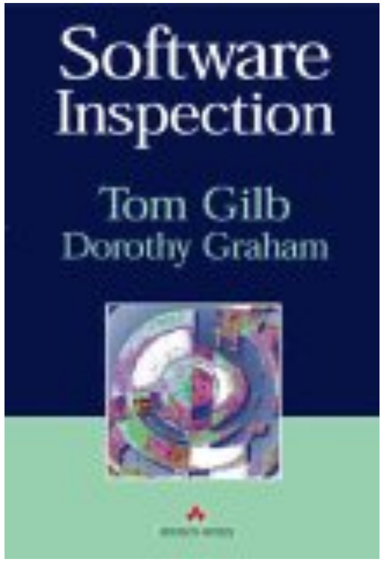
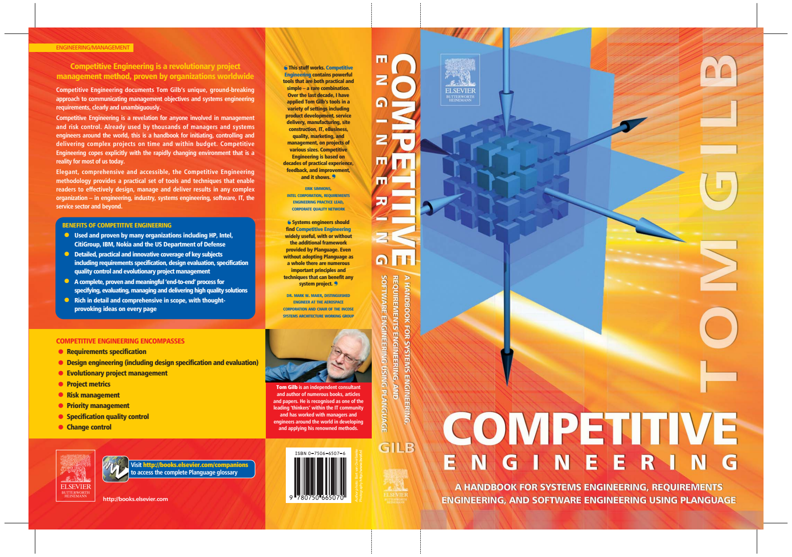
Project Predictability:
reducing costs and improving profitability in systems engineering

(admittedly in advance of my deeper analysis of specific problems!)

