b>						
<b>Ambition</b>	Reduce operator time by at least factor 2, when laying out the spread: cables and connection					
<b>Justification</b>	Business Economics, specifically <Operational Cost, system efficiency>					
<b>Scale</b>	<b>Average Time for defined [Crews {Layout Crew, Pickup Crew}] of defined [Crew Size] with a defined [Spread Configuration] per [1,000-Sensors], to successfully complete defined [Layout Work {Initial Layout, Layout Rolling}].</b>					
<b>Meter</b>	Real field trial and operational data manually collected					
<b>Goal</b>	[1 <sup>st</sup> Release, Layout Crew, 5,000 Sensors, Desert, Crew Size = 10, Initial Layout] X/2 hour?					
<b>Past</b>	[2004, Layout Crew, 5,000 Sensors, Desert, Crew Size = 10] X hour?					
<b>Links</b>	req 2.5.3					

# Example: Crew Usability

ID	8	Title Reduced battery handling				
Priority		1	Status	Open	Version	0.5
Category		Usability/User Productivity		Type	Quality Requirement	
Date submitted		28.09.2004		Last Update	3 Feb 2005	
Reporter		S		Assigned to		
Stakeholders		Battery Handling Crew				
Ambition		reduce battery charging and replacement effort				
Comment		Assumption: The number of batteries will be reduced by reducing the power consumption per channel (This is a solution <-BN)				
Scale		Effort-hours per day for Battery Handling {Charging and Replacement}.				
Meter		Manual logs observing real operations.				
Goal		[]X/2?				
Past		[ ] X				
Links		req 2.5.4, supported by requirement 25Battery Power Consumption				



ID	20	Title				System Overhead Time: Note, name title needs reworking to reflect content) <- BN	
Priority	1	Status	Open	Version	0.51		
Category	Availability/Recoverability		Type	Quality Requirement			
Date submitted	28.09.2004		Last Update	3.2.2005			
Reporter			Assigned to	J			
Stakeholders	Field Operations (all levels).						
Ambition	"The system must be capable of passing uninterrupted seismic data from the full channel count (100,000 minimum live channels), plus any display information required, control information flow, QC information required, plus routing all data from any single broken link <b>without significant time overhead</b> " <- Stuart Papworth						
Comment							
Scale	<p><b>Time in seconds from when a Single Failure occurs, until Full Recovery achieved.</b></p> <p>Single Failure: defined as: broken link, or broken transport network node,</p> <p>Full Recovery: defined as: system is Operational again, and no data is lost.</p> <p>Operational: defined as: The network integrity and bandwidth is restored.</p> <p><i>Note 1: this includes the time to pass uninterrupted seismic data from the full channel count (100,000 minimum live channels), plus any display information required, control information flow, QC information required, plus routing all data from any single broken link.</i></p> <p><i>Note 2: exceptions, short circuit? – cost implications, under investigation. &lt;- Thorleiv</i></p>						
Meter	<p><b>Gist:</b> Measure from &lt;Single Failure occurred&gt; to &lt;Full Recovery&gt;.</p> <p><b>Description:</b> A set of artificial Single Failures is injected as a test, and time is measured until Full Recovery, using built in measure.</p> <p><b>Issue:</b> is this already built in or do we have to plan a design to build it in – the seconds measure to recovery.</p>						
Goal	[First Version] < 0.5 seconds ?? <- J He says 'closer to 10 seconds'						
Past	About 10 to 60 minutes?? "The old system does not have rapid automatic recovery. Manual fix". <-BN						
Links	req 5.3						

Scale Detail  
on next slide

real case

# Detail of Scale for 'System Overhead Time' requirement

<b>Scale</b>	<p><b>Time in seconds from when a Single Failure occurs, until Full Recovery achieved.</b></p> <p>Single Failure: defined as: broken link, or broken transport network node,</p> <p>Full Recovery: defined as: system is Operational again, and no data is lost.</p> <p>Operational: defined as: The network integrity and bandwidth is restored.</p> <p><i>Note 1: this includes the time to pass uninterrupted seismic data from the full channel count (100,000 minimum live channels), plus any display information required, control information flow, QC information required, plus routing all data from any single broken link.</i></p> <p><i>Note 2: exceptions, short circuit? – <u>cost implications</u>, under investigation. &lt;- <u>T</u></i></p>
--------------	--



Priority	1	Status	Open	Version	0.5
Category	Availability Recoverability		Type	Quality requirement	
Date submitted	3.2.2005		Last Update	3.Feb.2005	
Reporter	Bj		Assigned to	yyy	
Stakeholders	Field Operations				
Ambition	Substantial reduction in component recovery speed				
Scale	<b>Mean time in minutes to recover a defined [Sub-System] from a Failed State until the Sub-system is in a defined [State]: default Locally Fixed.</b>  State: {Failed, Locally Fixed, Repositioned}.				
Meter	Manual calculation from Introspection statistics				
Goal	[Whole System] 30 minutes? <- BN [Sub-system = Central System Software, 1 <sup>st</sup> Release] 5 minutes? <- BN [Central System Hardware, 1 <sup>st</sup> Release] 10 min.? <-BN [Sensor Network] 60 mins. ? [Transport Network] 60 mins. ? [Operators] 10 mins. ? [Power Supply] ? [All Other Components] ? <what else is there? Trucks?, Air Conditioning>				
Past	[Whole System] [Central System Software, 2004] 1? <- 2004 field observation? [Central System Hardware, 2004] ? [Sensor Network] ? [Transport Network] ? [Operators] ? [Power Supply] ? [All Other Components] <what else is there? Trucks?, Air Conditioning>				
Justification	Business productivity				
Definitions	Whole System: defined as: {Central Software System, Central hardware System, Sensor Network, Transport network, Operators, Power Supply, All Other Components}.				

September 12, 2009

real case

# Quality Requirement Recoverability

- " Notice:
  - " multiple Goal Levels
  - " Parameterized Scale

## 4.1.1. Readiness

real case


<b>ID</b>	21	<b>Title</b>	System boot time			
<b>Priority</b>	1	<b>Status</b>	Open	<b>Version</b>	0.5	
<b>Supports</b>	Availability/Readiness		<b>Type</b>	Quality		
<b>Date submitted</b>	28.09.2004		<b>Last Update</b>	3.2.2005		
<b>Reporter</b>	S.....		<b>Assigned to</b>			
<b>Stakeholders</b>	Field Operations					
<b>Ambition</b>	Substantially reduce the time from power is turned on, until ready for acquisition.					
<b>Justification</b>	More productive earning time. <refer to a higher level business objective>					
<b>Scale</b>	<b>Maximum time from power is turned on to Ready For Acquisition.</b>  Ready For Acquisition: defined as: the system is completely ready to record data. The Master Display is fully on screen including GIS View Map, with Status information for all sensors and boxes.  Assumption: the time to lay out the Spread is independent of this, and presumed completed by power on.					
<b>Meter</b>	Manual test and stopwatch recording.					
<b>Goal</b>	Goal1: [Spread] 3 minutes. Goal2: [Central System] 10 minutes					
<b>Past</b>	Crew2, 2004] ~30 min. ?? <-BN					
<b>Links</b>	req ??					

## Business Objective

TTM

Same Format

**2.1. Time to market**

ID	1	Title	Time to market			
Priority	1	Status	Open	Version	0.5	
Category	Time to market		Type	Business requirement		
Date submitted	28.09.2004		Last Update	28.09.2004		
Reporter	S: 		Assigned to			
Stakeholders						
Description	It is expected that an average of 2 QX crews will be manufactured and deployed per year after 2007					
Scale	Point in time successful delivery to first customer					
Meter						
Goal	Goal1 [Q1 2007] 30000 live channel system earning revenue Goal2 [July 2007] 45000 live channel system earning revenue					
Past						
Links	req 2.7					



# Template for Quality Requirements

## Template for Quality Requirements:

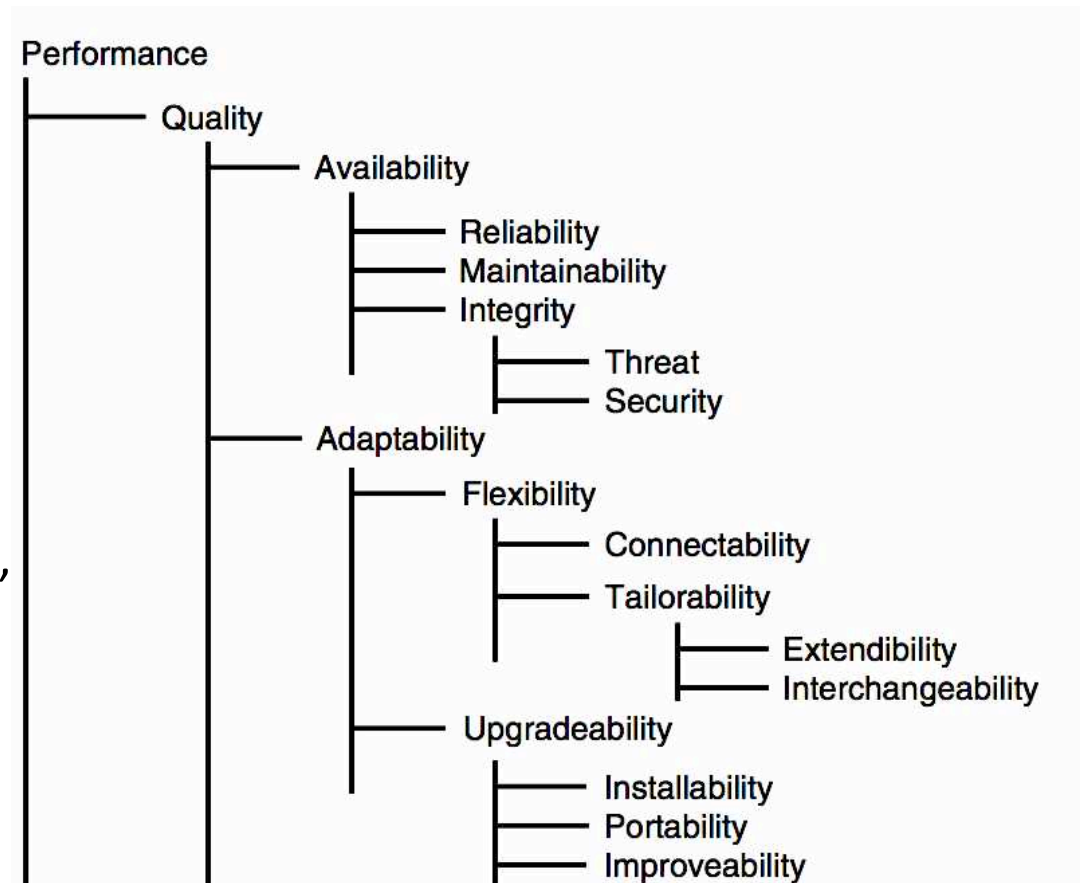
ID	?	Title				
Priority	?	Status	Open	Version	0.5	
Category				Type	Quality Requirement	
Date submitted	x.x.2005			Last Update	X.X.2005	
Reporter	xxx			Assigned to	yyy	
Scope	<define what this applies to of operations or system components>					
Stakeholders	Zz, xx					
Ambition						
Scale						
Meter						
Goal						
Past						
Justification	<link to business requirements>					
Links						

Developed by BN

# 9. Internal Stakeholder requirements quantification (maintainability, testability, reliability)

# Broader Maintainability Concepts

- " Maintainability in the strict engineering sense is usually taken to mean **bug fixing**.
- " I have however been using it *thus far* to describe ***any software change activity or process***.
- " We could perhaps better call it **'software change ability'**.
- " Different classes of change, will have different requirements related to them,
  - " and consequently **different technical solutions**.
- " It is important that we be very clear
  - " in setting requirements,
  - " and doing corresponding design,
  - " exactly what **types of change** we are talking about.



