



Dynamic Design Prioritization

in the 'Evo' Agile Framework
for Scrum or other Iterative Methods
October 9 2013, 14:00 to 15:30

Tom Gilb

Tom@Gilb.com, www.Gilb.com, @ImTomGilb

Author of 'Competitive Engineering'
Norway and London



Talk Content

Prioritization Method Basis

1. Incremental Delivery invites a different approach to Architecture
2. We can, and should, measure the costs and effects of different architecture ideas, before committing to specific ideas, and scaling up.
3. Software Metrics: a pre-requisite
4. We need to specify performance (quality) requirements, and cost budgets numerically
5. We need to estimate the multiple impacts of design components in advance, as the basis for prioritization
6. We need to deliver architecture components incrementally, and measure their actual effects in practice.
7. We need to replace or modify architecture components that fail expectations

REMARKS

- This is an 'engineering' approach.
- This is a scientific approach
- This is a risk management method

The highest priority for human survival is:

- Water
- Air
- Food



Did you answer the 'right question' ?

- If you answered 'air'
- You were probably answering the question:
 - “Which of these 3 things, if totally denied would kill a human fastest”
- That was not the question!
- *Rule One:*
 - *Listen to the Question*
 - *Make sure you understood it*
 - *Answer the question asked*

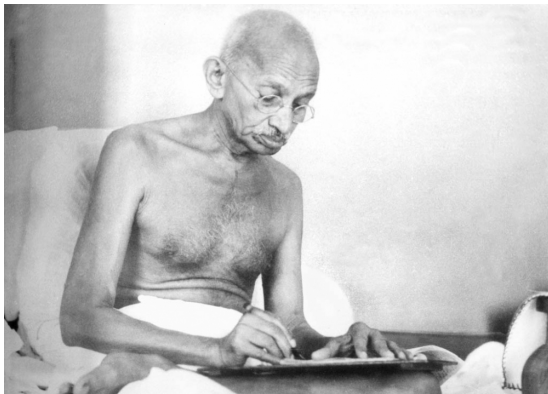
The highest priority for human survival is:

- Water
- Air
- Food



Is your 'AIR' answer valid if:

- We are in the Death Valley Desert
 - And we have enough air, but have no water and ate the rest of our food 5 days ago?
- You are on a hunger strike (Gandhi) in the 43rd Day and Nehru is worried?



Problems with the Question

- No information was given on ‘Constraints’
 - the ‘**survival** level’, or the ‘Fail level’
 - Suffocation, gasping for breath
- No information was given on degree of **satisfaction** of the requirements.
 - ‘Goal level’
 - (when we don’t have to think about breathing)



Problems with your mode of answering:

- You answered without asking necessary information
 - (requirements, degree of satisfaction)
- Your ‘wrong’ answer might be mistaken as an *authoritative* demand
 - (you are the manager, you *did not hesitate* to give a clear answer)
- You did not even ask ‘*why the question was asked*’
 - My answer: to trick you into unreasonable, dangerous behavior in public!
 - *Next time* your enemy might pull this trick!



Problems with your answer content:

- Based on insufficient information ABOUT
 - Needs (requirements, objectives)
 - Satisfaction degree (up to now)
 - Resources *available* (to satisfy *all* needs)



No limits stated as to:

- resources *to be used* for this *one* need
- time to satisfy the need before ‘death’ (TTM!)
- degree of satisfaction of your ‘priority’ choice
 - You risk ***over-satisfaction*** (no ‘*added value*’)
 - At the expense of *other* vital needs (food, water)

General observations on ‘priorities’

- They are *multiple* Critical Factors simultaneously
- Critical Factor’s needs - *vary* from project to project
- CF’s ‘degree of satisfaction’ *varies* as time goes by
- The ‘biggest gap’ *must* get our scarce resources
 - Otherwise we risk exhausting scarce resources *before* ‘survival’ levels are achieved for some critical factors
- The ‘real’ (‘Fundamental Objectives’) priority is:
SURVIVAL
 - Food, water, air are the ‘Strategic Objectives’



Now we have **basis** for 'Prioritization1

Prioritisation: defined

- Making a choice of alternatives, based on their potential ability to serve our defined needs
- We might prioritize a trial, or experiment, so that we can measure and see if the alternatives really do meet needs as expected

Prioritisation Policies

- Prioritisation must be based on a specific set of rules, but these rules can vary from time to time
- Basic policies include the concepts of
 - Effectiveness
 - Efficiency (Cost effectiveness)

Prioritization Principles

1. **EFFICIENCY:** In the Long Run: we should prioritise so that we maximize the stakeholder values delivered for limited resources
2. **RISK:** In the short term: we can prioritise in order to explore high value, but high risk, options.
3. **POLITICS:** We can consciously prioritise in order to maximise one particular objective
4. **SURVIVAL:** But in the medium term we must prioritise so that all system capabilities and costs avoid violating constraints
5. **BALANCE:** And in the long term we need to prioritise so that all top level critical objectives are met

Prioritization Policies

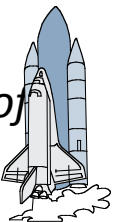
1. Change the priority decision rules to satisfy certain stakeholders.
 1. Please the major project funders
 2. Please your boss
 3. Please the market
2. Change priority rules to satisfy emerging events
 1. Stop prioritising designs, that improve performance attributes, that have already *reached* their Goal levels.
 2. Start choosing designs that use resources, which are currently far more available than others
3. Apply rules that consider explicit risk for a design:
 1. \pm uncertainty of results (wide range of experience)
 2. Credibility level (evidence for effects, sources of effects)

Cleanroom

In the Cleanroom Method, developed by IBM's Harlan Mills (1980) they reported:



- *“Software Engineering began to emerge in FSD” (IBM Federal Systems Division, from 1996 a part of Lockheed Martin Marietta) “some ten years ago [Ed. about 1970] in a continuing evolution that is still underway:*
- *Ten years ago general management expected the worst from software projects – cost overruns, late deliveries, unreliable and incomplete software*
- *Today [Ed. 1980!], management has learned to expect on-time, within budget deliveries of high-quality software. A Navy helicopter ship system, called LAMPS, provides a recent example. LAMPS software was a four-year project of over 200 person-years of effort, developing over three million, and integrating over seven million words of program and data for eight different processors distributed between a helicopter and a ship in 45 incremental deliveries [Ed. Note 2%!].s. Every one of those deliveries was on time and under budget*
- *A more extended example can be found in the NASA space program,*
- *- Where in the past ten years, FSD has managed some 7,000 person-years of software development, developing and integrating over a hundred million bytes of program and data for ground and space processors in over a dozen projects.*
- *- There were few late or overrun deliveries in that decade, and none at all in the past four years.”*



In the Cleanroom Method, developed by IBM's Harlan Mills (1980) they reported:

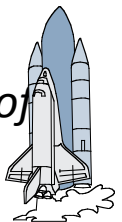


- “Software Engineering began to emerge in FSD” (IBM Federal Systems Division,

in 45 incremental deliveries

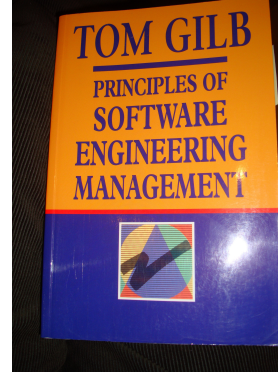
cost overruns, late deliveries, unreliable and incomplete software

- Today [Ed. 1980!], management has learned to expect on-time, within budget deliveries of high-quality software. A Navy helicopter ship system, called LAMPS, provides a recent example. LAMPS software was a four-year project of over 200 person-years of effort developing over three million and integrating over seven million lines of code. The project was completed on time and within budget. There were few late or overrun deliveries in that decade, and none at all in the past four years.
- A more recent example is the development of the LAMPS software for the Navy helicopter ship system. The project was completed on time and within budget. There were few late or overrun deliveries in that decade, and none at all in the past four years.
- - When the software was delivered, it was of high quality. There were few late or overrun deliveries in that decade, and none at all in the past four years.
- - There were few late or overrun deliveries in that decade, and none at all in the past four years.



Quinnan: IBM FSD Cleanroom

Dynamic Design to Cost



Quinnan describes the process control loop used by IBM FSD to ensure that cost targets are met.

'Cost management. . . yields valid cost plans linked to technical performance. Our practice carries cost management farther by introducing design-to-cost guidance. Design, development, and managerial practices are applied in an integrated way to ensure that software technical management is consistent with cost management. The method [illustrated in this book by Figure 7.10] consists of developing a design, estimating its cost, and ensuring that the design is cost-effective.' (p. 473)

-
He goes on to describe a design iteration process trying to meet cost targets by either redesign or by sacrificing 'planned capability.' When a satisfactory design at cost target is achieved for a single increment, the 'development of each increment can proceed concurrently with the program design of the others.'

'Design is an iterative process in which each design level is a refinement of the previous level.' (p. 474)

It is clear from this that they avoid the big bang cost estimation approach. Not only do they iterate in seeking the appropriate balance between cost and design for a single increment, but they iterate through a series of increments, thus reducing the complexity of the task, and increasing the probability of learning from experience, won as each increment develops, and as the true cost of the increment becomes a fact.

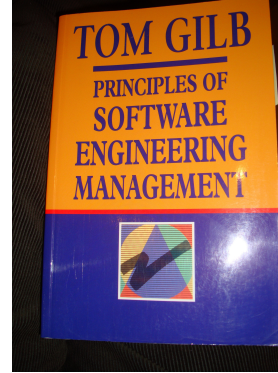
'When the development and test of an increment are complete, an estimate to complete the remaining increments is computed.' (p. 474)

Source: Robert E. Quinnan, 'Software Engineering Management Practices', IBM Systems Journal, Vol. 19, No. 4, 1980, pp. 466~77

This text is cut from Gilb: The Principles of Software Engineering Management, 1988

Quinnan: IBM FSD Cleanroom

Dynamic Design to Cost



Quinnan describes the process control loop used by IBM FSD to ensure that cost targets are met.

'Cost management far management far are applied in an management. Th its cost, and ens

- He goes o sacrificing 'plan the 'development

of developing a design, estimating its cost, and ensuring that the design is cost-effective

actice carries cost and managerial practices consistent with cost ing a design, estimating

y either redesign or by for a single increment, gn of the others.'

'Design is an iterative process in which each design level is a refinement of the previous level.' (p. 474)

It is clear from this that they avoid the big bang cost estimation approach. Not only do they iterate in seeking the appropriate balance between cost and design for a single increment, but they iterate through a series of increments, thus reducing the complexity of the task, and increasing the probability of learning from experience, won as each increment develops, and as the true cost of the increment becomes a fact.

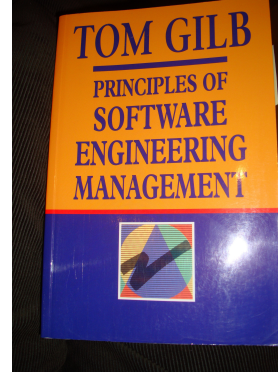
'When the development and test of an increment are complete, an estimate to complete the remaining increments is computed.' (p. 474)

Source: Robert E. Quinnan, 'Software Engineering Management Practices', IBM Systems Journal, Vol. 19, No. 4, 1980, pp. 466~77

This text is cut from Gilb: The Principles of Software Engineering Management, 1988

Quinnan: IBM FSD Cleanroom

Dynamic Design to Cost



Quinnan describes the process control loop used by IBM FSD to ensure that cost targets are met.

'Cost management. . . yields valid cost plans linked to technical performance. Our practice carries cost management farther by introducing design-to-cost guidance. Design, development, and managerial practices are applied in an integrated way to ensure that software technical management is consistent with cost management. The method [illustrated in this book by Figure 7.10] consists of developing a design, estimating its cost, and ensuring that the design is cost-effective.' (p. 473)

He goes on to describe the process of sacrificing 'planned' capability for the 'development' of the system.