Some preparatory notes and Ideas for meeting with XX
Dec 7 2009 08:00
BY

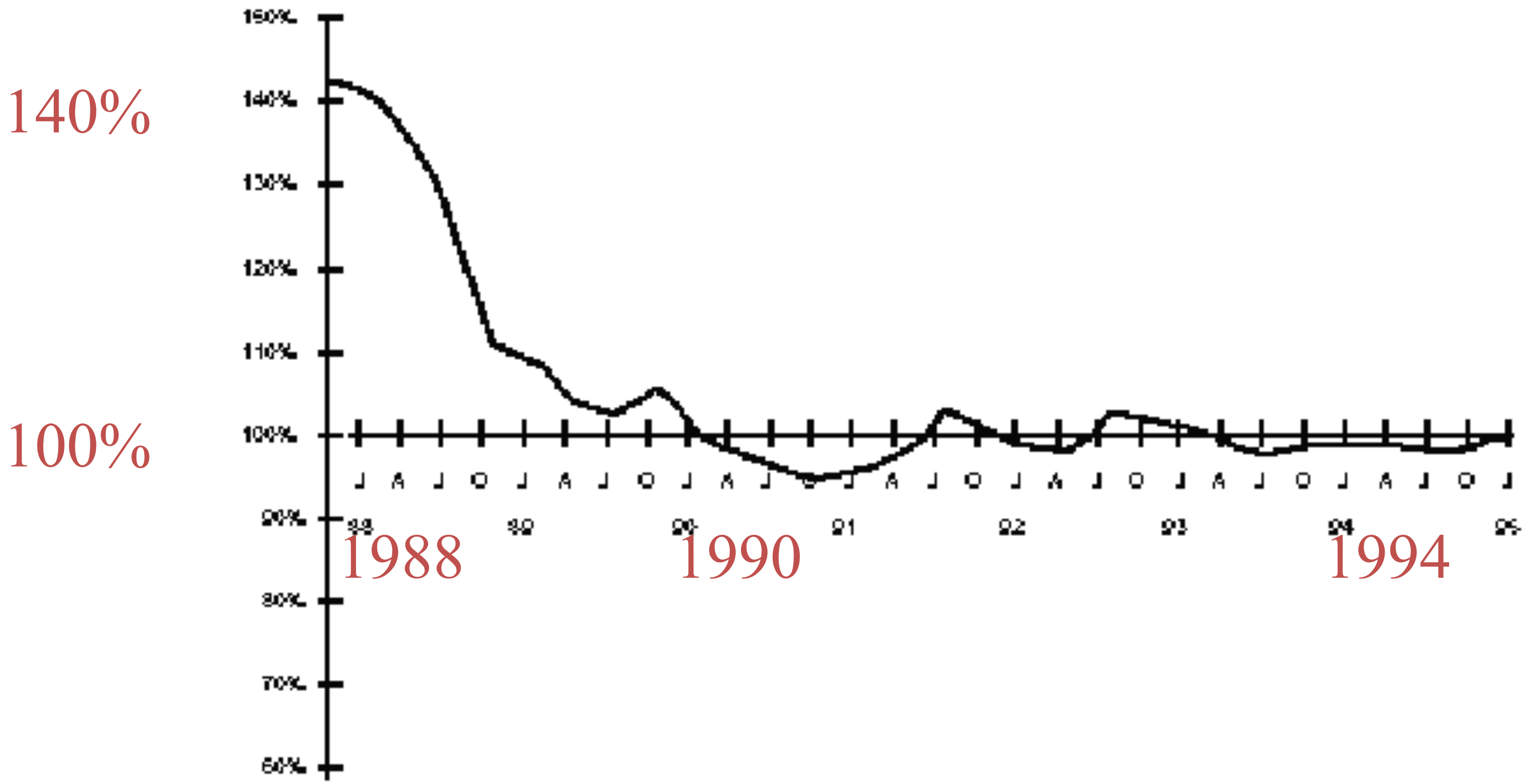
Tom@Gilb.com
www.gilb.com
+47 920 66 705





Achieving Project Predictability: Raytheon, Using Inspection + DPP*

Cost At Completion / Budget %



* DPP = IBM Mays Defection Prevention Process.
See chapters 7 & 17 Gilb, Software Inspection

SEE PPTx NOTE in this slide FOR Predictability DEFINITION.