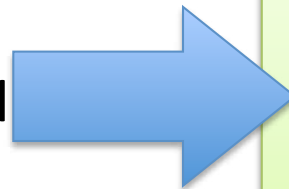
•"

# General 'Change Attribute' Tailoring

- " The following slides will give a **general set of patterns** for
  - " defining and distinguishing *different classes* of 'maintenance'.
- " But in your *real* world, you will want to **tailor** the definitions to *your* domain.
  - " You can initially tailor using the '**Scale**' of measure definition.
  - " And continued tailoring can be done by defining **[conditions]** in the requirement level qual



**Scale:**  
**% of transactions  
successfully completed  
by defined [Person]  
doing defined [Task].**



**Goal [Task = Update,  
Person = New Hire,  
Deadline = Phase 3]  
60%**



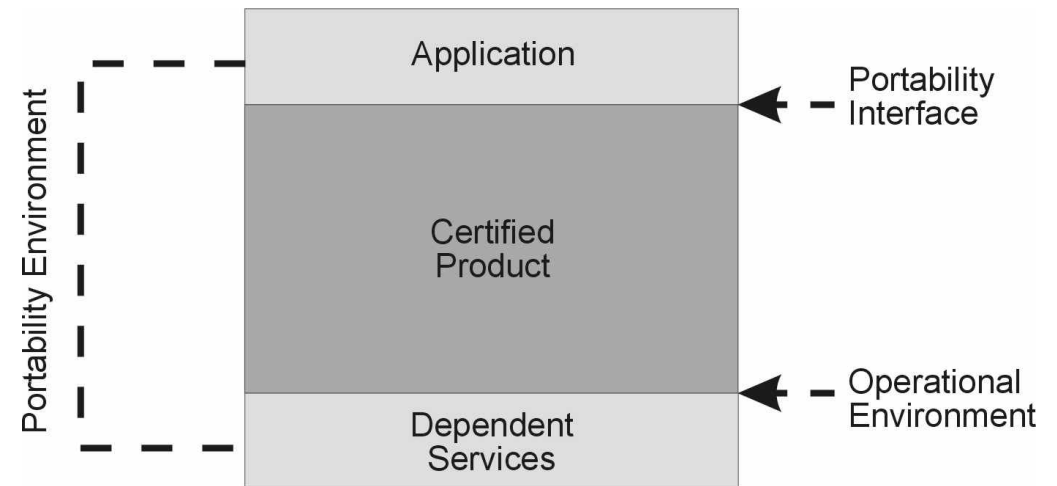
*A generic set of performance measures, including several related to change.*

**For example:**

**Code Portability:**

**Scale:**

**Effort in Hours  
needed to Port  
each 1000 Non-Commentary Lines of Code  
from a defined [Home Environment]  
to a defined [Target Environment],  
using defined [Tools]  
and defined [Personnel].**



**Goal**

**[Home Environment = {.net, Oracle,} ,**

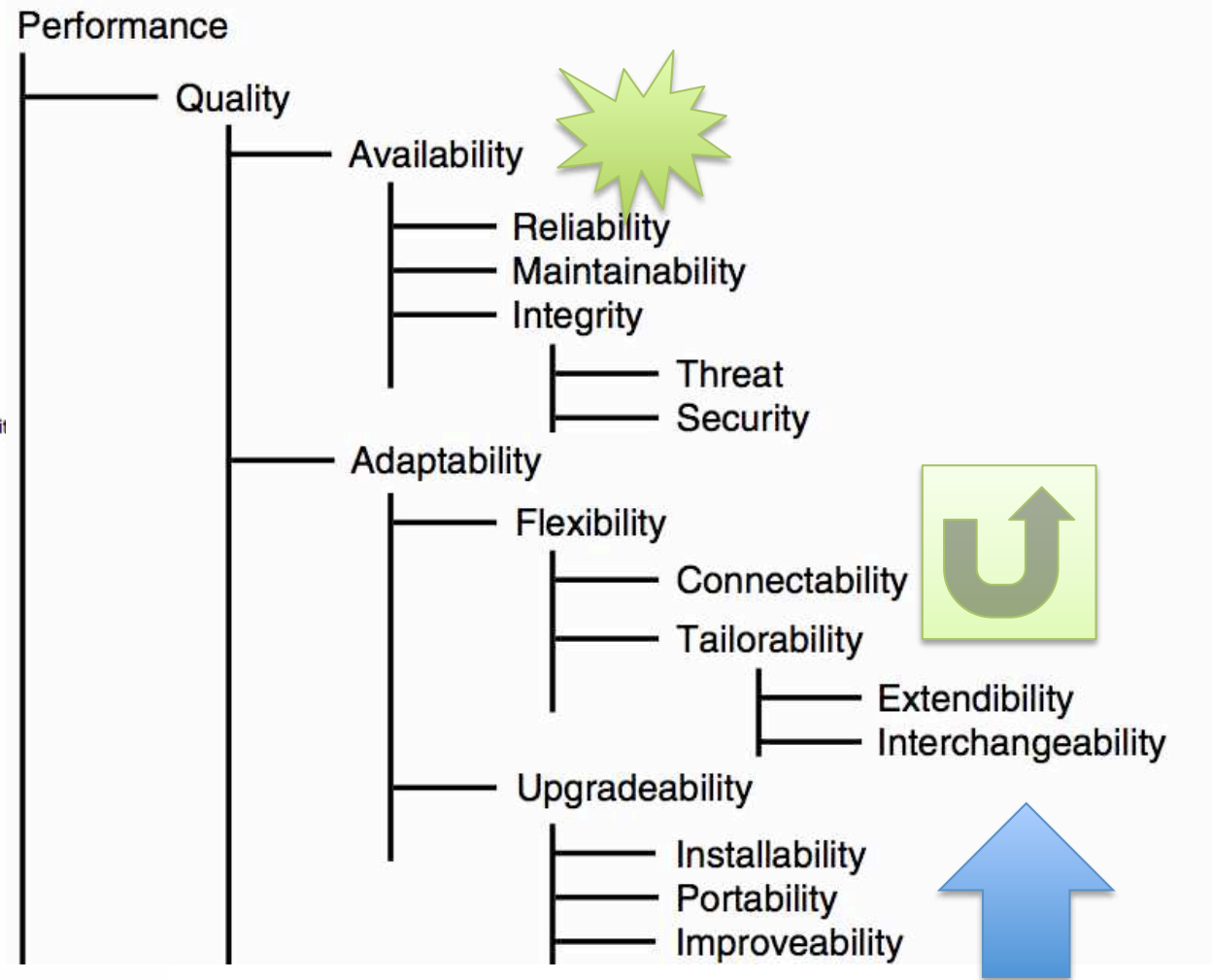
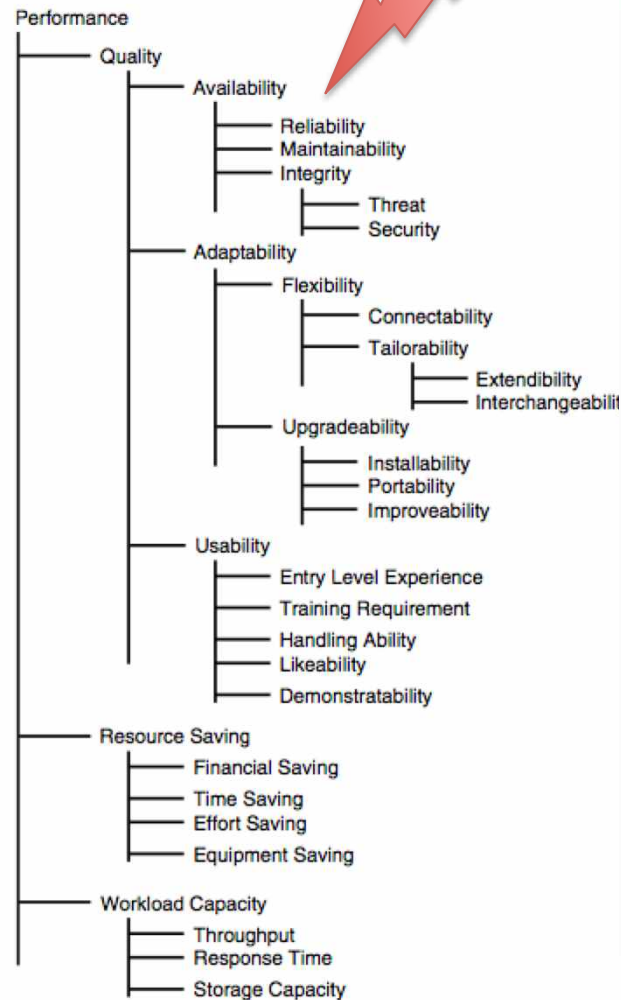
**Target Environment = {Java++, Open Source, Linux},**