'Design is an iterative process'

It is clear that the process of seeking the appropriate series of incremental improvements through experience, won't be a simple one.

'When the development of the system is complete, the remaining increments are sacrificed.'

Source: Robert E. Quinn, 'The Cleanroom Approach to Software Development', IBM Systems Journal, Vol. 19, No. 4, 1980, pp. 466~77.

**iteration process
trying to meet cost
targets by either
redesign or by
sacrificing 'planned
capability'**

by either redesign or by sacrificing 'planned' capability for a single increment, or by the sacrifice of the others.

ous level.' (p. 474)

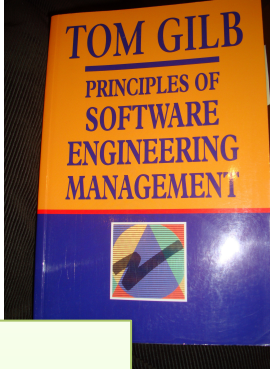
only do they iterate in the process, they iterate through a series of incremental improvements until the probability of learning from experience becomes a fact.

te the remaining

. 19, No. 4, 1980, pp. 466~77

Quinnan: IBM FSD Cleanroom

Dynamic Design to Cost



Design is an iterative process

It is clear from this that they avoid the big bang cost estimation approach. Not only do they iterate in seeking the appropriate balance between cost and design for a single increment, but they iterate through a series of increments, thus reducing the complexity of the task, and increasing the probability of learning from experience, won as each increment develops, and as the true cost of the increment becomes a fact.

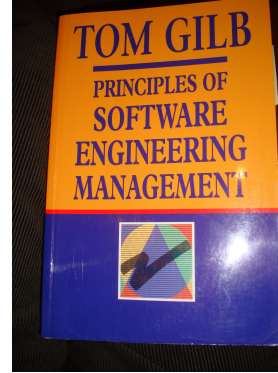
'When the development and test of an increment are complete, an estimate to complete the remaining increments is computed.' (p. 474)

Source: Robert E. Quinnan, 'Software Engineering Management Practices', IBM Systems Journal, Vol. 19, No. 4, 1980, pp. 466~77

This text is cut from Gilb: The Principles of Software Engineering Management, 1988

Quinnan: IBM FSD Cleanroom

Dynamic Design to Cost



Quinnan describes the process control loop used by IBM FSD to ensure that cost targets are met.

**but they iterate through a series of
increments,
thus *reducing the complexity of the
task,*
and *increasing the probability of
learning from experience***

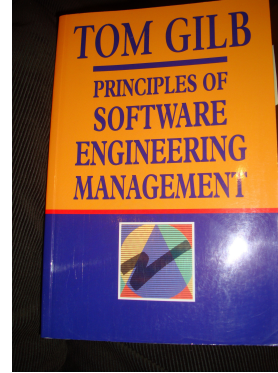
'When the development and test of an increment are complete, an estimate to complete the remaining increments is computed.' (p. 474)

Source: Robert E. Quinnan, 'Software Engineering Management Practices', IBM Systems Journal, Vol. 19, No. 4, 1980, pp. 466~77

This text is cut from Gilb: The Principles of Software Engineering Management, 1988

Quinnan: IBM FSD Cleanroom

Dynamic Design to Cost



Quinnan describes the process control loop used by IBM FSD to ensure that cost targets are met.

**an estimate to
complete the remaining
increments is
computed.**

'When the development and test of an increment are complete, an estimate to complete the remaining increments is computed.' (p. 474)

Source: Robert E. Quinnan, 'Software Engineering Management Practices', IBM Systems Journal, Vol. 19, No. 4, 1980, pp. 466~77

This text is cut from Gilb: The Principles of Software Engineering Management, 1988

Confirmit

The Confirmit Case Study 2003-2009



Market
Research
& Feedback

MR

See paper on this case at www.gilb.com

Papers/Cases/Slides, Gilb Library,

value slide w... http://www.gilb.com/tiki-download_file.php?fileId=152

ppr wrong ag... http://www.gilb.com/tiki-download_file.php?fileId=50

Paper Firm http://www.gilb.com/tiki-download_file.php?fileId=32

And see papers (IEEE Software Fall 2006) by Geir K Hanssen, SINTEF

Their product = **confirmit**✓



Chief Storyteller = Trond Johansen

Customer Successes in Corporate Sector

Real Example of 1 of the 25 Quality Requirements

Usability.Productivity *(taken from Confirmit 8.5,*
performed a set of predefined steps, to produce a
standard MR Report.

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



Shift: from Function to Quality

- **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.**

Design Process

Design Suggestions



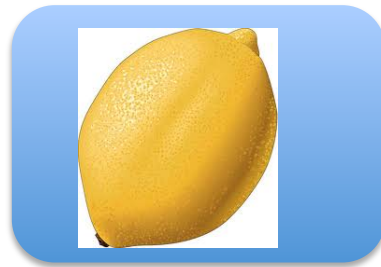
Design A



Design B



Design C



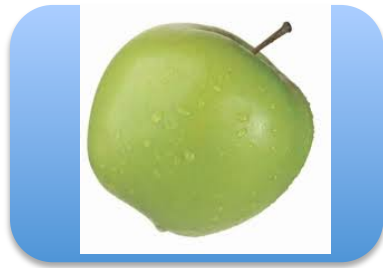
Design D

Impacts to Cost Evaluation

	A	B	C	D
Goal 1	30%	10%'	-10%	80%
Cost	10	50	1	20
G/C	3:1	1:5	?	4:1

Design Process: The winner

Design Suggestions



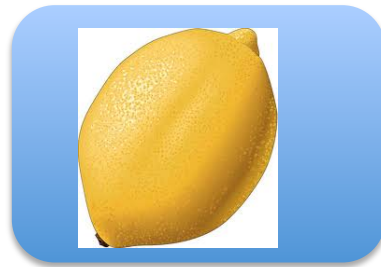
Design A



Design B



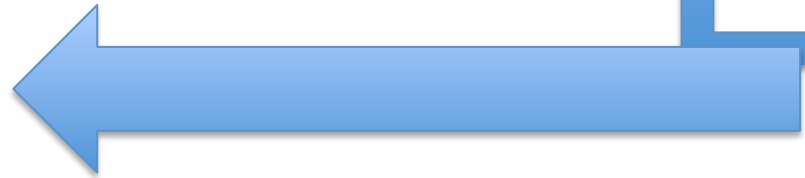
Design C



Design D

Impacts to Cost Evaluation

	A	B	C	D
Goal 1	30%	10%'	-10%	80%
Cost	10	50	1	20
G/C	3:1	1:5	?	4:1





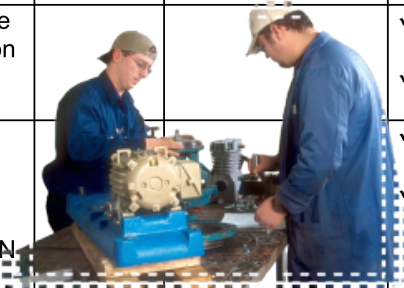
Trond Johansen

- 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	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

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

3

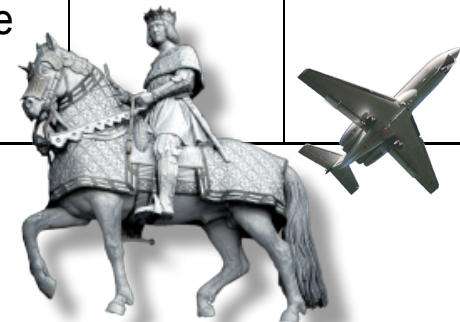
	Development Team	Users (PMT, Pros, Doc writer, other)	CTO (Sys Arch, Process Mgr)	QA (Configuration Manager & Test Manager)
Friday	<ul style="list-style-type: none"> ✓ PM: Send Version N detail plan to CTO + prior to Project Mgmt meeting ✓ PM: Attend Project Mgmt meeting: 12.00-15.00 ✓ Developers: Focus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



Evo's impact on Conformat product qualities 1st Qtr

- Only 5 highlights of the 25 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 Conformat 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

- **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**



confirmit✓

© Tom@Gilb.com
www.gilb.com



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 -

- **The method's positive impact on Confirmit 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.**



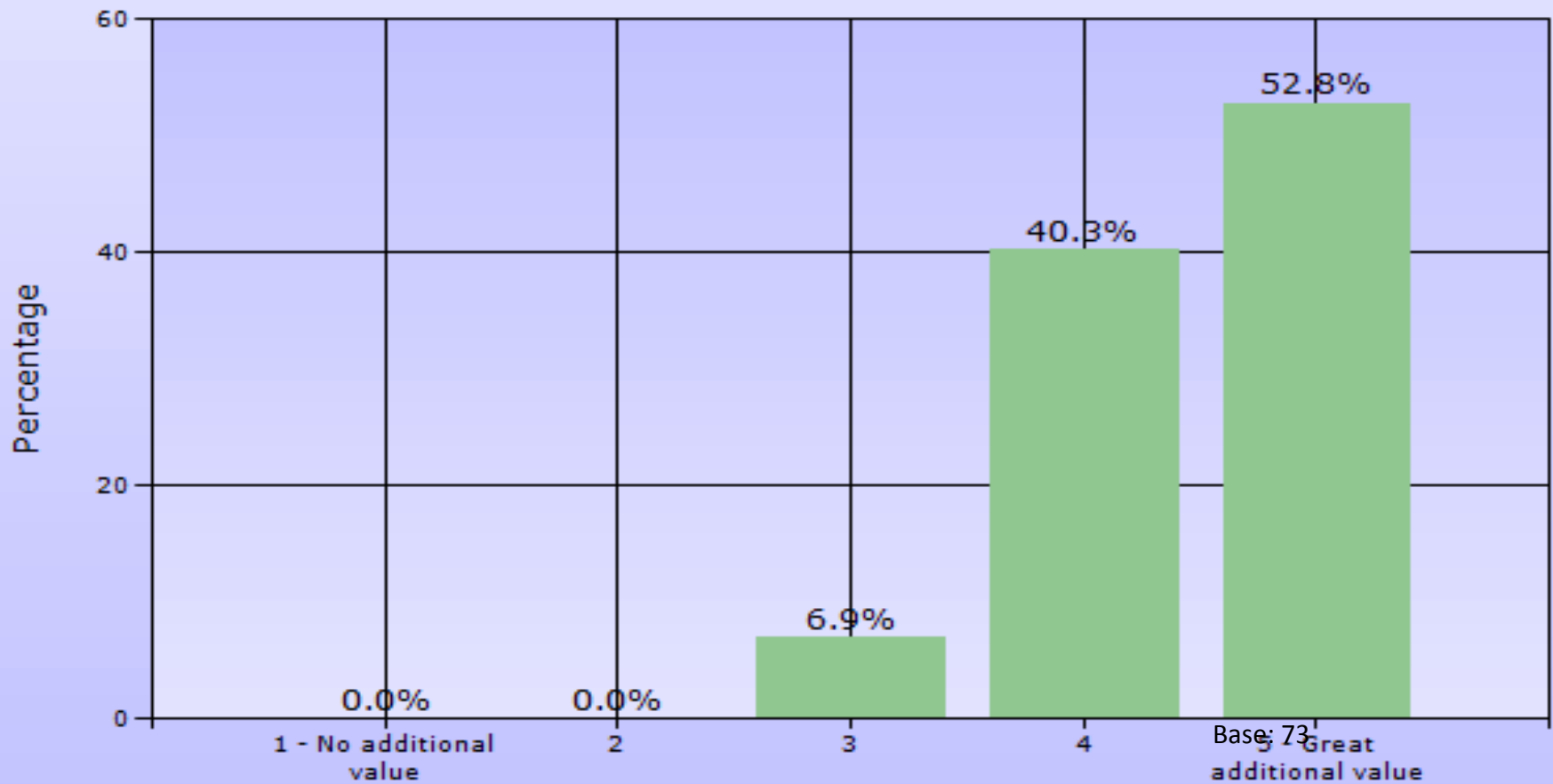
Trond Johansen

Initial Customer Feedback on the new Confrimit 9.0

November 24th, 2004

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

To what extent do you feel Conformat 9.0 will give you additional value?