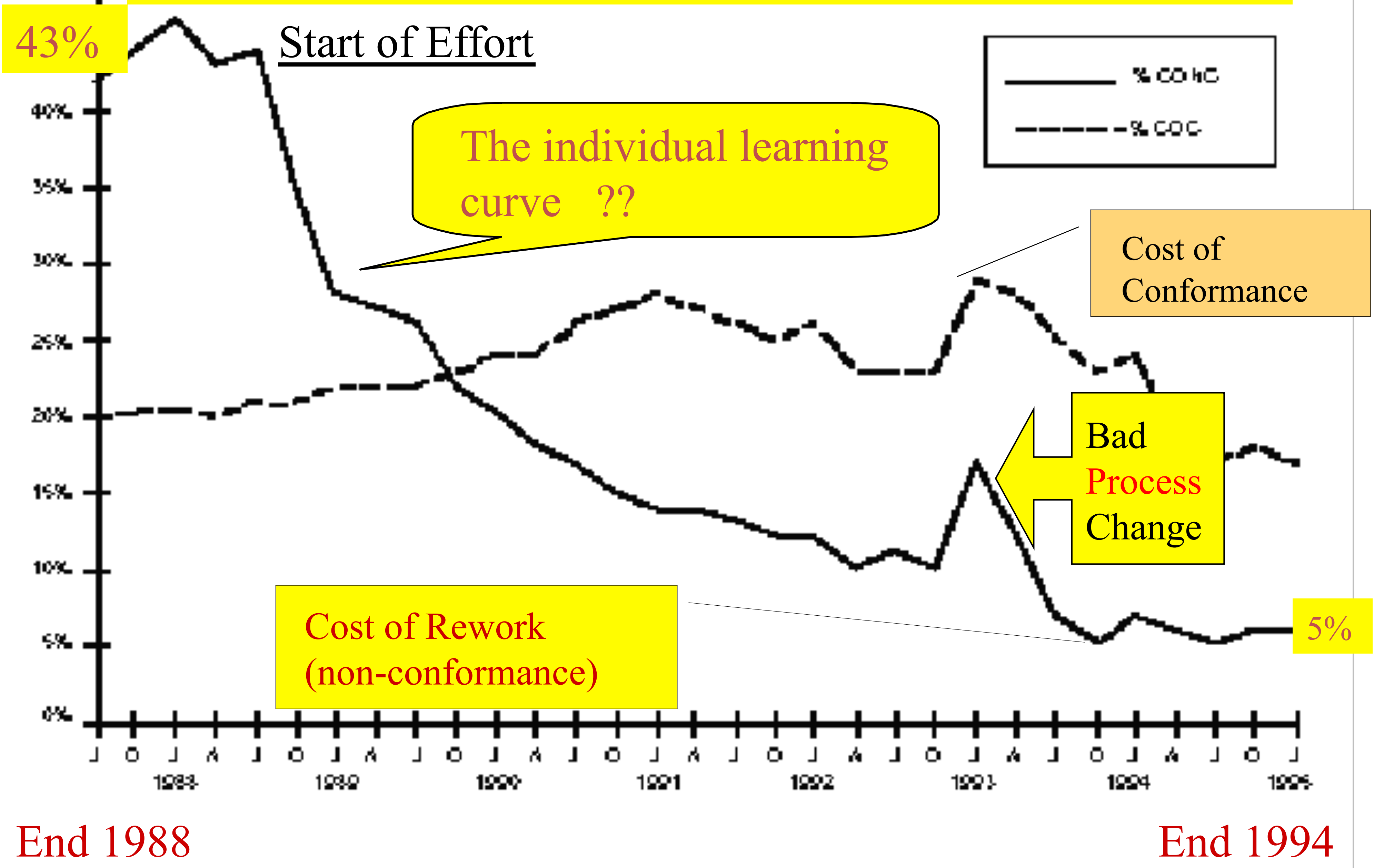
Causes – main ones!

of financial 'variability' in Systems Engineering

- **Management Causes**: bad practices, good ones not taught
  - Unclear Main Critical Project Objectives: un-quantified
  - Unclear Organization Improvement Objectives
  - Bad Contracting practices
  - Bad Bidding practices
- **Technical Causes:**
  - Allowing GIGO* in all processes: no numeric exit
  - Bad 'Standards': weak and not taken seriously
  - Poor Architecture: not designing to objectives
  - Slow feedback from complex reality
- Other causes
  - Political
    - Software sub-suppliers incompetence and lack of ethics*

* GIGO = Garbage In Garbage Out. Uncontrolled flow of major spec defects

40% of your costs are unnecessary! (Raytheon)