**Tools = Convert Open ,**

**Personnel = {Experienced Experts, India}]      60 hours.**

# A Generic Set of Performance measures – including several related to ‘change’

154 Competitive Engineering

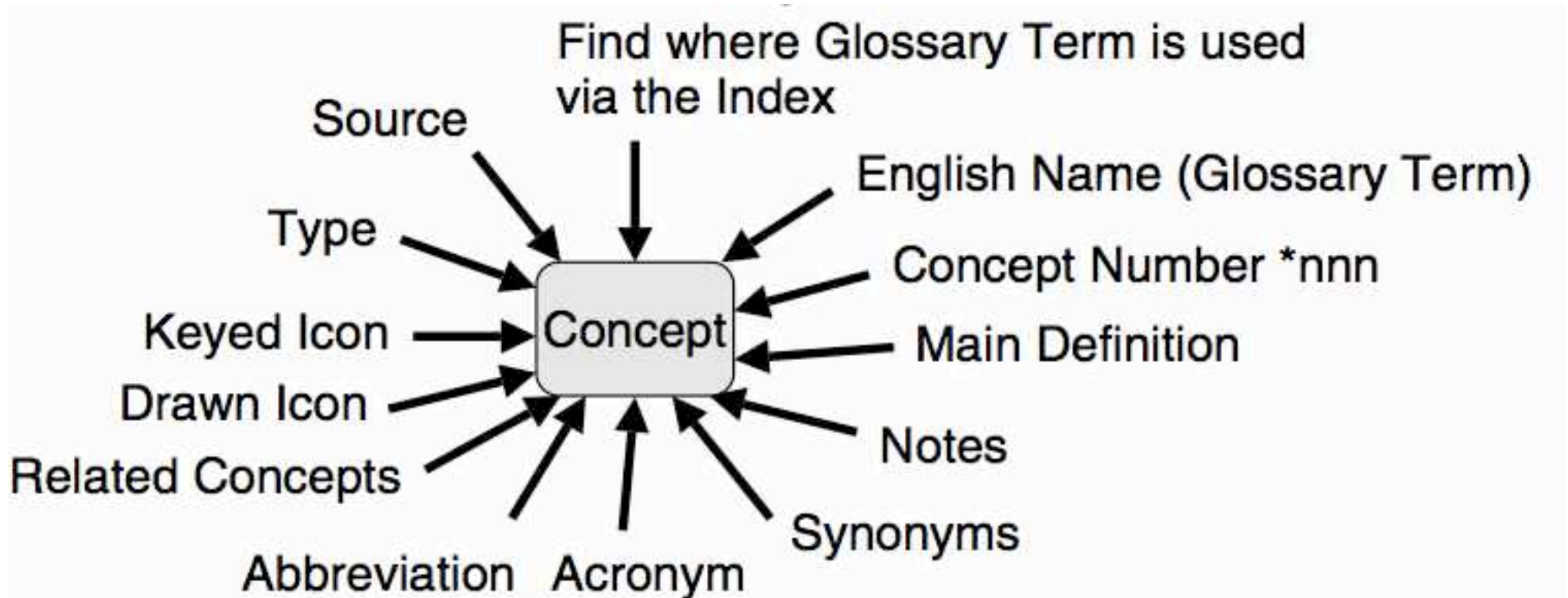


**Figure 5.3**

One decomposition possibility for performance attributes with emphasis on the detail of the quality attributes.

The *attribute names* used are arbitrary choices by the author.

- " They only start to take on meaning when defined,
  - " with a Scale of measure.
- " There are no accepted or acceptable standards here,
  - " and certainly not for software.
  - " Even in hardware engineering, there is an accepted pattern – such as “Scale: Mean Time to Repair”.
  - " But it is accepted that we have to *further* define such concepts *locally*,
    - " such as the meaning of ‘Repair’.



# Maintainability Measures

- "Here are some of the general **patterns** we can use to define and distinguish the different classes of change processes on software.
- "First the 'Bug Fixing' pattern (from which we derived the example at the beginning of this talk).



*Maintainability  
components,  
derived from a  
hardware  
engineering view,  
adopted for software.*

**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).



September 12, 2009

SPA BGS 2 Sept 2009 London

Notice that *Maintainability* in the narrow sense  
(fix bugs)  
is quite separate from other 'Adaptability' concepts.

- " This is normal *engineering*,
  - " Which places fault repair together with reliability and availability;
  - " Those 3 determine the *immediate* operational characteristics of the system.
- " The other forms of adaptability are more about *potential* future upgrades to the system,
  - " *change*, rather than *repair*.
- " Change and repair, have in common that
  - " our system *architecture* has to make it easy to change, analyze and test.
- " The system *itself* is unaware of
  - " whether we are *correcting a fault*
  - " or *improving* the system.
- " The consequence is that
  - " *much of the maintenance-impacting 'design' or 'architecture'*
  - " **benefits**
  - " *most of the types of maintenance (fix and adapt).*





# Here are a *generic* set of definitions for the '*Adaptability*' concepts.

**Adaptability**: 'The **efficiency** with which a system can be changed.'

**Gist**: Adaptability is a measure of a system's ability to change.

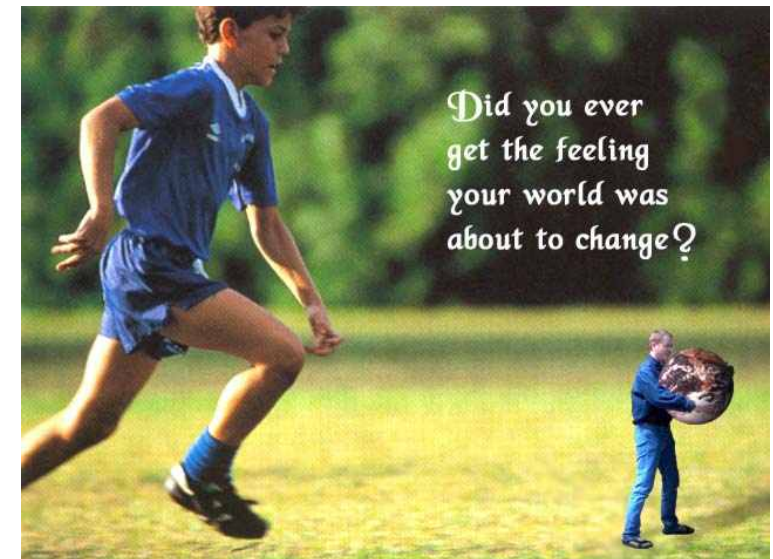
**Includes**: { a set of scalar variables, such as Portability}.

Note: probably not simple enough to define with a **single** Scale.

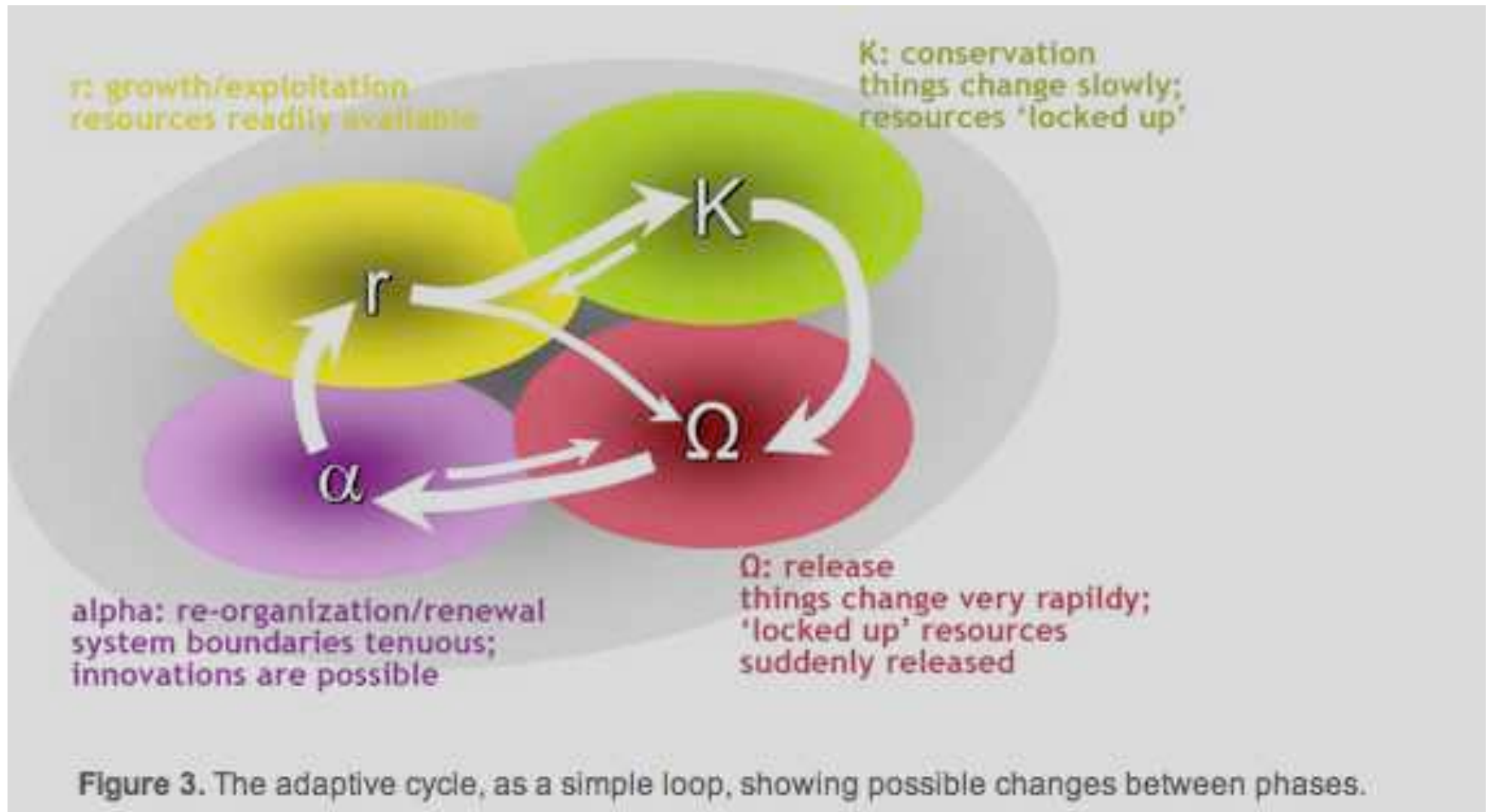
**Type**: Complex Quality Attribute.

Since,

- " if given sufficient resource, a system can be changed in
  - " almost any way,
- " the primary concern is with the amount of
  - " resources
    - " (such as time, people, tools and finance)
- " needed to bring about specific changes
  - " (the change 'cost').



# The Adaptive Cycle



<http://www.resalliance.org/564.php>

**Adaptability:**  
Viewed as  
**Elementary** or **Complex** concept..

**Adaptability:**

**Type:** Elementary Quality Requirement.

**Scale:** Time needed to adapt a defined [System]  
from a defined [**Initial State**] to another defined  
[**Final State**] using defined [**Means**].