Evo's impact on Confirmit 9.0 product qualities

Results from the second quarter of using Evo. 1/2

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 38%

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 83% and error tracking increased by 25%

Evo's impact on Conformat 9.0 product qualities

Results from the second quarter of using Evo. 2/2

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 1500%
Scalability	Ability to accomplish a bulk-update of X panelists within a timeframe of Z second	Number of panelists increased by 700%
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 1400%

Do you report such results in real released products in your second quarter of using a new lean or agile method like Scrum?

Code quality – "green" week

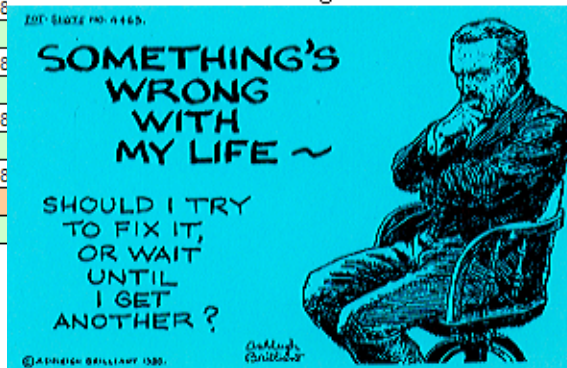
Confermit (2005) Norway

decided to design 'ease of change' in, to a legacy system, in one-week delivery-cycles, per month, using 'Evo' Agile
'Refactoring to reduce technical debt' -> Re-Engineering

- 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 (week 15)
	Units			Past	Tolerable	Goal	Estimated Impact	Actual Impact	Estimated Impact
	100,0	100,0	0	80	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	0	90	100				
PeerTests									
	100,0	100,0	0	90	100				100
FxCop									
	0,0	10,0	10	0	0				
TestDirectorTests									
	100,0	100,0	0	90	100				100
Robustness.Correctness									
	2,0	2,0	0	1	2		2	2	
Robustness.BoundaryConditions									
	0,0	0,0	0	8					
Speed									
	0,0	0,0	0	8					
ResourceUsage.CPU									
	100,0	0,0	100	8					
Maintainability.Doc.Code									
	100,0	100,0	0	8					
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

The Monthly 'Green Week'

User Week 1

Select a Goal

Brainstorm Designs

Estimate Design Impact/
Cost

Pick best design

Implement design

Test design

Update Progress to Goal

User Week 2

Select a Goal

Brainstorm Designs

Estimate Design Impact/
Cost

Pick best design

Implement design

Test design

Update Progress to Goal

User Week 3

Select a Goal

Brainstorm Designs

Estimate Design Impact/
Cost

Pick best design

Implement design

Test design

Update Progress to Goal

Developer Week 4

Select a Goal

Brainstorm Designs

Estimate Design Impact/
Cost

Pick best design

Implement design

Test design

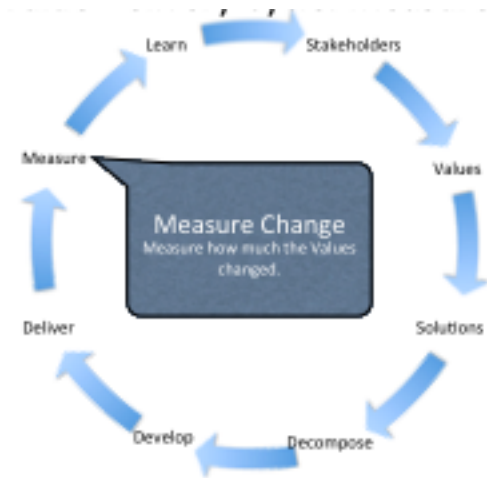
Update Progress to Goal

The 'Evo' start-up Week

- A 1 week 'feasibility study'
- Before starting real delivery of improved value to stakeholders

The Evo Startup Process a practical example

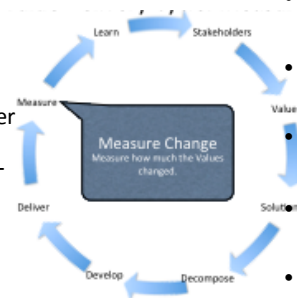
- The 'standards for Startup are at
 - Evo Startup Standard, Jan 12 2013
 - <http://www.gilb.com/dl562>



- Evo Project Management Standard, Jan 12 2013
- <http://www.gilb.com/dl563>

Startup Process Day 1 and 2

- Day 1: **Project Objectives:** The top few critical objectives quantified.
 - Objective: Determine, clarify, agree critical few project objectives – results – end states
 - **Process:**
 - Analyze current documentation and slides, for expressed or implied objectives (often implied by designs or lower level objectives)
 - Develop list of **Stakeholders** and their needs and values
 - Brainstorm 'top ten' critical objectives names list. Agree they are top critical few.
 - Detail definition in Planguage – meaning quantify and define clearly, unambiguously and in detail (a page)
 - Quality Control Objectives for Clarity: Major defect measurement. Exit if less than 1.0 majors per page
 - Quality Control Objectives for Relevance: Review against higher level objectives than project for alignment.
 - Define Constraints: resources, traditions, policies, corporate IT architecture, hidden assumptions.
 - Define Issues – yet unresolved
 - Note we might well choose to several things in *parallel*.
 - **Output:** A solid set of the top few critical *objectives* in quantified and measurable language. *Stakeholder* data specified.
 - **Participants:** anybody who is concerned with the business results, the higher the management level the better.
 - **End of Day Process:** meet 30 minutes with any responsible interested managers to present the outputs, and to get preliminary corrections and go-ahead.
 - **Note:** this process is so critical and can be time consuming, so if necessary it can spill over to next day. Perhaps in parallel with startup of the strategy identification. *Nothing is more critical or fundamental than doing this well.*



- Day 2: **Project Strategies and Architecture:** the top few critical strategies for reaching the critical objectives
 - **Objective:** to identify the top 'ten' most critical strategic decisions or architectures; the ones that will contribute or enable us most, to reach our primary objective goal levels on time.
 - **Process:**
 - Analysis of current documentation and slides to identify candidate strategies, implied or expressed.
 - Brainstorming of the 'names' of the specific strategy list, the top ten and a set of less powerful ideas (say 11-30)
 - Detail each top ten strategy sufficiently to understand impacts (on objectives, time and costs)
 - Specify, for each strategy all critical related information (like stakeholders, risks, assumptions, constraints, etc.)
 - Quality Control for clarity – correct unclear items. Exit based on defect level, or not.
 - Likely that work will need to be done in parallel in order to do ten strategies to a rich level of specification.
 - **Output:** A formal strategy specification, ready for evaluation, and decomposition and delivery of partial value results.
 - **Participants:** system architects, project architects, strategy planners. And members of the project team who will be in on the entire weeks process. The major input here is technical and organizational strategy (the means to reach the objectives)
 - **End of Day Process:** : meet 30 minutes with any responsible interested managers to present the outputs, and to get preliminary corrections and go-ahead.

Startup Process Day 3 and 4

Day 3: Evaluation of Strategies using Impact Estimation: our best estimates with experience and risk. How sure are of the major strategy decisions.

- **Objective:** to estimate to primary effects and all side effects of all top critical strategies on all top critical objectives, and on some resources (time, cost, effort). The estimates will be backed up by evidence, or their credibility will be rated low.
- **Process:**
 - Using the objectives and strategies developed on first 2 days as inputs
 - Populate an Impact Estimation table (aka Value Decision Table) with estimates of the expected result of deploying defined strategies. Estimate main intended impacts
 - And all side effects (on other core objectives)
 - And on all resources (time, money. Effort)
 - Estimate \pm ranges
 - Specify evidence and sources for estimates
 - Determine Credibility level
 - Quality Control the IE table against standards (Rules for IE in CE book), for possible 'exit' (meets standards)
 - Lots of parallel work needed and expected to do a good job.
- **Output:**
 - A fairly decent Impact Estimation table, possibly a several level set of them.
 - This will tell us if it is safe to proceed (we have good enough strategies)
 - And it will help us prioritize high value deliveries soon.
- **Participants:** architects, planners, anybody with strong views on any of the strategies. The team for the week.
- **Note:** *it might be necessary and desirable, now or later, to do this impact estimation process at 2 or 3 related levels (Business, Stakeholder, IT System) in order to see the Business-IT relationship clearly. This might exceed time limits and be done parallel or later.*
- **End of Day Process:** meet 30 minutes with any responsible interested managers to present the outputs, and to get preliminary corrections and go-ahead.

Day 4: Evolutionary Step Decomposition: what are the high value short term value delivery steps we can execute.

- **Objective:** to identify near team candidates for real value delivery to real stakeholders. What can we do for real next week!
- **Process:**
 - Identify highest value (to costs) strategies and sub-sets of strategies
 - Decompose into doable subsets in weekly to monthly cycles of result delivery
 - Plan the near steps (1 or more) in detail so that we are ready to execute the step in practice.
 - Who does it, main responsible, team.
 - Expected measurable results and costs
 - Stakeholder involved in receiving
 - Test process (for value)
- **Output:** 1 or more potential steps for value delivery to some stakeholders, a plan good enough to approve and execute in practice.
- **Participants:** Project Management, architects prepared to decompose architecture in practice. The weeks team for this start up study.
- **End of Day Process:** meet 30 minutes with any responsible interested managers to present the outputs, and to get preliminary corrections and go-ahead.



Day 5

- Boss approves doing the next week

111111

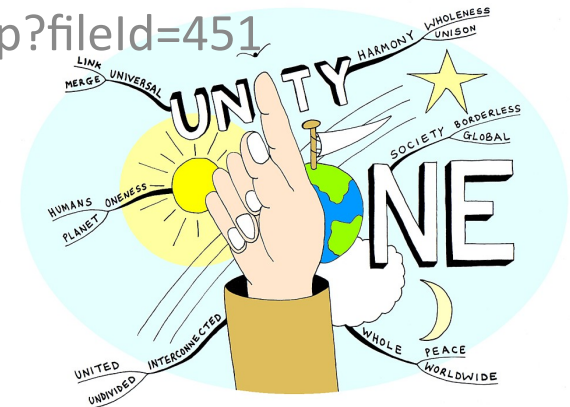
The Unity Method **111111**

**for decomposition into
iterative value delivery
steps**

By Tom@Gilb.com

Slides at www.gilb.com/downloads

http://www.gilb.com/tiki-download_file.php?fileId=451



'1' 4 U2



One

Bono U2



Is it getting better?

Or do you feel the same?