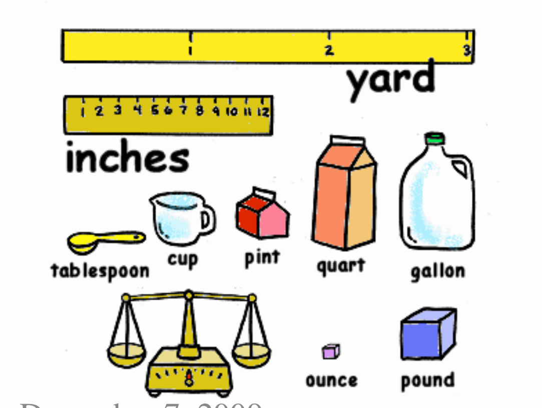
INSPECTION AND DPP = MAIN MOTOR AND MAIN ENGINE

Cures: Result and Value Orientation (which leads to cost and time predictability)

- **Management** Cures:
  - Quantified, Quality Control of Management-Level specifications ('Agile SQC')
    - Like Objectives, strategies, bids, contracts, major decisions, work processes (like estimation).
    - Based on strong 'rules' (like quantify all qualities)
    - Strong Quantified Exit/Entry to Management Processes
      - Max 1.0 majors/page at release (exit)
  - Train Management to live in a quantified quality world (above)
  - Use 'Impact Estimation Tables" to get management overview of multiple strategies versus multiple investments = Intelligent Prioritization

CASE: $100 MILLION PROJECT LOSS
Summary of Top '8' Project Objectives

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



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

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

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

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

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

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

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

*7. **Robustness** is an essential system requirement (see rewrite in example below)*

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

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

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

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:

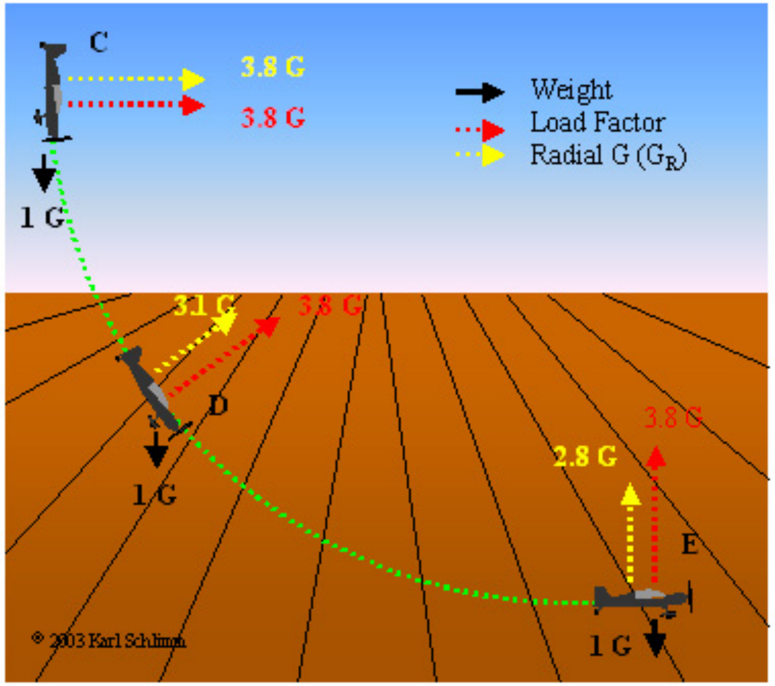
Software Downtime:
Type: Software Quality Requirement.
Ambition: *to have minimal downtime due to software failures <- HFA 6.1*
***Issue**: does this not imply that there is a system*
requirement?

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

Fail [Any Release or Evo Step, Activity = Recompute, Inrensity = 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. <-6.1.2 HFA.

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

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

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

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

Catastrophe: 100 minutes.