© Alistair Boddy-Evans 2002

**Adaptability:**

**Type:** Complex Quality Requirement.

**Includes:** {*Flexibility, Upgradeability*}.

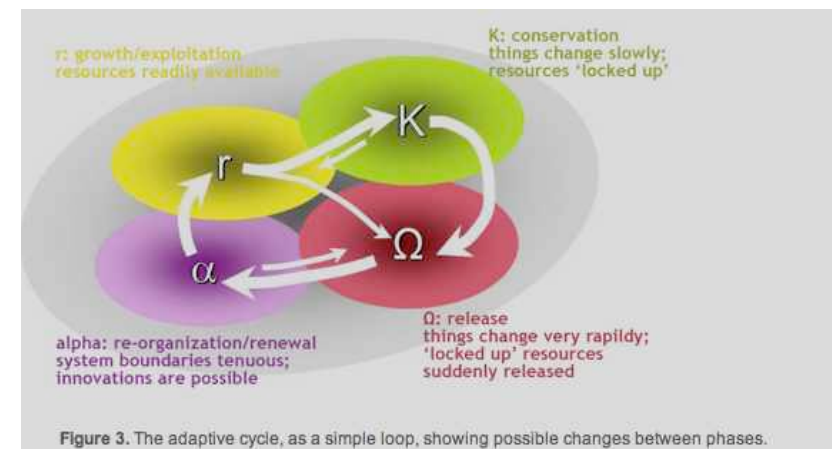


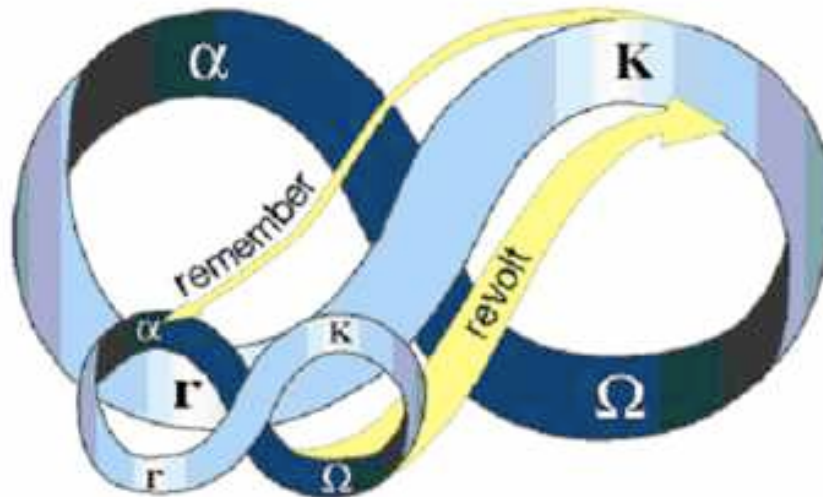
Figure 3. The adaptive cycle, as a simple loop, showing possible changes between phases.

# **“No system can be understood or managed by focusing on it at a *single* scale.”**

## **Multiple scales and cross-scale effects - "Panarchy"**

**No system can be understood or managed by focusing on it at a single scale.**

- **" All systems (and SESs especially) exist and function at multiple scales of space, time and social organization,**
  - **" and the interactions across scales are fundamentally important in determining the dynamics of the system at any particular focal scale.**
  - **" This interacting set of hierarchically structured scales has been termed a "panarchy" (Gunderson and Holling 2003).**



**Figure 4. "Panarchy" - nested adaptive cycles, with influences between scales.**

# Flexibility:

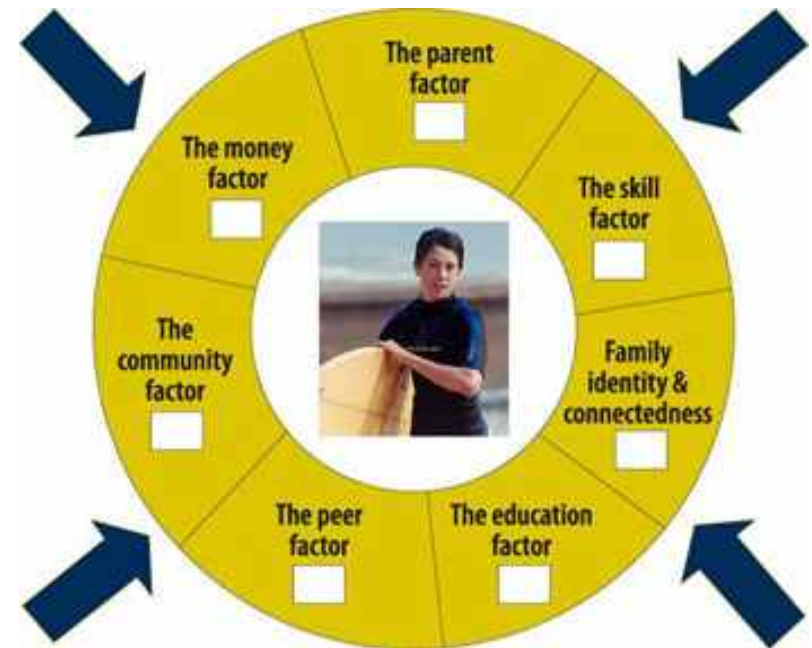
**Gist: 'Flexibility' concerns the  
'in-built' ability of the system  
to adapt,  
or to be adapted,  
by its users,  
to suit conditions  
*(without any fundamental system  
modification  
by system development).***

**Type: Complex Quality Requirement.**

**Includes: {Connectability, Tailorability}.**

**See next 2 slides!**

**Possible Synonyms: Resilience,  
Robustness**



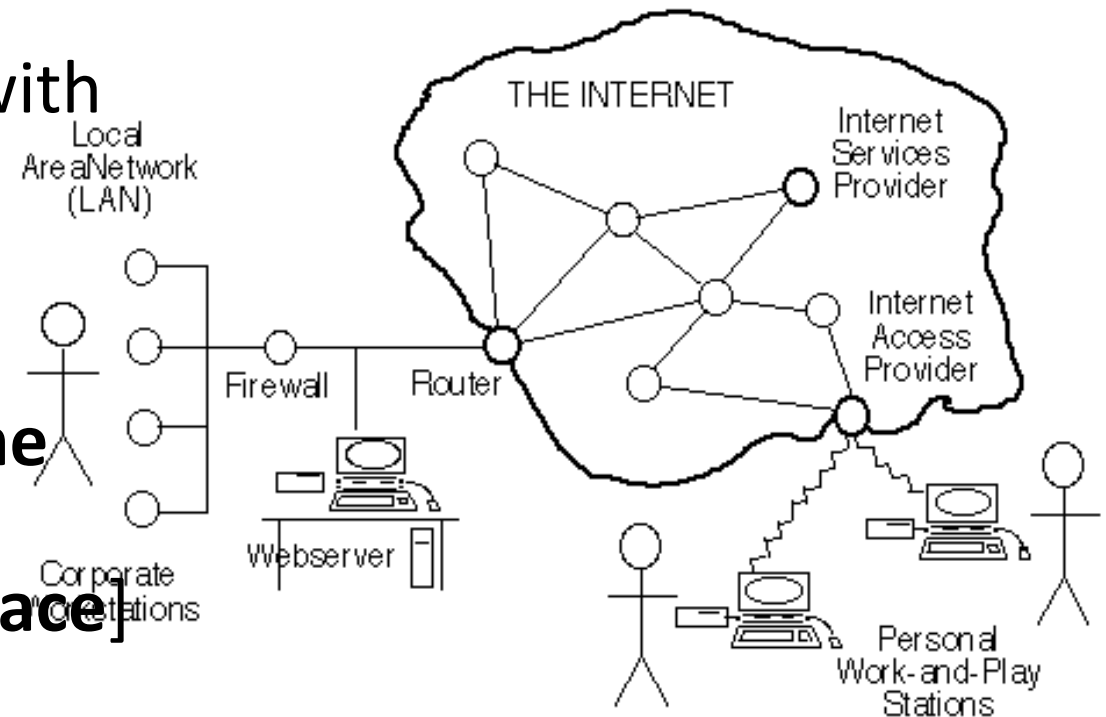
## Connectability:

‘The cost to interconnect the system to its environment.’

**Gist:** The cost of connecting  
**one set of interfaces** to  
defined **environments** with  
**other interfaces**

**Part Of:** Flexibility.

**Scale:** the **Effort** needed  
to connect a defined [**Home  
Interface**]  
to a defined [**Target Interface**]  
using defined [**Methods**]  
with minimum allowed  
system [**Degradation**].





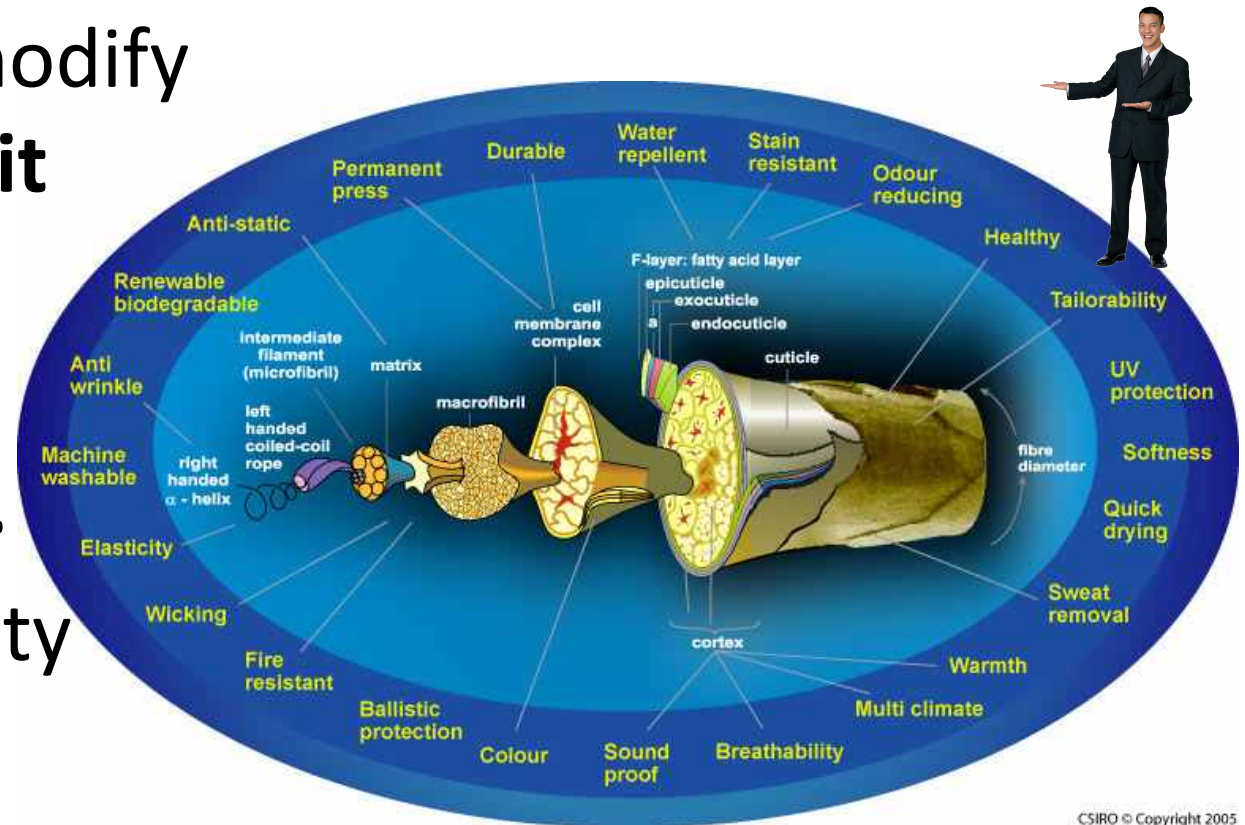
# Tailorability:

**Gist:** The **cost** to modify the system to **suit** defined **future** conditions.

**Part Of:** Flexibility.

**Type:** *Complex* Quality Requirement.

**Includes:** {Extendibility, Interchangeability}.



CSIRO © Copyright 2005

Multiple Attributes of Wool Fiber !