**Will it make it easier on
you now?**

You got someone to blame

You say, one love, one life

**When it's one need in the
night**

**One love, we get to share
it**

Leaves you baby if you don't care for it

'One' lyrics

One love, one blood

**One life, you got to do what
you should**

**One life, with each other
Sisters, brothers**

**One life but we're not the
same**

**We get to carry each other,
carry each other**

One

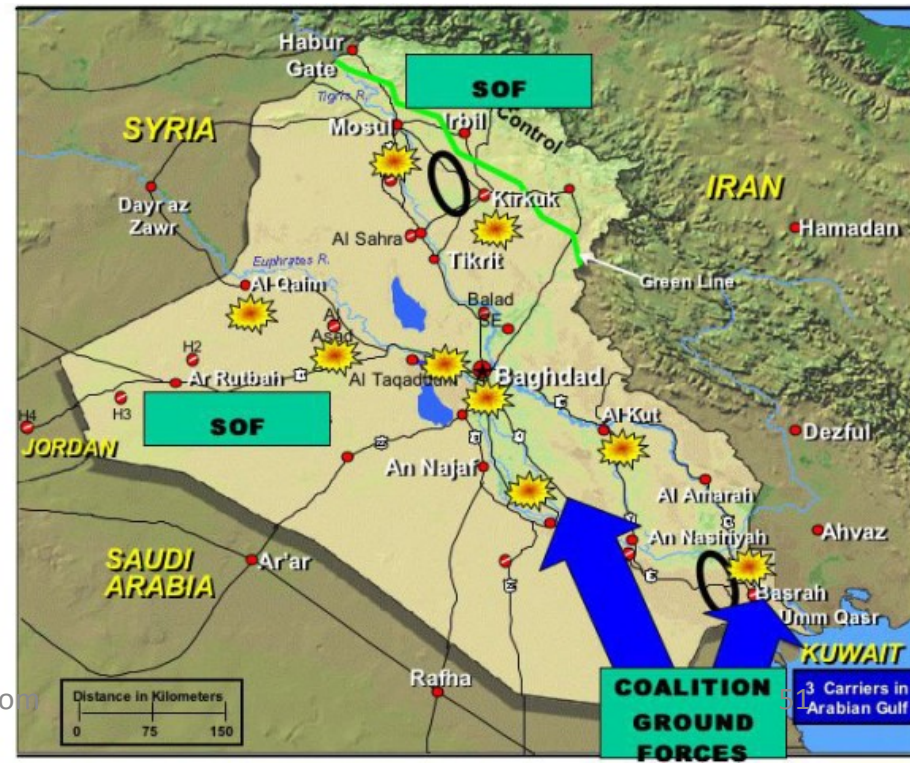
One

© POLYGRAM INT. MUSIC PUBL. B.V.;

A True War Story

111111 in practice

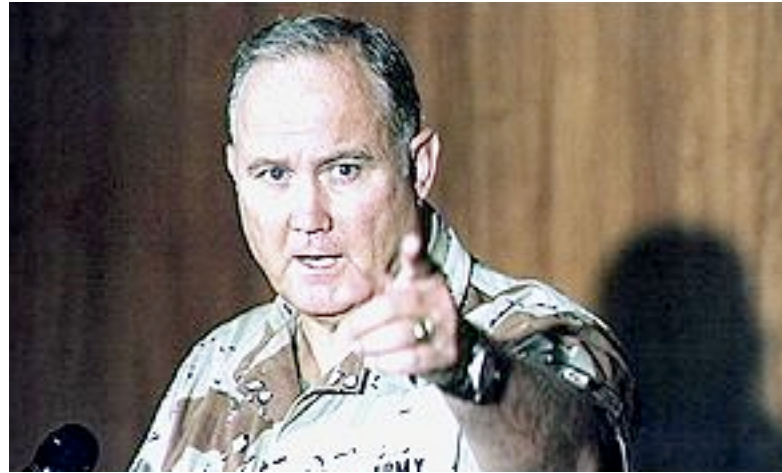
- How we found a value delivery step 'next week'
 - a week of value delivery beat 11 years of waterfall method



The *Persinscom IT System* Case



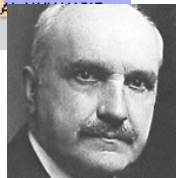
**Commanding General
Norman Schwarzkopf**



‘Stormin’ Norman’



He who does not learn from history
Is doomed to repeat it



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



-

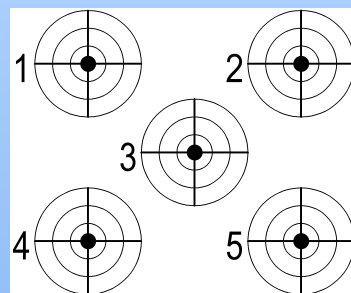
US Army Example: PERSINSCOM

[illegible]

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: PERSINSCOM: 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.

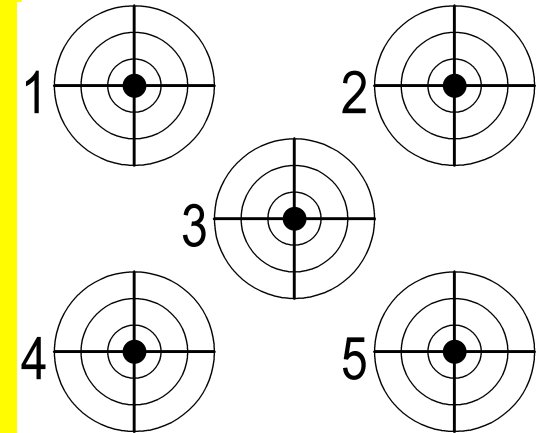
Meter: Log of Violations.

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

Record [NARDAC] 0 ? ← NARDAC Reports Last Year

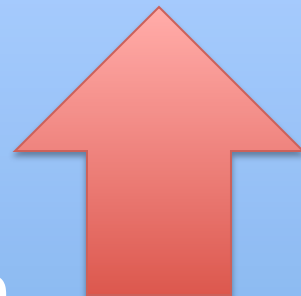

Fail : <must be better than Past, Unknown number> ← CG

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><div>Tuesday</div><div>The Top Ten Critical Strategies For reaching the ←objectives Were decided</div><div></div><div></div></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: Sanity Check

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%
SUM IMPACT FOR EACH SOLUTION	482%	280%	305%	390%	315%	649%	
Money % of total budget	15%	4%	3%	4%	6%	4%	
Time % total work months/year	15%	15%	20%	10%	20%	18%	
SUM RESOURCES	30	19	23	14	26	22	
BENEFIT/RESOURCES RATIO	16:1	14:7	13:3	27:9	12:1	29:5	



MEASURING HAND FOR GLOVE SIZE

US DoD. Persinscom **Impact EstimationTable:**

Designs

<i>Design Ideas -></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>
Requirements	50%	100%	5%	5%	5%	60%	185%
Availability 90% <-> 99.5% Up time	50%		5-10%	0%	0%	200%	265%
Usability 200 <-> 60 Requests by Users			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%						303%
Morale 72 <-> 60 per month on Sick Leave	50%						251%
Data Integrity 88% <-> 97% Data Error %	42%						177%
Technology Adaptability 75% Adapt Technology	5%						160%
Requirement Adaptability ? <-> 2.6% Adapt to Change	80%						260%
Resource Adaptability 2.1M <-> ? Resource Change	10%						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 :1</i>	

**Estimated Impact
of
Design
-> Requirements**

US Army Example: PERSINSCOM: Personnel System



STRATEGIES →	Technology Investment	Business Practices	People	Empow- erment	<i>Principles of IMA Management</i>	Business Process Re- engineering	SUM
OBJECTIVES							
Customer Service ? → 0 Violation of agreement	50%	10%	5%	5%	5%	60%	185%
Availability 90% → 99.5% Up time	50%	5%	5-10%	0	0	200%	265%
Usability 200 → 60 Requests by Users	50%	5-10%	5-10%	50%	0	10%	130%
Responsiveness 70% → ECP's on time	50%	10%	90%	25%	5%	50%	180%
Productivity 3:1 Return on Investment	45%	60%	10%	35%	100%	53%	303%
Morale 72 → 60 per mo. Sick Leave	50%	5%	75%	45%	15%	61%	251%
Data Integrity 88% → 97% Data Error %	42%	10%	25%	5%	70%	25%	177%
Technology Adaptability 75% Adapt Technology	5%	30%	5%	60%	0	60%	160%
Requirement Adaptability ? → 2.6% Adapt to Change	80%	20%	60%	75%	20%	5%	260%
Resource Adaptability 2.1M → ? Resource Change	10%	80%	5%	50%	50%	75%	270%
Cost Reduction FADS → 30% Total Funding	50%	40%	10%	40%	50%	50%	240%
SUM IMPACT FOR EACH SOLUTION	482%	280%	305%	390%	315%	649%	
Money % of total budget	15%	4%	3%	4%	6%	4%	
Time % total work months/year	15%	15%	20%	10%	20%	18%	
SUM RESOURCES	30	19	23	14	26	22	
BENEFIT/RESOURCES RATIO	16:1	14:7	13:3	27:9	12:1	29.5 :1	



Impact Estimation: Value-for-Money Delivery Table

STRATEGIES → OBJECTIVES	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%
SUM IMPACT FOR EACH SOLUTION	482%	280%	305%	390%	315%	649%	
Money % of total budget	15%	4%	3%	4%	6%	4%	
Time % total work months/year	15%	15%	20%	10%	20%	18%	
SUM RESOURCES	30	19	23	14	26	22	
BENEFIT/RESOURCES RATIO	16:1	14:7	13:3	27:9	12:1	29.5 : 1	

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 1 Unity**
 - **1% increase at least**
 - **1 stakeholder**
 - **1 quality/value**
 - **1 week delivery cycle**
 - **1 function focus**
 - **1 design used**

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%
SUM IMPACT FOR EACH SOLUTION	482%	280%	305%	390%	315%	649%	
Money % of total budget	15%	4%	3%	4%	6%	4%	
Time % total work months/year	15%	15%	20%	10%	20%	18%	
SUM RESOURCES	30	19	23	14	26	22	
BENEFIT/RESOURCES RATIO	16:1	14:7	13:3	27:9	12:1	29:5	



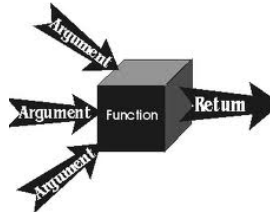
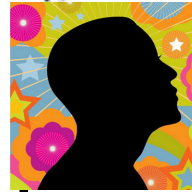
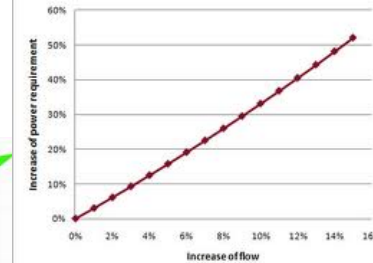
Next weeks Evo Step??

- **“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.**
- **Can you deliver it next week?**
 - **Its already done: If General, move to head of queue’**



1 1 1 1 1 1 Unity

- 1% increase at least
- 1 stakeholder
- 1 quality or value
- 1-week delivery cycle
- 1 function focus
- 1 design used



"I kill men for a living! (General Pellicci)



UNITED STATES ARMY
PERSONNEL INFORMATION
SYSTEMS COMMAND
CERTIFICATE of APPRECIATION

is awarded to

MR. TOM GILB

for

SELFLESS AND DEDICATED SERVICE IN SUPPORT OF THE PERSONNEL INFORMATION SYSTEMS COMMAND. AS A MANAGEMENT CONSULTANT IN RESULT DELIVERY PLANNING, HIS PATRIOTISM, PROFESSIONAL COMPETENCE AND PERSONAL SACRIFICES ARE HIGHLY COMMENDABLE. TOM GILB'S DEDICATION AND THE EXCEPTIONAL MANNER IN WHICH HE PERFORMED HIS DUTIES HAD A DIRECT AND SIGNIFICANT IMPACT ON PERSINSCOM'S MISSION. HIS OUTSTANDING CONTRIBUTIONS AND DISTINGUISHED SERVICE REFLECT GREAT CREDIT ON HIM AND THE UNITED STATES ARMY. CONGRATULATIONS FOR A JOB WELL DONE.

30 AUGUST 1991

Personnel Information Systems Command

Jack A. Pellicci
JACK A. PELLICCI
Brigadier General, USA
Commanding

Decomposition Principles A Teachable Discipline

Decomposition of Projects into small steps 11/12/2008 13:38

Decomposition of Projects: How to design small, early and frequent incremental and evolutionary feedback, stakeholder result delivery steps, at the level of 2% of project resources.