# Extendibility: Scalability

## Extendibility:

**Part Of:** Tailorability.

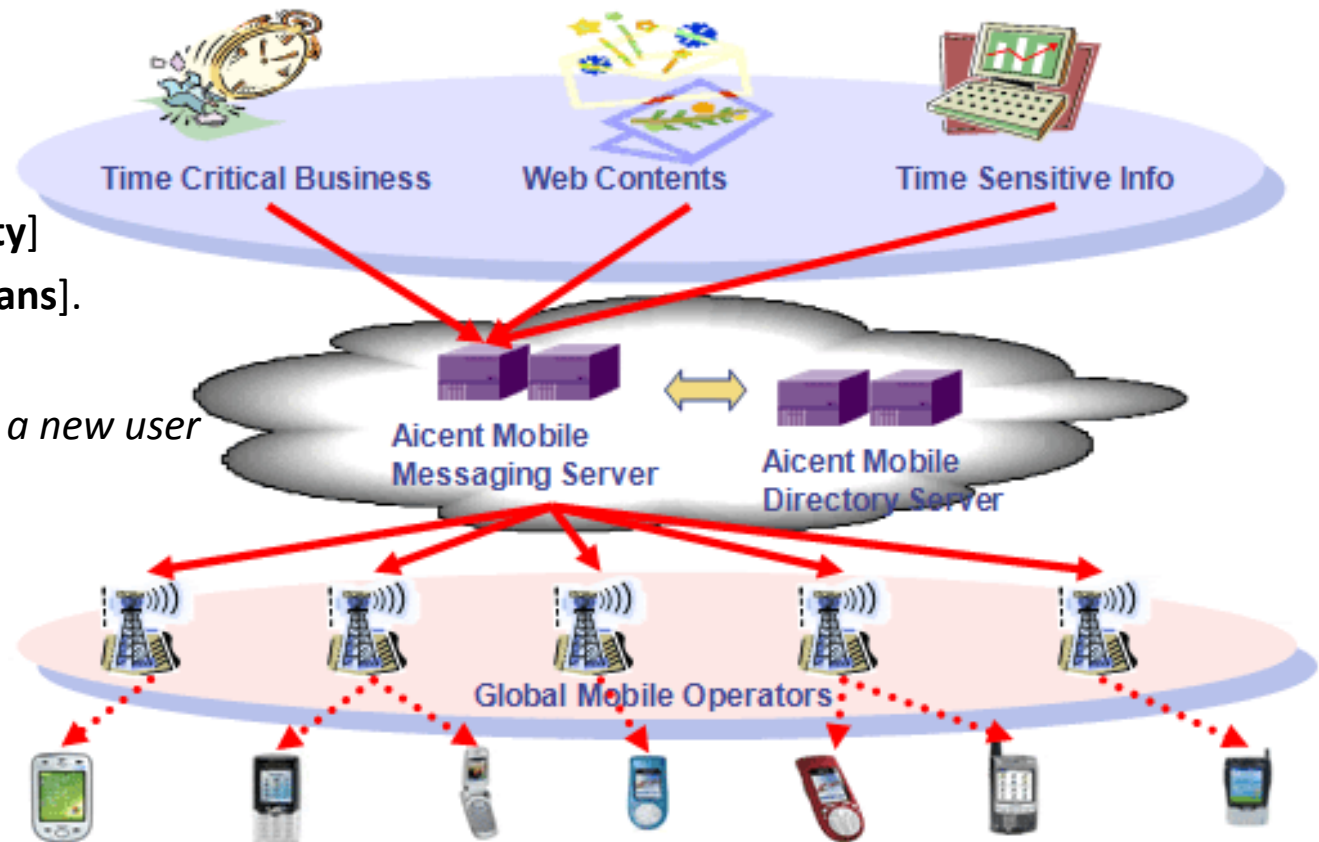
**Synonym:** Scalability.

**Scale:** The **cost** to add to  
a defined [**System**]  
a defined [**Extension Class**]  
and defined [**Extension Quantity**]  
using a defined [**Extension Means**].

*“In other words, add such things as a new user  
or  
a new node.”*

**Type:** *Complex* Quality Attribute.

**Includes:** {Node Addability,  
Connection Addability,  
Application Addability,  
Subscriber Addability}.



# Interchangeability:

‘The cost to modify use of system components.’

## Interchangeability

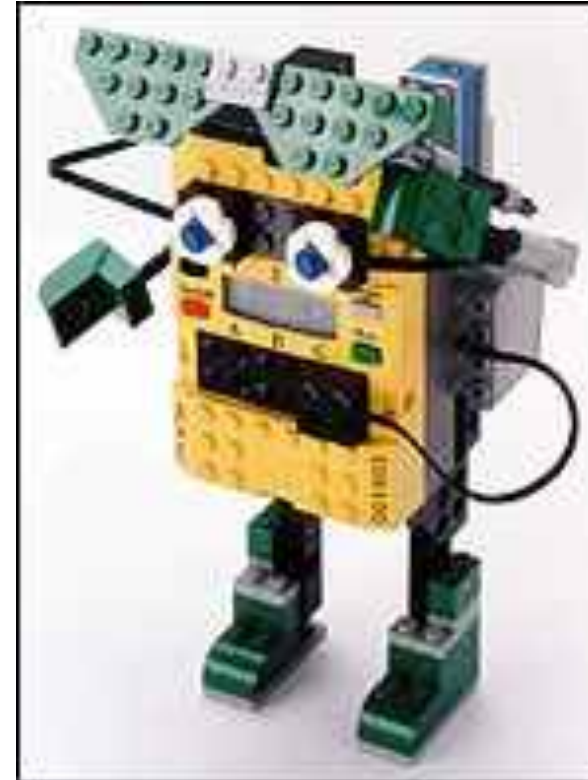
**Gist:** This is concerned with the ability to modify the system, to switch from using a certain set of system components, to using another set.

**Part Of:** Tailorability.

**Type:** Elementary Quality Attribute.

*“For example, this could be a daily occurrence switching system mode from day to night use.”*

**Scale:** the Effort needed to  
Successfully,  
without Intolerable Side Effects,  
replace a defined [Initial Set] of components,  
with a defined [Replacement Set] of  
system components,  
using defined [Means].



# Upgradeability:

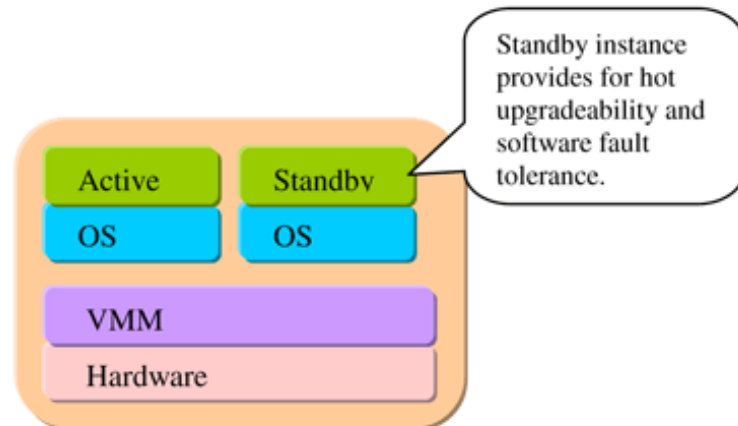
‘The cost to modify the system fundamentally;  
either to install it, or to change out system components.’

## Upgradeability:

**Gist:** This concerns the ability of the system to be modified by the system developers or system support in planned stages (as opposed to unplanned maintenance or tailoring the system).

**Type:** *Complex* Quality Requirement.

**Includes:** {Installability, Portability, Improveability}.



**Installability:** ‘The cost to install in defined conditions.’

**Pattern:** This concerns installing the system code and also, installing it in new locations to extend the system coverage. Could include conditions such as the installation being carried out by a customer or, by an IT professional on-site.

**Portability:** ‘The cost to move from location to location.’

**Scale:** The cost to transport a defined [System] from a defined [Initial Environment] to a defined [Target Environment] using defined [Means].

**Type:** Complex Quality Requirement.

**Includes:** {Data Portability, Logic Portability, Command Portability, Media Portability}.

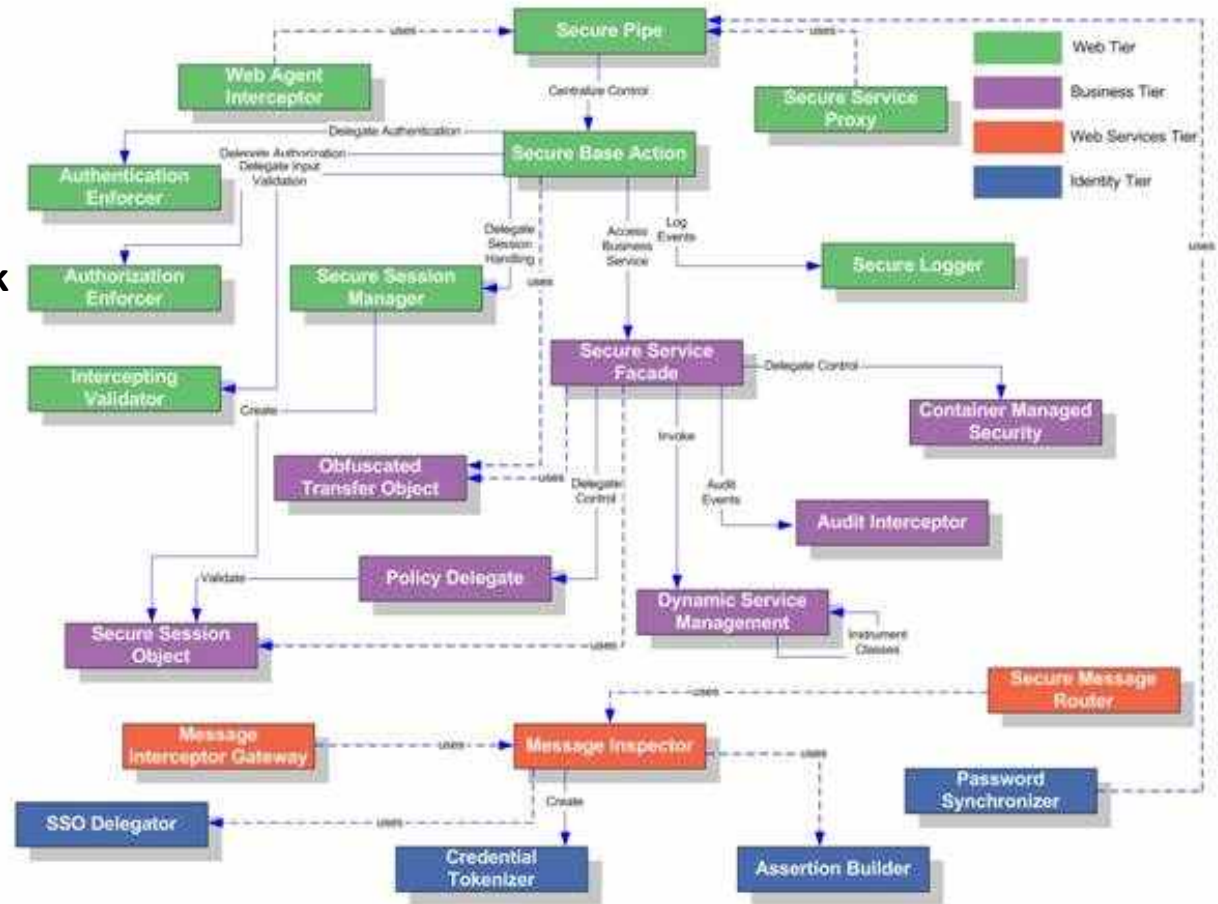
**Improveability:** ‘The cost to enhance the system.’