By Tom Gilb, Norway

Introduction

- The basic premise of iterative, incremental and evolutionary project management [Larman 03 MG] is that a project is divided into early, frequent and short duration delivery steps.
- One basic premise of these methods is that each step will attempt to deliver some real value to stakeholders.
- It is not difficult to envisage steps of *construction* for a system; the difficulty is when a step has to *deliver* something of *value* to *stakeholders*, in particular to end users.
- This paper will give some teachable guidelines, policies and principles for decomposition. It will also give short examples from practical experience.

A Policy for Evo Planning

One way of guiding Evo planners is by means of a 'policy'. A general policy looks like this (you can modify the policy parameters to your local needs):

Evo Planning Policy (example)

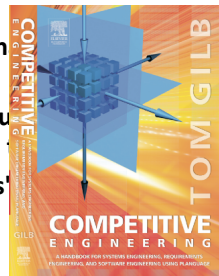
P1: Steps will be sequenced on the basis of their overall benefit-to-cost efficiency.

P2: No step may normally exceed 2% of total project financial budget.

How to decompose systems into small evolutionary steps:

some principles to apply:

- 1• *Believe* there is a way to do it, you just have not *found* it yet!
- 2• *Identify* obstacles, but don't use them as excuses: use your imagination to get *rid* of them!
- 3• Focus on *some usefulness* for the user or customer, however small.
- 4• Do not focus on the design ideas themselves, they are distracting, especially for small initial cycles. Sometimes you have to ignore them entirely in the short term!
- 5• Think; one customer, tomorrow, one interesting improvement.
- 6• Focus on the *results* (which you should have defined in your goals, moving toward target levels).
- 7• Don't be afraid to use temporary-scaffolding designs. Their cost must be seen in the light of the value of making some progress, and getting practical experience.
- 8• Don't be worried that your design is inelegant; it is results that count.
- 9• Don't be afraid that the customer won't like it. *If you are focusing on what they want*, then by definition, *they* should like it. If you are not, they won't.
- 10• Don't get so worried about "what might happen afterwards" that you make no practical progress.
- 11• You cannot foresee everything. Don't even *think* about it!
- 12• If you focus on helping your customer in practice, *now*, where they *really* need it, you will be forgiven a lot of 'sins'!
- 13• You can understand things much better, by getting *some* practical experience (and removing *some* of your fears).
- 14• Do *early* cycles, on willing local mature parts of your user community.
- 15• When some cycles, like a purchase-order cycle, take a long time, initiate them early, and do other useful cycles while you wait.
- 16• If something seems to need to wait for 'the big new system', ask if you cannot usefully do it with the 'awful old system', so as to pilot it realistically, and perhaps alleviate some 'pain' in the old system.
- 17• If something seems too costly to buy, for limited initial use, see if you can negotiate some kind of 'pay as you really use' contract. Most suppliers would like to do this to get your patronage, and to avoid competitors making the same deal.
- 18• If *you* can't think of some useful small cycles, then talk directly with the real 'customer' or end user. They probably have dozens of suggestions.
- 19• Talk with end users in *any* case, they have insights you need.
- 20• Don't be afraid to use the old system and the old 'culture' as a launching platform for the radical new system. There is a lot of merit in this, and many



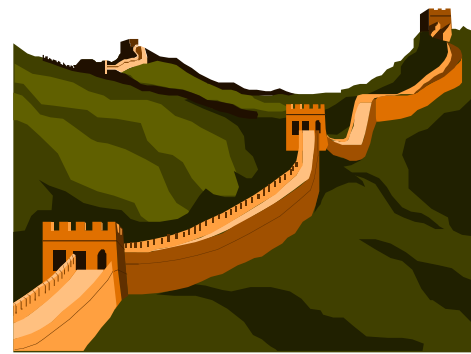
Rene Descartes on Focus

- **“We should bring the whole force of our minds**
 - to bear upon the most minute and simple details**
 - and to dwell upon them for a long time**
 - so that we become accustomed to perceive the truth clearly and distinctly.”**
- **Rene Descartes, Rules for the Direction of the Mind, 1628**

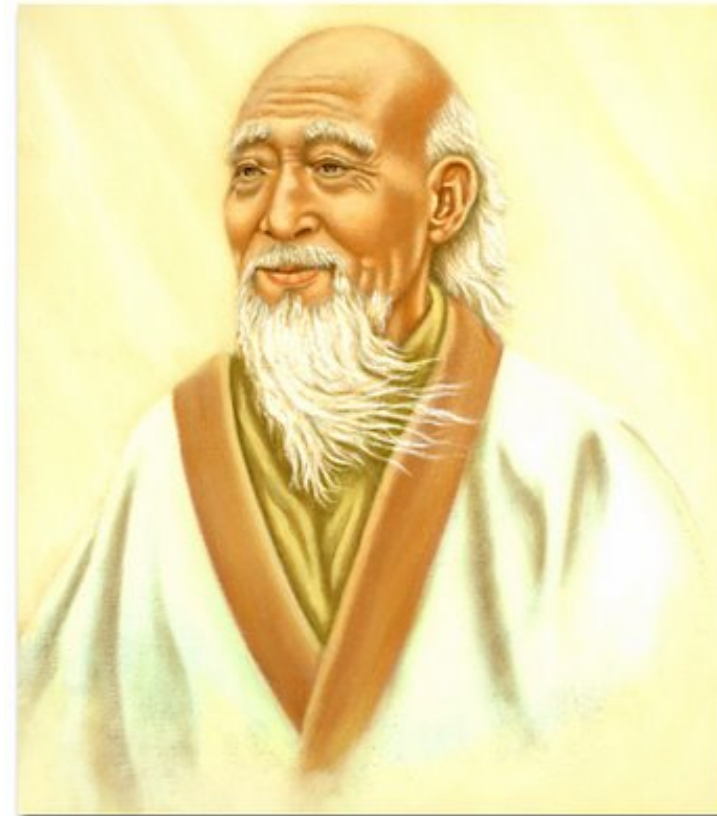




Tao Te Ching (500BC)



- **That which remains quiet, is easy to handle.**
- **That which is not yet developed is easy to manage.**
- **That which is weak is easy to control.**
- **That which is still small is easy to direct.**
- **Deal with little troubles before they become big.**
- **Attend to little problems before they get out of hand.**
 - **For the largest tree was once a sprout,**
- **the tallest tower started with the first brick,**
- **and the longest journey started with the first step.**



– From Lao Tzu in Bahn, 1980 (also quoted in Gilb, Principles of Software Engineering Management page 96), Penguin book

6 October 2013

© Gilb.com

68

“Evo” Project Management

- Evo is short for Evolutionary
- But
- We have played with acronyms
- Evolutionary
- Value
- Optimization

Summary:

- A recent London Times survey report indicated that only 13% of 1500 surveyed IT projects were 'successful' [Times].
- Other reports ([Standish], Chaos) indicate that about half of the surveyed projects were considered total failures,
 - the same percentage as US Department of Defense estimated its software projects failed.
- We must be doing something very wrong.
- What can the IT Manager and IT Project Manager do about this situation in practice?
- Some people recommend complex development process standards such as CMM, CMMI, SPICE and their like.
 - I am not convinced that these are good medicine for even very large systems engineering projects,
 - and certainly they are overly complex for most IT projects in Europe.
- Some people recommend agile and extreme programming methods –
 - these are closer to my heart –
 - but maybe, for non-trivial projects -
 - they are 'too simple' ?
- I will offer you my advice in the form of a short simple defined process.
- My main addition to the agile concepts is that I believe they need to focus on the top few critical stakeholder objectives.
- These top objectives need to be quantified and measurable in practice.
- This simple quantification device is missing from most methods,
 - but I believe that quantified management a necessary minimum to control all but the smallest upgrade efforts.

The Simplest and Best Agile Project Method

- Background:
- A number of ‘agile’ methods have appeared, trying to simplify project management and systems implementation.
- They have all missed the central point,
 - namely evolutionary project management (Evo),
 - using **quantified feedback about central goals and budgets**
 - which would allow them complete freedom to simplify, and to succeed.
 - Here is my suggestion for ultimate agility.

The Simplest and Best Agile Project Method (1 pg Σ)

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 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.

“The End”.

- **That is the end of this slides. You need read no more. But I can write an ‘appendix’ , in case anyone would like more detail! Here it is.**

APPENDIX!

- **If we re running short of time on the talk, we will skip this**
- **But we include the slides for you to study later**

References in Notes of this
slide

I will comment on the process
definition, statement by statement.

‘The Simplest and Best Agile Project Method’



- **The Gilb Agile Process (“GAP” of course) is ‘simplest method because of its sharp focus at a ‘high level’, on the ‘end results’.**
- **This allows us to avoid distracting management attention**
 - **with the supporting processes, designs and requirements, needed to deliver the results.**
- **The supporting processes, designs, and requirements do need to exist of course,**
 - **but our GAP process is neutral,**
 - **and in fact encourages competition and selection**
 - **of the fittest supporting processes at any step.**
- **This is essentially different from making user-driven lists of functions to program into the system – typical of conventional Agile methods.**
 - **It focuses on the main outcome, for example high security, ease of use, or flexibility.**
- **GAP is the ‘Best’ Agile process because it focuses on *numerically defined* and tracked critical business or technical goals of a project.**
 - **This numeric focus is in sharp contrast to the non-numeric ‘yellow sticky’ mentality of Conventional Routine Agile Processes.**



Richard Smith

“ I attended a 3-day course with you and Kai whilst at Citigroup in 2006”

Previous PM Methods:
No 'Value delivery tracking'.
No change reaction ability



Richard Smith

- “However, (**our old** project management methodology) main failings were that
- it almost **totally missed the ability to track delivery of actual *value* improvements to a project's stakeholders,**
- and **the ability to react to changes**
 - in requirements and
 - priority
 - for the project's duration”



We only had the illusion of control.
But little help to testers and analysts



Richard Smith

- “The (old) toolset generated lots of charts and stats
- that provided **the illusion of risk control**.
- But actually provided very little help to the analysts, developers and testers actually doing the work at the coal face.”



The proof is in the pudding;



Richard Smith

- “The proof is in the pudding;
- I have **used Evo**
 - *(albeit in disguise sometimes)*
 - on two large, high-risk projects in front-office investment banking businesses,
 - and several smaller tasks. “



Experience: if top level requirements are *separated* from design, the 'requirements' are **stable**!



Richard Smith

- “On the largest critical project,
- the original ***business functions & performance objective*** requirements document,
- ***which included no design***,
- essentially remained ***unchanged***
- over the **14 months** the project took to deliver,....”



Dynamic (Agile, Evo) design testing: not unlike 'Lean Startup'



Richard Smith

- "... but **the detailed designs**
 - (of the GUI, business logic, performance characteristics)
- **changed many many times,**
 - guided by lessons learnt
 - and **feedback** gained by
 - delivering a succession of early deliveries
 - to real users"

"I attended a 3-day course with you and Kai whilst at Citigroup in 2006", Richard Smith

It looks like the stakeholders liked the top level system qualities, on first try



Richard Smith

- “ In the end, the new system responsible for 10s of USD billions of notional risk,
- **successfully went live**
- **over one weekend**
- **for 800 users worldwide,**
- and **was seen as a big success**
- **by the sponsoring stakeholders.”**

“ I attended a 3-day course with you and Kai whilst at Citigroup in 2006” , Richard Smith

‘Evo’ Process Description

- *<http://www.gilb.com/dl487>*
- *The Evo ‘Standard’ Process Description*
- *Evo Chapter in CE Book*
- *http://www.gilb.com/tiki-download_file.php?fileId=77*



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).*

- **Projects need to learn to focus on *all stakeholders* that arguably can affect the success or failure.**
- **The *needs* of these stakeholders must be *determined* – by any useful methods – and *converted* into project *requirements*.**
- **By contrast the Conventional Agile Model**
 - focuses on a User/Customer (‘in the next room’).
 - Good enough if they *were* the only stakeholder.
 - But disastrous for most real projects,
 - where the critical stakeholders are more varied in type and number.
- **Conventional Agile processes, due to this dangerously narrow requirements focus, risk outright failure,**
 - even if the ‘Customer’ gets all *their* needs fulfilled.

Real Bank Project : Project Progress Testability

Quantification of the most-critical project objectives on day 1

ONE PAGE PROJECT REQUIREMENTS QUANTIFIED

<p><u>Speed-To-Deliver:</u> Scale: average Calendar days needed from New Idea Approved until Idea Operational, for given Tasks, on given Markets.</p> <p>Past [2009, Market = EURex, Task =Bond Execution] 2-3 months ?</p> <p>Goal [Deadline =End 20xz, Market = EURex, Task =Bond Execution] 5 days</p>	<p><u>Front-Office-Trade-Management-Efficiency</u>Scale: Time from <u>Ticket Launch</u> to trade updating real-time risk view</p> <p>Past [20xx, Function = Risk Mgt, Region = Global] ~ 80s +/- 45s ??</p> <p>Goal [End 20xz, Function = Risk Mgt, Region = Global] ~ 50% better?</p> <p>Managing Risk – Accurate – Consolidated – Real Time</p>
--	--

<p>Operational-Control: Scale: % of trades per day, where the calculated economic difference between OUR CO and Marketplace/Clients, is less than “1 Yen”(or equivalent).</p> <p>Past [April 20xx] 10% change this to 90% NH Goal [Dec. 20xy] 100%</p>	<p>Risk.Cross-Product Scale: % of financial products that risk metrics can be displayed in a single position blotter in a way appropriate for the trader (i.e. – around a benchmark vs. across the curve).</p> <p>Past [April 20xx] 0% 95%. Goal [Dec. 20xy] 100%</p>
---	--

<p><u>Operational-Control.Consistent:</u> Scale: % of defined [Trades] failing full metrics is delayed by more than 0.5 sec. Past [April 20xx, NA] 1% Past STP across the transaction cycle. Past [April 20xx, Trades=Voice Trades] [April 20xx, EMEA] ???% Past [April 20xx, AP] 100% Goal [Dec. 20xy] 0%</p> <p>95%</p> <p>Past [April 20xx, Trades=eTrades] 93%</p> <p>Goal [April 20xz, Trades=Voice Trades] <95 ± 2%></p> <p>Goal [April 20xz, Trades=eTrades] 98.5 ± 0.5 %</p>	<p>Risk.Accuracy</p> <p><u>Risk. user-configurable</u> Scale: ??? pretty binary – feature is there or not – how do we represent?</p> <p>Past [April 20xy] 1% Goal [Dec. 20xy] 0%</p>
---	---

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

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

ONE PAGE PROJECT REQUIREMENTS QUANTIFIED

Operational-Control:

Scale: % of trades per day, where the calculated economic difference between OUR CO and Marketplace/Clients, is less than “1 Yen”(or equivalent).

Past [April 20xx] 10%

Goal [Dec. 20xy] 100%

P&L-Consistent
Actual (T+1) s

Speed-To-

Approved

Past [2009

Goal [Dead

days

Operation

economic c

than “1 Ye

Past [April

Operation

STP across

95%

Past [April

Goal [April

Goal [April

Operation

times, per

defined [B

Past [April

Run=Overnight] <0.5> Past [April 20xx, Batch-Run= T+1] 1 Goal [Dec.

20xy, Batch-Run=End-Of-Day, Delay<1hour] 1

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

Goal (EOY 20xy, cost type = I 2 – REGION = ALL) Reduce cost by x %

Goal (EOY 20xy, cost type = E 1 – REGION = ALL) Reduce cost by x %

Goal (EOY 20xy, cost type = E 2 – REGION = ALL) Reduce cost by 100%

Goal (EOY 20xy, cost type = E 3 – REGION = ALL) Reduce cost by x %

*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.*

- **In the Gilb Agile Process, the project is initially defined in terms of clearly stated, quantified, critical objectives.**
- **During the project, these long-term (Project completion term)**
 - **objectives can be changed, and tuned,**
 - **based on practical experience and feedback,**
 - **from each Evo step.**
 - **They are not cast in concrete, even though they are extremely clear.**
- **Conventional Agile methods do not have any such quantification concept.**
- **Conventional vague ideas, un-measurable, un-testable, un-quantified, and un-deadlined requirements, do not count as true long term goals, in our view.**

Acer: VERY TOP LEVEL PROJECT STRATEGIES

Note: These very top level project strategies specify how we are going to achieve the top level project goals.

Identify Binding Compliance Requirements Strategy:

Gist: Identify all officially binding security administration requirements with which we must become compliant both from THE CORP and Regulatory Authorities.

How much do these strategies cost?

System Control Strategy:

Gist: a formal system or process we can use to decide what characteristics a [system; default = application] has with regard to our compliance, performance, availability and cost goals

Note: *an inspection process, for instance*

Define and implement inspection for security administration-related business requirements specifications

Define and implement inspection for [systems; default = applications] which already exist in CitiTech environments

Note: *systems include applications, databases, data service and machines. Project ACER ought to be extensible.*

System Implementation Strategy:

Gist: a formal system or process we can use to actually change a [system; default = application] so that it meets our compliance, performance, availability and cost goals

All systems ought to feed EERS

Publish best practices for developing security administration requirement specifications

Publish a security administration requirement specification template

Application technology managers are service providers in the formal change process, that are required to meet all project goals within defined timescales

How much impact on our 4 Goals do these strategies have?

Find Services That Meet Our Goals Strategy:

Gist: a formal system or process we can use to evaluate security administration services offered by internal and external services providers so that we can meet our defined goals

Note: *this strategy avoids pre-supposition that one solution is the only option (EG all applications must migrate to RSA and that RSA is the only security administration services offering)*

Use The Lowest Cost Provider Strategy:

Gist: use the services provider that meets all signed-off project goals for the lowest \$US cost.

Note: *if all project goals can be met by more than one services provider, the provider offering the lowest \$US cost for meeting the goals and no more than the goals ought to be used*

3. Define approximately 4 budgets for your most limited resources (for example, time, people, money, and equipment).

- **Conventional methods**
 - do not seem to directly, and in detail, manage the array of limited resources we have.
 - But admittedly there are some such devices in place in the Conventional Agile methods,
 - such as the incremental weekly (or so) development cycle.
- **the Evo method sets an explicit numeric budget for any useful set of limited resources – but it does not stop there!**
- **Our Evo cycles will both**
 - estimate,
 - record actual resource use,
 - and analyze the deviation, on *every* Evo cycle,
 - in order to understand and control the economics of the project –
 - concurrently with the performance characteristics.
- **This is the essential distinction between incremental and evolutionary development methods.**
- **Implementation Money**
- **Implementation Time**
- **Operational Cost**
- **Operational Work Hours**

4. Write up these plans for the goals and budgets (Try to ensure this is kept to only one page).

- all these key quantified performance targets, and resource budgets, are presented simultaneously on a single overview page.
- additional detail about them can, of course, be captured off of this one 'focus' page.
- this set of top level objectives is not frozen.
- It can be updated as the result of
 - both internal Evolutionary (Evo) step learning,
 - or of external pressures and insights.

Objectives

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

Acer Project: Impact Estimation Table

Objectives

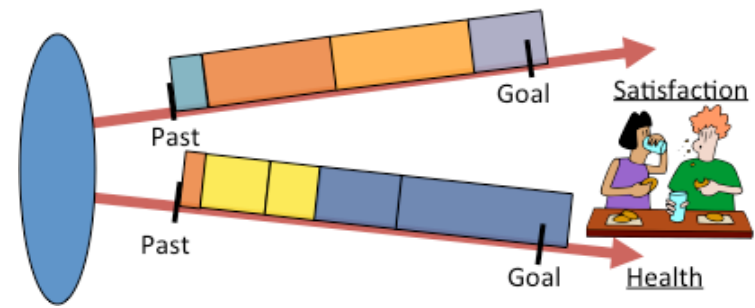
Strategies	Identify Binding Compliance Requirements Strategy	System Control Strategy	System Implementation	Find Services That Meet Our Strategy	Use The Lowest Cost Provider Strategy
Goals		Strategies			
Security Administration Compliance 25% → 90%	100%	100%	100%	50%	0%
Security Administration Performance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

5. *Negotiate with the key stakeholders to formally agree the goals and budgets.*

- once the objectives, the version derived from our *developer's* understanding of stakeholder needs, are clearly articulated –
 - we need to go back to the *real stakeholders*
 - and check that they agree with our 'clear' (but potentially incorrect or outdated) interpretation.
- it is certainly a wise precaution to *check back later*,
 - during the project evolution,
 - with the *specific stakeholders*
 - that will be *impacted* with a *particular Evo step*,
 - as to how they feel about a particular choice of step content (design) -
(that impacts the performance and cost aspect estimates):
 - are estimates realistic in the real implementation environment?,
 - and to check for any new insights regarding the long term objectives.