**Gist:** The ability to replace system components with others, which possesses improved (function, performance, cost and/or design) attributes.

**Scale:** The cost to add to a defined [System] a defined [Improvement] using a defined [Means].

- " Hopefully this set of patterns
  - " gives you a departure point
  - " for defining those maintenance attributes
  - " you might want to control, quantitatively.
- " The above adaptability definition
  - " was use to co-ordinate the work
    - " of 5,000 software engineers,
    - " and 5,000 hardware engineers,
    - " in UK,
    - " in bringing out a new product line at a computer manufacturer.
    - " Where 'Adaptability' was the Number One Product Characteristic
  - " The Company became profitable for the next 14 years..

## This Basic 'Adaptability' Pattern Was Successfully Applied



## Security Patterns



# The Software Architect Role in Maintainability

The role of the software architect is:

- to participate in **clarification of the requirements** that will be used as inputs to their architecture process.
- to insist that the requirements are **testably clear**: that means with defined and agreed scales of measure, and defined required levels of performance.
- to then **discover appropriate architecture**,
  - " capable of delivering those levels of performance, hopefully within resource constraints, and
- **estimate** the probable **impact** of the architecture,
  - " on the requirements (Impact Estimation)
- **define** the architecture in such **detail**
  - " that the intent **cannot be misunderstood** by implementers,
  - " and the desired **effects** are bound to be **delivered**.
- **monitor** the developing system as the architecture is applied in practice,
- and **make necessary adjustments**.
- finally **monitor** the **performance characteristics** throughout the lifetime of the system,
  - " and make necessary **adjustments** to requirements
  - " and to architecture,
  - " in order to **maintain** needed system **performance** characteristics.





# Engineering “Maintainability”: Green Week

## Weekly ‘Refactoring’ at Conconfirm

Current Status		Improvement	Goals			Step 6 (week 14)		Step 7 (week 15)	
	Units		Past	Tolerable	Goal	Estimated Impact	Actual Impact	Estimated Impact	Actual Impact
	100,0	100,0	0	80	100			100	100
Speed									
	100,0	100,0	0	80	100	100	100		
Maintainability.Doc.Code									
	100,0	100,0	0	80	100	100	100		
InterviewerConsole									
NUnitTests									
	0,0	0	0	90	100				
PeerTests									
	100,0	100,0	0	90	100			100	100
FxCop									
	0,0	10,0	10	0	0				
TestDirectorTests									
	100,0	100,0	0	90	100			100	100
Robustness.Correctness									
	2,0	2,0	0	1	2	2	2		
Robustness.BoundaryConditions									
	0,0	0,0	0	80	100				
Speed									
	0,0	0,0	0	80	100				
ResourceUsage.CPU									
	100,0	0,0	100	80	70	70			
Maintainability.Doc.Code									
	100,0	100,0	0	80	100	100	100		
SynchronizationStatus									
NUnitTests									

Speed

Maintainability

Nunit Tests

PeerTests

TestDirectorTests

Robustness.Correctness

Robustness.Boundary  
Conditions

ResourceUsage.CPU

Maintainability.DocCode

SynchronizationStatus

POT-SHOTS — Brilliant Thoughts in 17 words or less



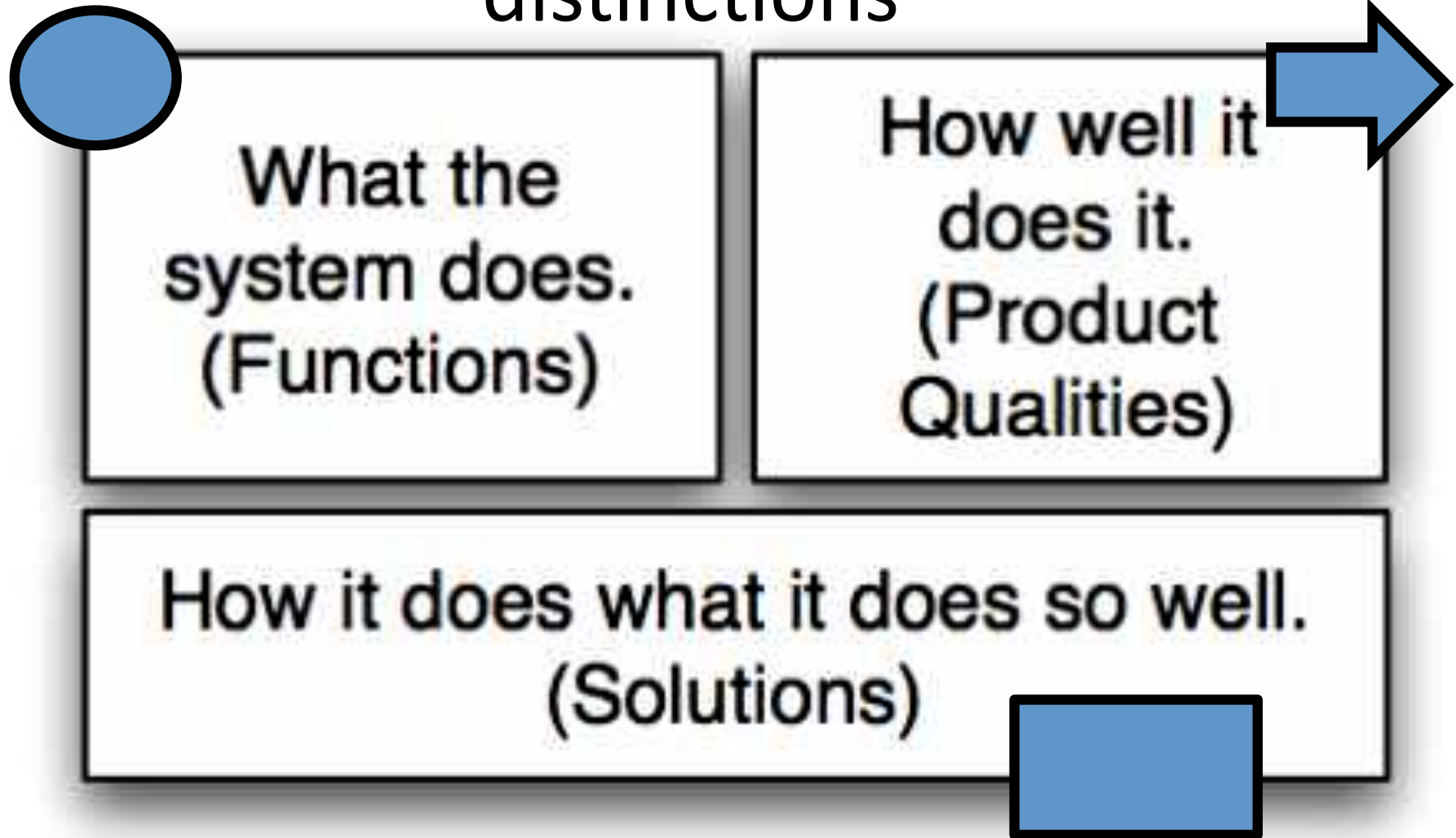
September 12, 2009

SPA BCS 2 Sept 2009 London

232

# 10. Specifying Technical Means for meeting quantified Quality requirements

# 3 views of a system: Powerful distinctions



# What is a 'design'? (architecture, solution)

## Design Idea !!

Concept \*047 March 15, 2003 !

- " **A design idea is**
  - " *anything*
  - " *that will satisfy*
  - " *some requirements.*
- " **A set of design ideas**
  - " *is usually needed to solve a larger 'design problem'.*

•"

Marketing

Brand

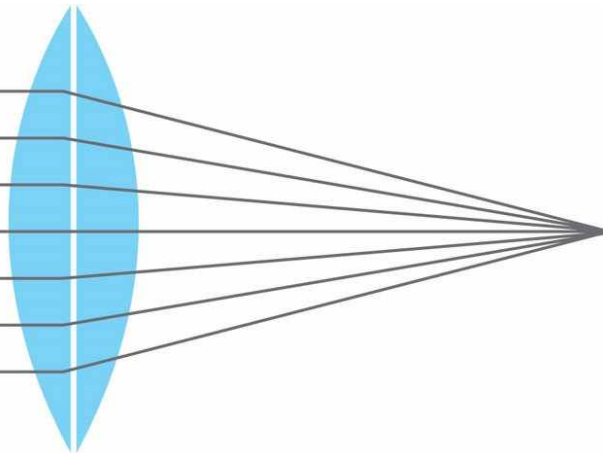
Product

Customer services

Legal

Operations

Retail



### SCALAR REQUIREMENT SPECIFICATION

**Participation:** Scale: % of worldwide membership participating. Goal: 10%.

**Representation:** Scale: % of worldwide membership represented within defined <groups>.

Goal [Age under 25 or equating to <student status>]: 10%.

**Information:** Scale: % of talks rated as 'good' or better (5+ on feedback sheet scale). Goal: 50%.

**Conviction:** Scale: % participants wanting to return next conference. Goal: 80%.

**Influence:** Scale: % participants who <improve as result of the conference>.

Past: 90%, Goal: 95%.

**Fun:** Scale: % participants rating the conference-city quality as 'good' or better (5+ on feedback sheet scale).

Past: 45%. Plan: 60%.

**Cost:** Resource Budget: Scale: total cost for an individual participant including travel costs.

Fail: \$2,000. Goal: \$1,200 or less.