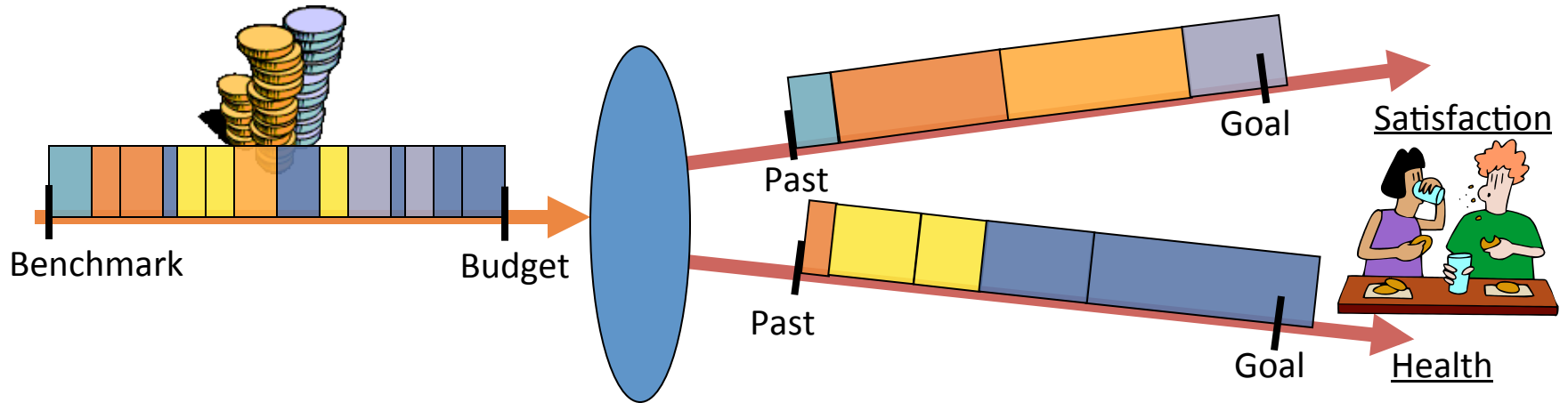


6. Plan to deliver some benefit
(that is, 'progress towards the goals')
in weekly (or shorter) increments (Evo steps).

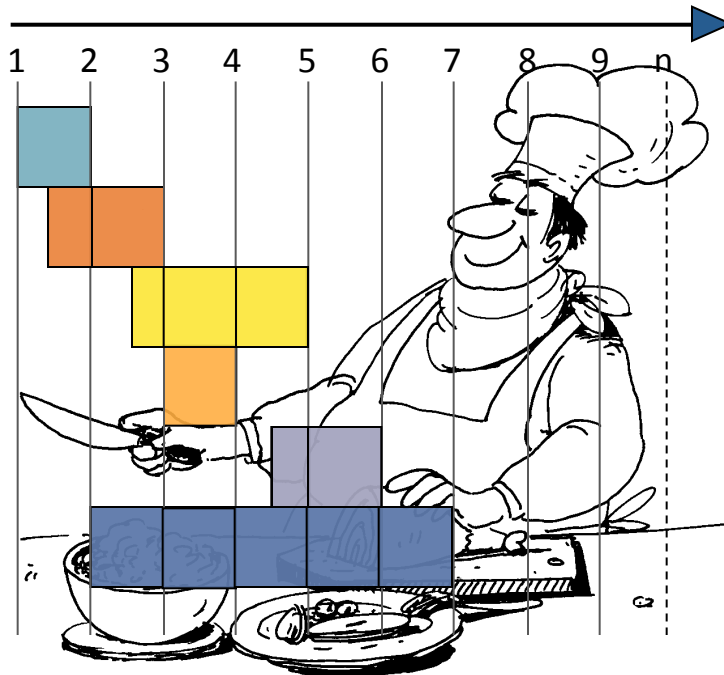


- the weekly delivery cycle is adopted by Conventional Agile methods – good.
- but the notion of *measurement*, on multiple performance and resource objectives, is absent.
- the Conventional notion of *agreeing with a user*, about function to be built, during that weekly cycle is healthy, but
 - the Evo method is focused on
 - systematic, weekly cycle, measured delivery
 - towards long-range higher-level objectives, w
 - ithin numeric, multiple, resource-constraints.
- this means that the Evo method is more clearly focused on
 - the wider stakeholder set values,
 - and on the total resource cost management.
- the Evo method is NOT focused on system 'construction' ('we are programmers, therefore we write code').
- the Evo method is focused on delivering useful results from an organically whole system.
 - This means that we are not focused on 'writing code' .
we reuse, buy, or exploit existing code just as happily as to write our own.
 - We build databases, train and motivate users, improve hardware, telecommunications, websites, improve working environment, improve motivation.
 - So we become more like systems engineers ('any technology to deliver the results!'), than programmers ('what can we code for you today?').

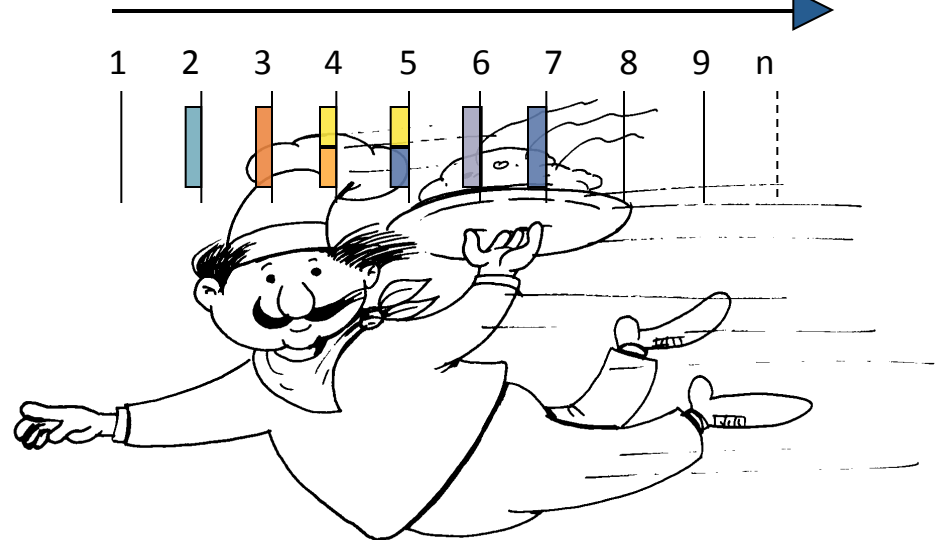
Costs / Effects



Back-room Design Development



Front-room Evolutionary Delivery



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.

- **All agile methods agree that the development needs to be done in short, frequent, delivery cycles.**
- **the Evo method, specifically insists that the closed loop control of each cycle is:**
 - **done by numeric pre-cycle estimates,**
 - **end-cycle measurements,**
 - **analysis of deviation from estimates,**
 - **and appropriate change to immediate planned cycles,**
 - **to estimates,**
 - **and to stakeholder expectation management**
 - (‘this is going to late, if we don’ t do X’).

Value Delivery Cycle: Measure

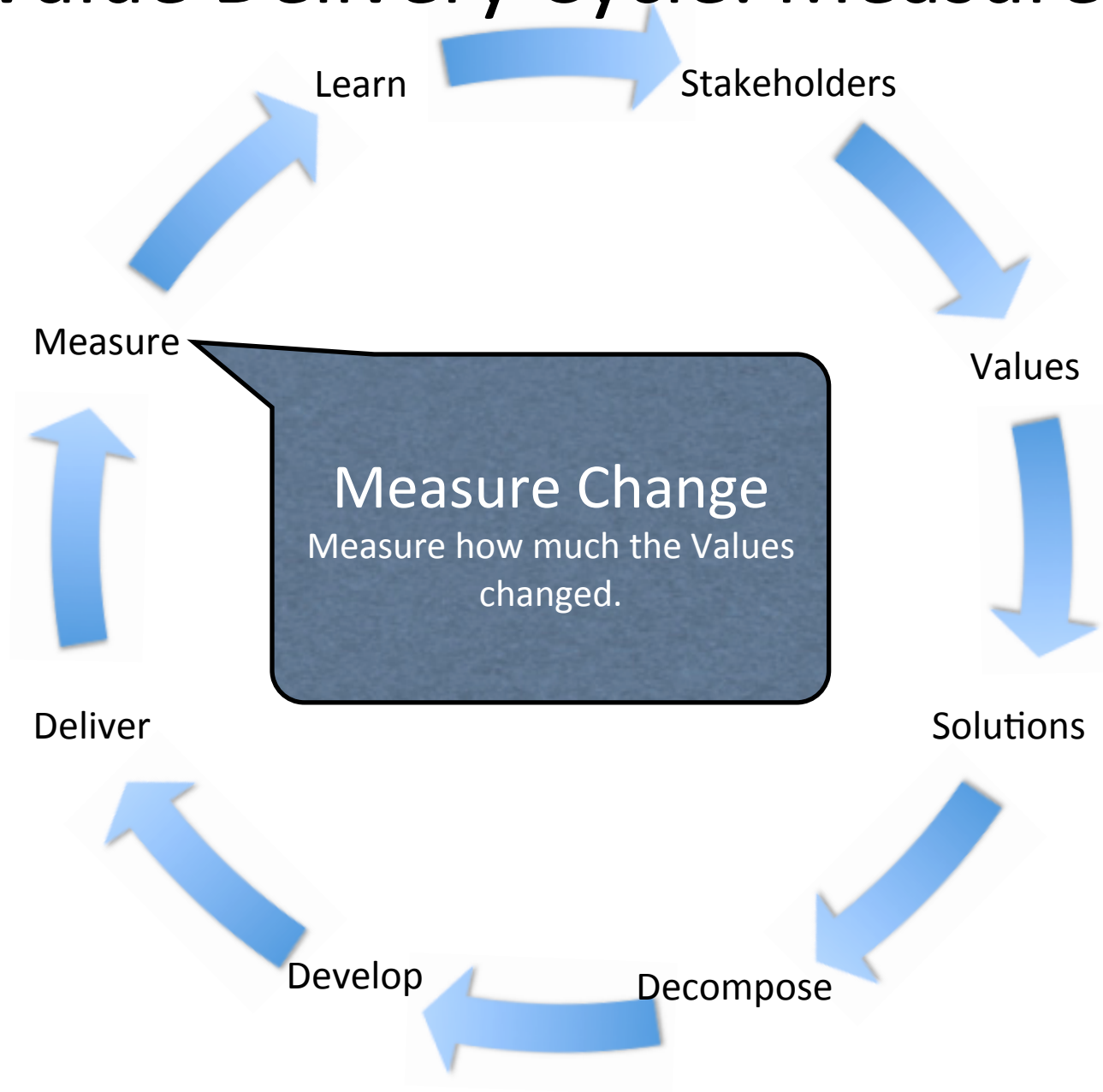


Figure: the use of an Impact Estimation table [CE, POSEM, Gilb.com] to plan and track critical performance and cost characteristics of a system (ill. courtesy Kai Gilb).

			Estimate		Actual		Estimate		Actual	
			Step 12 Buttons.Rubber				Step 13 Buttons.Shape & Layout			
Goals			Impacts				Impacts			
1	USER-FRIENDLINESS.LEARN		-10	33%	-5	17%	-5	20%	5	-20%
	30	5								
	by one year									
2	RELIABILITY		-3	-3%	-1	-1%	20	20%	2	2%
	99	200								
	by one year									
Resources			Impacts				Impacts			
	PROJECT-BUDGET		2000	2%	2500	3%	1000	1%	1000	1%
	2500	100000								
	by one year									

The pair of numbers in the three left hand columns (30, 5 etc.) are defined benchmarks (30, 99, 2500) and Goal levels (5, 200, 100,000).

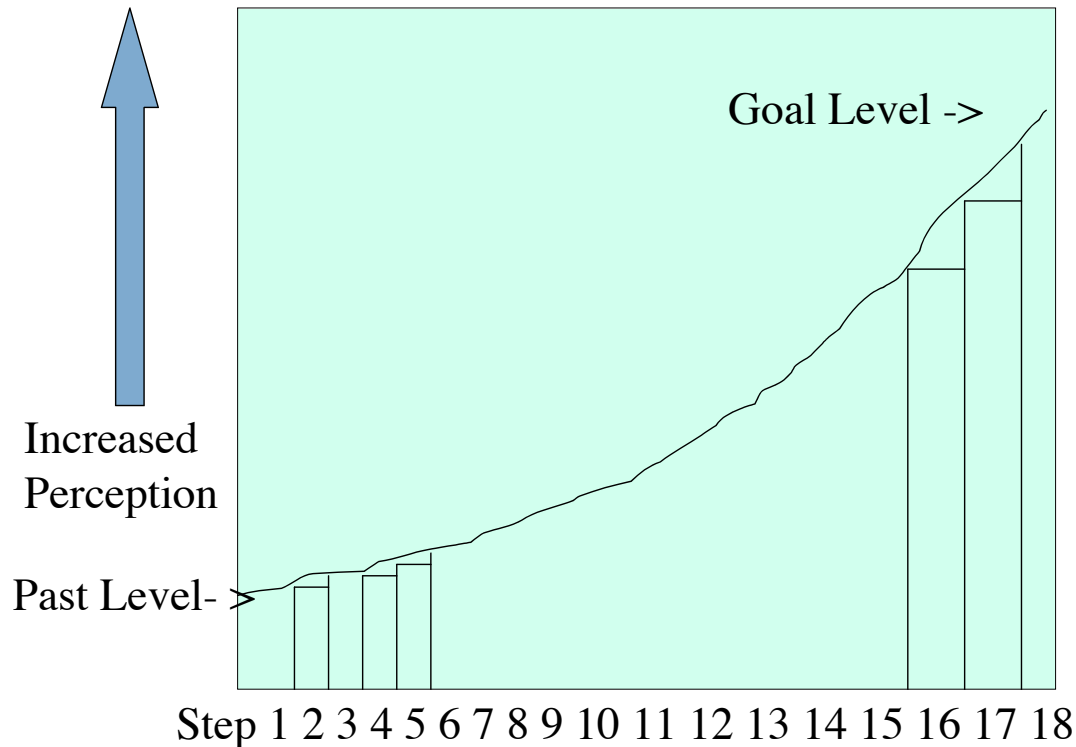
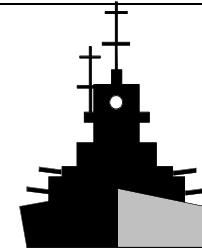
The ‘%’ figures are the real scale impacts (like 20) converted to a % of the way from benchmark to the Goal levels (like 20% of the distance from benchmark to Goal).

*8. When all Goals are reached:
‘Claim success and move on’ [Gerstner]
Free remaining resources for more profitable ventures.*

- **one advantage with numeric Goal levels,**
 - compared to a stream of yellow stickies from users,
 - is that it is quite clear when your objective is reached.
 - No additional effort should be expended to improve upon it,
 - unless a new improved target level is set.
- **the numeric goal level is the success level,**
 - success is well defined formally in advance.
- **a ‘Fail’ level (a ‘constraint’ , not a ‘target’) can also be set,**
 - in each required objective’ s specification,
 - to announce a lower limit (constraint).
 - Fail levels define an ‘acceptable’ (if not yet ‘successful’) range of each performance and cost characteristic.
- **Fail and Goal levels can be used to manage project decisions [CE].**
- **projects need to be evaluated on performance delivered in relation to resources used.**
- **This is a measure of project management ‘efficiency’ .**
- **When targets are reached,**
 - we need to avoid misusing resources to deliver more than is required.
 - Perfect performance and quality costs infinite resources.

***Results are cumulated numerically
step by step
until the Goal level is reached.***

The Naval Weapons System: Evo increase of Perception.



- In a UK Radar system I planned, the system was delivered by gradually building database info about plans and ships, tuning recognition logic, and tuning the radar hardware.***

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.

- Projects need to be judged primarily
 - on their ability to meet critical performance characteristics,
 - in a timely and profitable way.
- This cannot be expected if the project team is paid ‘by effort expended’ .

The team will be paid and rewarded for benefits delivered in relation to cost.

- Teams need to be paid by results delivered in relationship to costs. By their project efficiency.
- **Even if this means that super efficient teams get terribly rich! And failure teams go 'bankrupt'. Long live the capitalist free market mechanism!**
- **When only 13% of 1500 IT projects are 'successful' [Times],**
 - we clearly need to find better mechanisms for rewarding success,
 - and for not rewarding failure.
 - I suggest that sharp numeric definition of success levels
 - (Goal [China, End 2015] 65%),
 - and consequent rewards for reaching them,
 - is minimum appropriate behavior for any software project.

POLICY

“The team will find their own work process and their own design”.

- **Conventional Agile processes believe we need to reduce unnecessarily cumbersome corporate mandated processes.**
 - I agree.
- **They also believe in *empowering* the project team to find the processes, designs and methods that really work for them locally.**
 - I heartily agree!
- **But I believe that**
 - sharp numeric definition of objectives,
 - coupled with frequent estimation and measurement of progress,
 - is a clearly superior mechanism
 - for enabling this empowerment.
- **The price for this,**
 - a few estimates
 - and measures weekly,
 - seems a small price to pay
 - for superior control over project efficiency.

POLICY

*“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.
“*

- *No project team should be*
 - *‘stuck’ with trying to satisfy unrealistic*
 - *or conflicting stakeholder dreams*
 - *within constrained resources.*
- *The project team can only be charged with reasonable capability*
 - *to deliver inside ‘state of the art’ performance levels*
 - *and deliver inside ‘state of the art’ costs.*

A Management Policy for Architecture Prioritisation

- “Choose architecture that delivers early measurable stakeholder value, as defined by our top critical numeric objectives.”

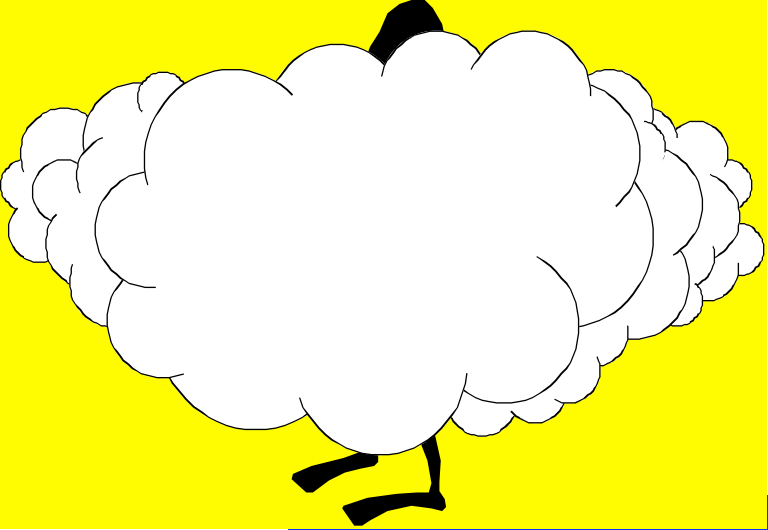
An Architecture Policy

- “Evaluate architecture in relation to all critical stakeholder targets and constraints,
- estimate impacts with regard to risk,
- deliver best impact designs early,
- drop architecture early with low impact to cost ratio.”

The 12 Tough Questions

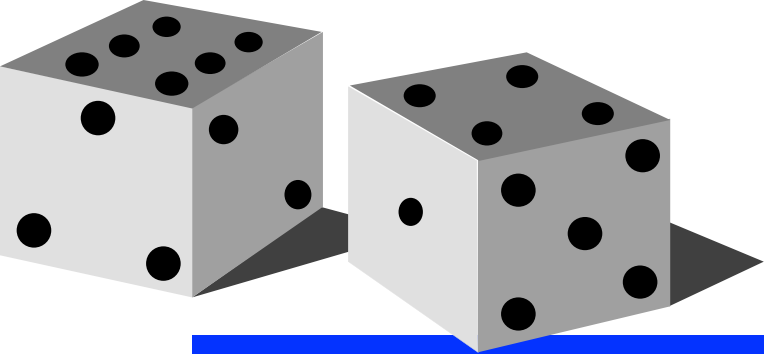
- Based on Competitive Engineering - Language concepts
- From a simplified management and common sense point of view
- Are you getting problems because you and your people do not ask these questions
- And demand good answers?
- Paper on 12 ?s at www.gilb.com
 - http://www.gilb.com/tiki-download_file.php?fileId=24





1. NUMBERS

- **Why isn't the improvement quantified?**



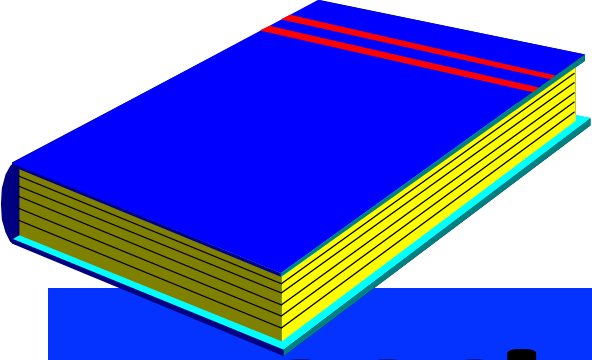
2. RISK

- **What's degree of the risk or uncertainty and why?**



3. DOUBT

- **Are you sure?**
- **If not, Why not?**

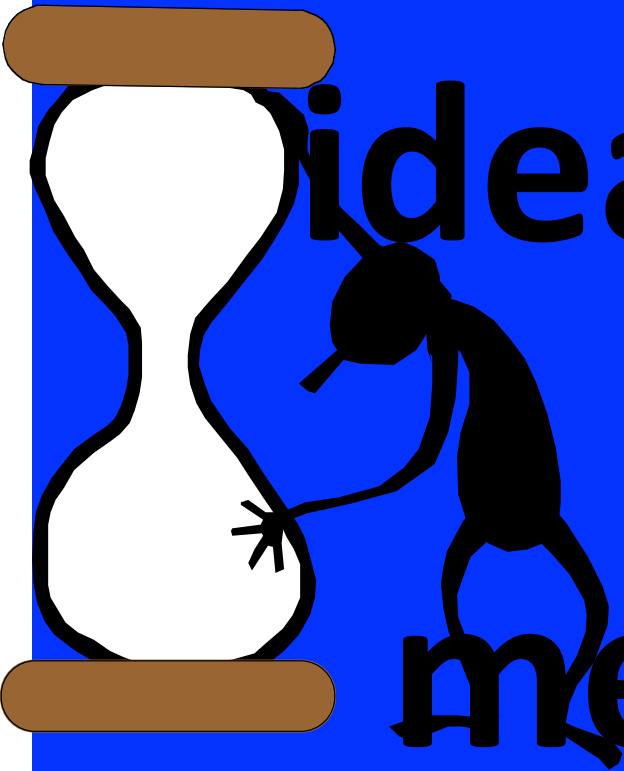


4. SOURCE

- **Where did you get that from?**
- **How can I check it out?**

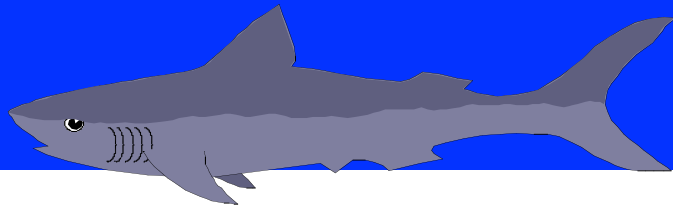
5. IMPACT

- How does your idea affect my goals, measurably?



6. ALL CRITICAL FACTORS

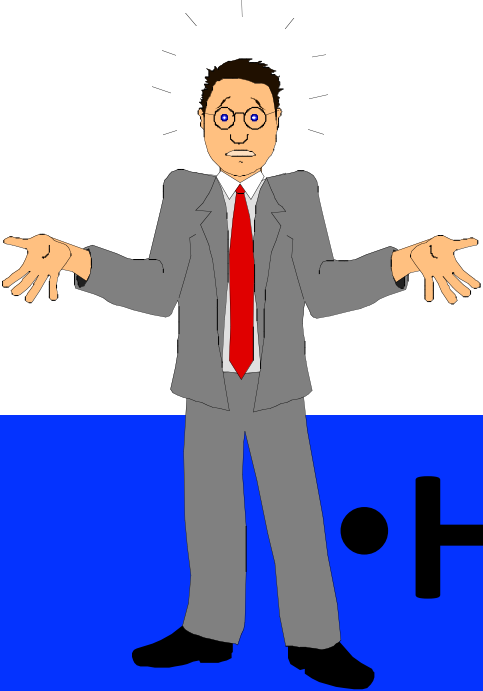
- Did we forget anything critical to survival?





7. EVIDENCE

- **How do you know it works that way?**
- **Did it before?**



8. ENOUGH

- Have we got a complete solution?
- Are all objectives satisfied?

9. PROFITABILITY FIRST

A black silhouette of a person is shown on the left side of the frame, reaching out towards a carrot. The carrot is orange with green leafy tops and is suspended by a thin black string from a small grey ring. A curved grey line starts from the top left and ends near the ring. The background is a solid blue rectangle.

**Are we planning
to do the
profitable things
first?**

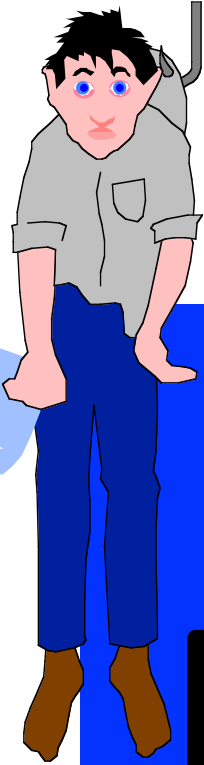
10. COMMITMENT

- Who's

Responsible for

failure or

success?





11. PROOF

- How can we be sure the plan is working, during the project, early?

12. NO CURE

- Is it 'no cure, no pay', in the contract?



No cure

No pay!

- Why not?

Concluding Remarks

- “PRIORITISE EFFICIENT ARCHITECTURE DYNAMICALLY”
- *I’ll be at the conference today and tomorrow for corridor discussion*
- *Call me or text to*
- *077 1 267 0707*
- *Or email tom @ gilb . com*

End

- Free digital copy of Competitive Engineering
- Will be sent to conference participants, this week, who email me [tomsgilb @ gmail.com](mailto:tomsgilb@gmail.com)
- With BOOK in subject