### DESIGN SPECIFICATION (simple version)

**Central:** Choose a location in the membership center of gravity (New York?)

**Youth:** Suggest and support local campaigns to finance 'sending' a young representative to conference.

**Facts:** Review all submitted papers on <content>.

**London:** Announce that the conference is to be in London next time.

**Diploma:** Give diplomas for attendance, and additional diplomas for individual tutorial courses.

**Events:** Have entertainment activities organized every evening: river tours, etc.

**Discounts:** Get discounts on airfare and hotels.

## Example of a (Real, partial) Design Specification using Planguage

**Tag:** OPP Integration.

**Type:** Design Idea [Architectural].

===== Basic Information =====

Version:

Status:

Quality Level:

Owner:

Expert:

Authority:

**Source:** System Specification Volume 1 Version 1.1, SIG, February 4. - Precise reference <to be supplied by Andy>.

**Gist:** The X-999 would integrate both 'Push Server' and 'Push Client' roles of the Object Push Profile (OPP).

**Description:** Defined X-999 software acts in accordance with the <specification> defined for both the Push Server and Push Client roles of the Object Push Profile (OPP).

Only when official certification is actually and correctly granted; has the {developer or supplier or any real integrator, whoever it really is doing the integration} completed their task correctly.

This includes correct proven interface to any other related modules specified in the specification.

**Stakeholders:** Phonebook, Scheduler, Testers, <Product Architect>, Product Planner, Software Engineers, User Interface Designer, Project Team Leader, Company engineers, Developers from other Company product departments which we interface with, the supplier of the III, CC. "Other than Owner and Expert. The people we are writing this particular requirement for"

===== Design Relationships =====

Reuse of Other Design:

Reuse of this Design:

Design Constraints:

Sub-Designs:

===== Impacts Relationships =====

Impacts [Intended]: Interoperability.

Impacts [Side Effects]:

Impacts [Costs]:

Impacts [Other Designs]:

Value:

Interoperability: Defined As: Certified that this device can exchange information with any other device produced by this project.

===== Impact Estimation/Feedback =====

**Impact Percentage** [Interoperability Estimate]: <100% of Interoperability objective with other devices that support OPP on time is estimated to be the result>.

===== Priority and Risk Management =====

**Assumptions:** There are some performance requirements within our certification process regarding probability of connection and transmission etc. that we do not remember <-TG.

Dependencies:

**Risks:** <none identified>.

We do not 'understand' fully (because we don't have information to hand here) our certification requirements, so we risk that our design will fail certification.  
<-TG

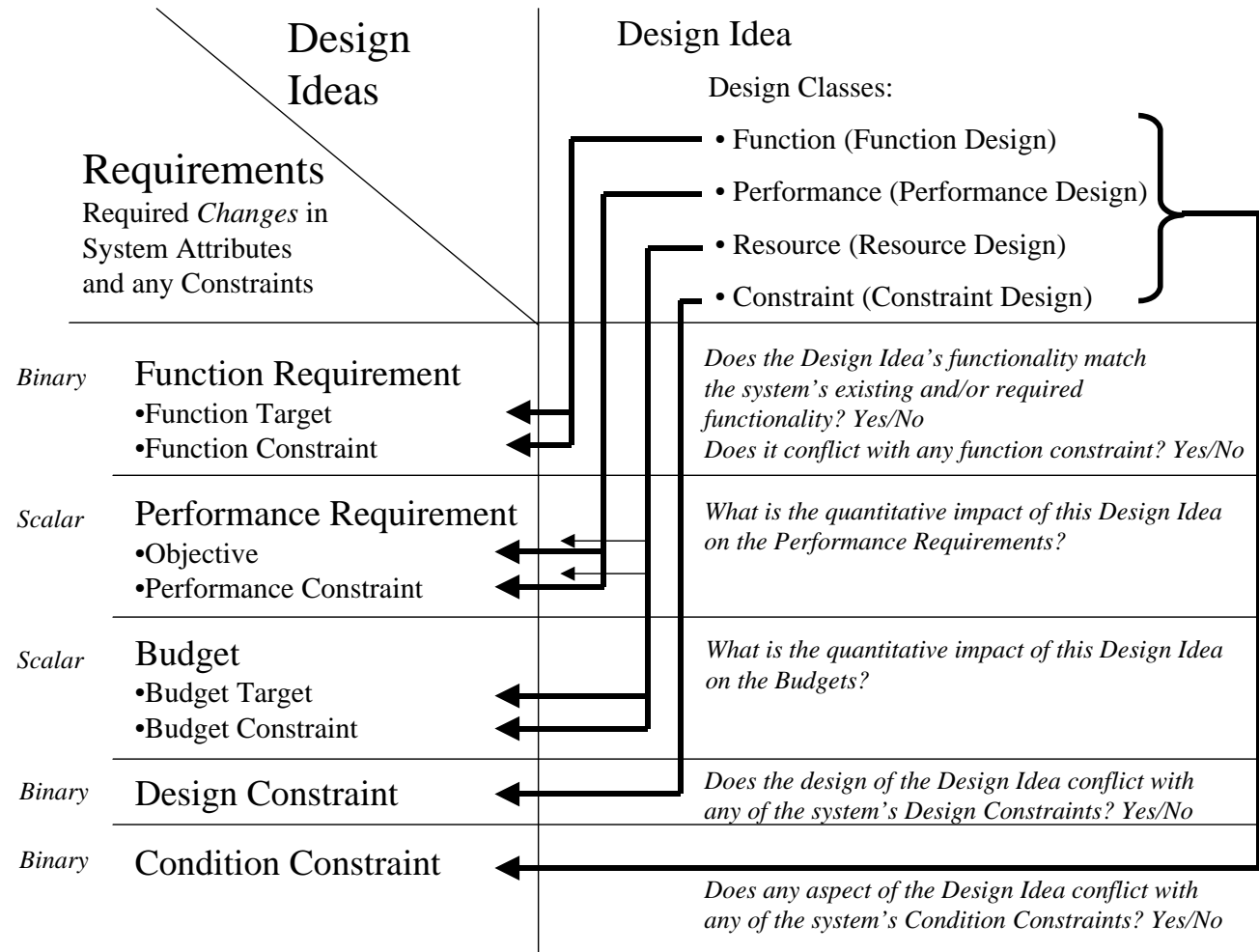
Priority:

Issues:

===== Location of Specification =====

**Location of Master Specification:** <Give the intranet web location of this master specification>.

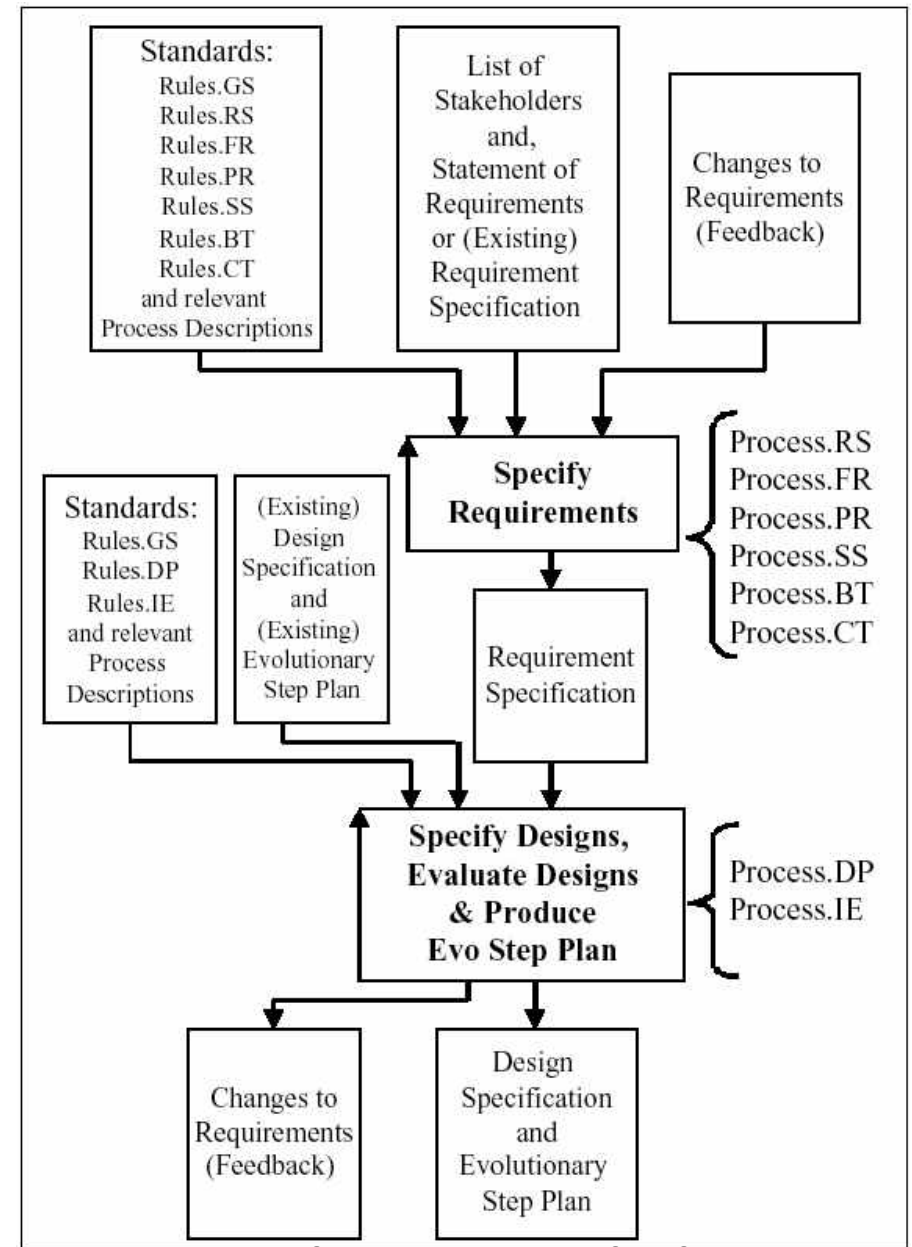
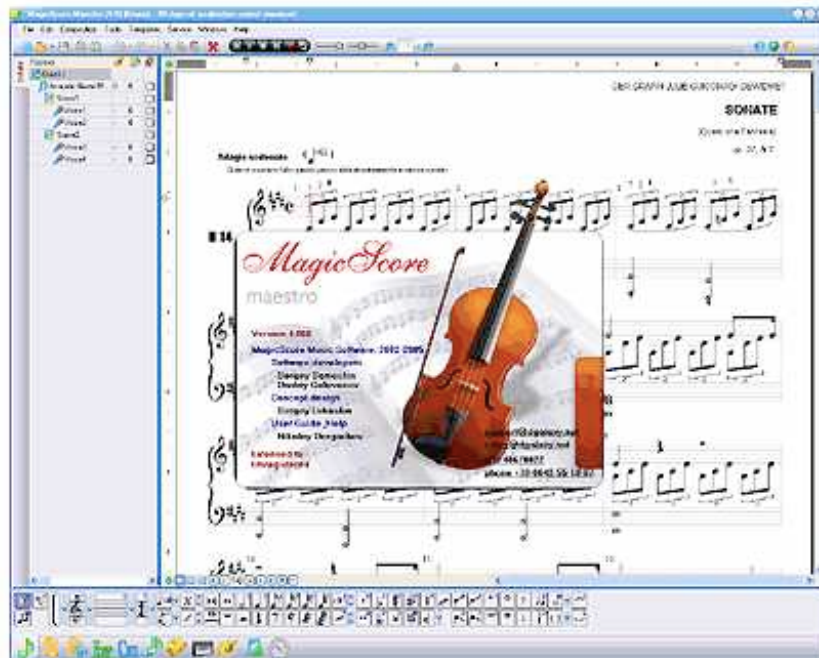
# What are the principles of evaluating a design?



- " Avoid violating constraints
- " Meet Target and Function requirements



# Planguage Standards for Design



Planguage standards

# Design Ideas Confirm Case

- " For every quality requirement we look for possible Design Ideas
- " E.g. for Quality Requirement: Usability.Productivity we identified the following Design Ideas:
  - " DesignIdea.Recoding    Estimated Impact    20 Minutes saved (of 40 minutes needed saved)
  - " DesignIdea.MRTotals                                 13
  - " DesignIdea.Categorizations                                 8
  - " DesignIdea.TripleS                                         3
  - " ..and many more
- " We evaluated all these, and specified in more detail those we believed would add the most value (take us closer to the goal)
- " A chosen Design Idea = Solution



# How do we specify a design with *impacts*?

## *A Template to make us think competitively*

Tag: <Unique Name Capitalized>

Type: Design Idea.

Version: <date and or version number of last change>

Owner: <originator, champion, expert, maintainer, architect, systems engineer>

Description: <describe the design in a dozen, or more, words. The detail should be sufficient to guarantee the expected impacts and costs estimated below>.

Reuse: <if a currently available component or design is specified, then give it's tag or reference code here to indicate that a known component is being applied>

**Primary Impacts:** <give the main impact or impacts which this design is expected to have on an objective . These are its main justification for existence!>.

**Secondary Impacts:** <list expected secondary impacts, good or bad>.

**Cost Impacts:** <give at least rough impacts on defined budget constraints>.

===== More Formal Impact Estimation =====

**Real Impact on defined Scale:** <give expected impact result on the Scale defined, when implemented>

**%Impact on Specific Goal:** <Convert real impact to % impact relative to the main planned level: 100% means meets defined Plan level on time>.

**± %Uncertainty:** <give optimistic/pessimistic % deviation, like ±20%, based on best and worst real observations>.

**Evidence:** <give the observed numbers, facts, dates, places where you have data about this designs impact>

**Source:** <give the person or written source of your evidence>

**Credibility:** <Credibility 0.0 low to 1.0 high. Rate the quality of your estimates, based on the historic data you have>

----- Repeat this sequence for any other major impact objectives you believe justify the specification effort here.

===== Other Useful Parameters for Design Specification =====

**Risks:** <name any factors, which can threaten your estimated impact or bring it to the lowest levels specified>

**Assumptions:** <state any implied unvoiced, threatening assumptions which if false could threaten your estimates>

Web Location of Master Specification: <give intranet web location of this master specification>.

# 11. Controlling the Evolutionary Delivery Cycle using Impact Estimation Tables.

Decomposition to small steps

Step measurement and testing

Learning from results

Changing short term plans to meet long term goals

# Evo planning - example

- IET for MR Project – Confrimit 8.5
- **Solution: Recoding**
  - Make it possible to recode variable on the fly from Reportal.
  - Estimated effort: 4 days

	A	B	C	D	E	F	G	BX	BY	BZ	CA
1											
2		Current Status	Improvements		Goals			Step9			
3								Recoding			
4								Estimated impact		Actual impact	
5		Units	Units	%	Past	Tolerable	Goal	Units	%	Units	%
6					Usability.Replacability (feature count)						
7		1,00	1,0	50,0	2	1	0				
8					Usability.Speed.NewFeaturesImpact (%)						
9		5,00	5,0	100,0	0	15	5				
10		10,00	10,0	200,0	0	15	5				
11		0,00	0,0	0,0	0	30	10				
12					Usability.Intuitiveness (%)						
13		0,00	0,0	0,0	0	60	80				
14					Usability.Productivity (minutes)						
15		20,00	45,0	112,5	65	35	25	20,00	50,00	38,00	95,00
20					Development resources						
21			101,0	91,8	0		110	4,00	3,64	4,00	3,64

# Impact Estimation DD Case

## numeric evaluation of design



Goal: Increase Time to Sell (Individual hours devoted to direct sales activities) from 12 hrs/wk to 28 hrs/wk (30% to 70% of their time)		Design: Build New Accounts Wizard	Design: Electronically send data to SOR	Design	Totals
		Design Ideas			
Current Benchmark	12 hrs / wk	12	12	12	
Target Goal	28 hrs / wk	28	28	28	
Scale Impact	hrs / wk	1	2.5	0	3.5
Scale Uncertainty	+ / - hrs/wk	0.5	1	0	1.5
Percentage Impact	on design	6%	16%	0%	22%
Percentage Uncertainty	percentage	3%	6%	0%	9%
Evidence	based upon	Aneecdotal	High level estimate		
Source	person or doc	Ryan [06/18/07]	Ryan [06/20/07]		
Credibility	and 1	0.7	0.5		
Costs					
Solution Owner	effort hours	20	30	0	
Analysis	effort hours	70	200	0	270
Development	effort hours	100	300	0	400
Testing	effort hours	20	60	0	80
Total Resources	effort hours	210	590	0	800
Performance to Cost Ratio	of design	0.030	0.026	#VALUE!	
Credibility-adjusted					
Performance to Cost Ratio	factored in	0.021	0.013	#VALUE!	

September 12, 2009

SPA BCS 2 Sept 2009 London

243



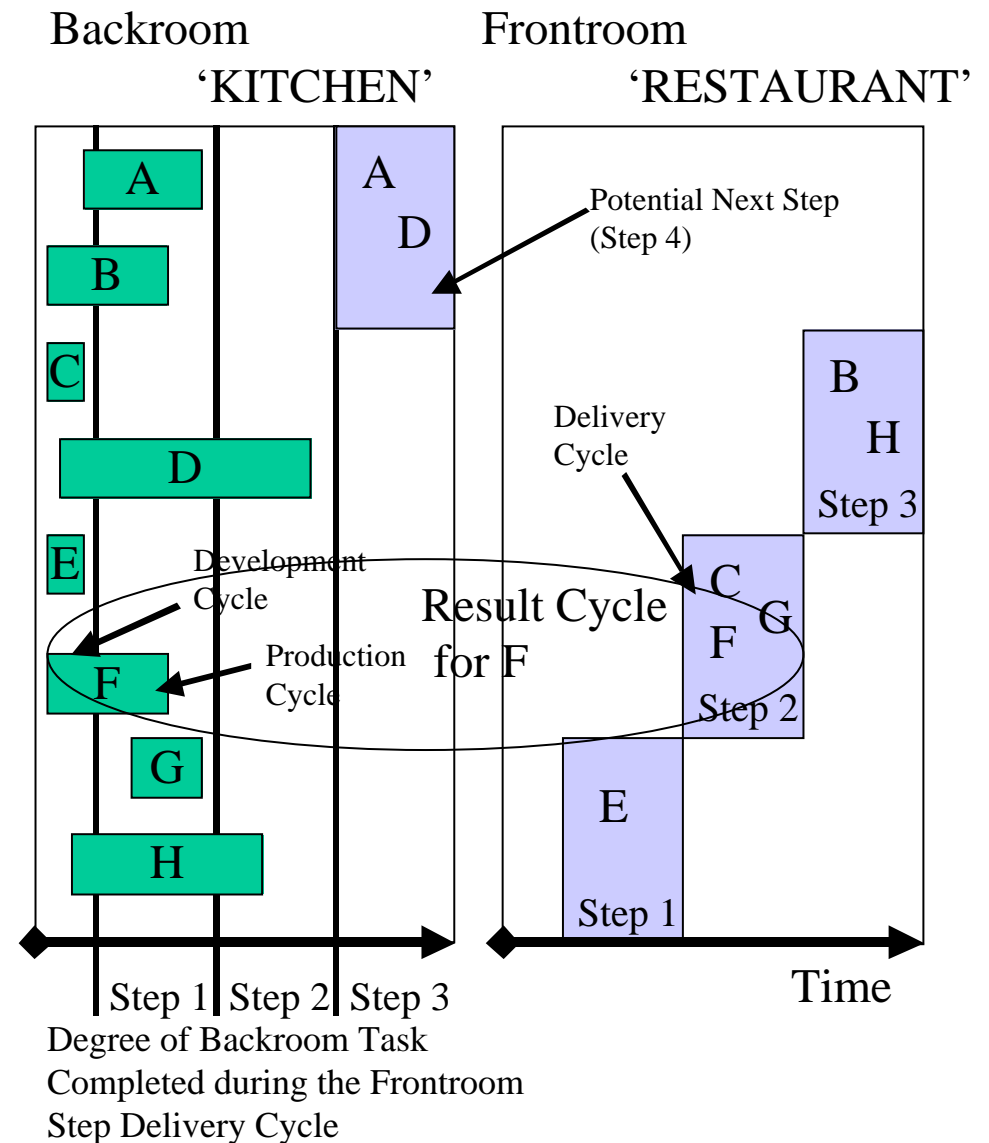
# How does Evo relate to requirements?

Step-> Target Require- ment	<u>STEP1</u> Plan % (of Target)	actual %	deviation %	<u>STEP2 to STEP20</u> Plan %	plan cumulated to here %	<u>STEP21</u> [CA,NV,WA] Plan %	plan cumulated to here %	<u>STEP22</u> [all others] Plan %	plan cumulated to here %
<u>PERF-1</u>	5	3	-2	40	43	40	83	-20	63
<u>PERF-2</u>	10	12	+2	50	62	30	92	60	152
<u>PERF-3</u>	20	13	-7	20	33	20	53	30	83
<u>COST-A</u>	1	3	+2	25	28	10	38	20	58
<u>COST-B</u>	4	6	+2	38	44	0	44	5	49

- " Evo relates directly, measurably, testably, early and frequently to unfulfilled requirements.
- " Evo is always seeking the most efficient way to close the requirements gap and complete a project
- " The primary measure of Evo project progress is the degree of stakeholder satisfaction (in terms of agreed requirements) as a result of delivered Evo steps.

# How does Evo relate to process improvement?

- " Evo can measure
  - " the success of current processes against expectations,
  - " or new experimental ones against expectations
- " Evo can signal the need for process improvement and verify that such improvement has taken place
- " Evo can help you
  - " *early* in the project,
  - " continuously,
  - " and helps to *train* new people
    - " in the adopted processes
    - " by frequent cycles of practice and feedback



# How does Evo relate to competitiveness?

- "Evo is focused on delivery of quantified specified stakeholder value
- "Evo is 'agile'
  - "and can change plans, designs, processes, and requirements -
  - "in order to deliver the most competitive solutions
  - "early, gradually, and with smart priorities.