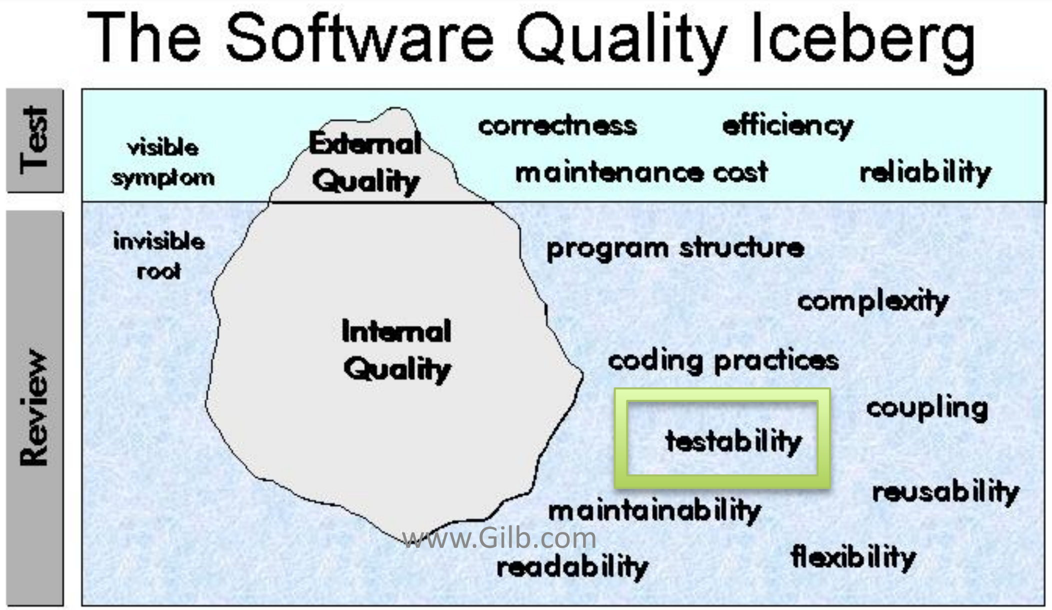
Testability:

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*



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

1,000 Person Organization, Subsidary of much larger multinational

Business Objectives Quantified

Business objective	Measure	Goal (200X)	Stretch goal ('0X)	Volume	Value	Profit	Cash
Time to market	Normal project time from GT to GT5	<9 mo	[illegible]6 mo	X		X	X
Mid-range	Min BoM for The Corp phone	<$9[illegible]	[illegible]3	[illegible]		[illegible]	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	[illegible]	[illegible]	X		X	X
Productivity						[illegible]	X
Get Torden	Lyn goes for Technology 66 in Sep-04	Yes		X		X	X
Fragmentation	Share of components m[illegible]ified	<10%	<5%		X	X	X
Commoditisation	Switching cost for a UI to another System	>1y[illegible]	>[illegible]rs			[illegible]	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 XXXX sales	>50%	>60%	X		X	X

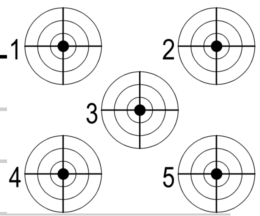
Numbers are intentionally changed from real ones

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

Technical Strategies

Objectives

Viking Deliverables



Business Objective	hardware adaptation	Telephony	Reference designs	IFace	Modularity	Defend vs Technology 66	Tools	User Experce	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%	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	5%	10%	5%	5%	0%	0%
Platformisation Technology	25%	10%	30%	0%	[illegible]	10%	0%	5%	0%	10%	0%	5%
Interface	5%	15%	15%	0%	5%	0%	5%	0%	0%	10%	0%	10%
Operator preference	0%	[illegible]	[illegible]	[illegible]	[illegible]	20%	5%	10%	10%	20%	5%	10%
Get Torden	25%	10%	[illegible]	-10%	0%	20%	0%	10%	-20%	10%	10%	5%
Commoditisation	20%	10%	20%	10%	-20%	25%	15%	0%	0%	5%	10%	5%
Duplication	15%	[illegible]	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%	[illegible]	0%	0%	[illegible]	0%	0%	30%	10%	0%	0%	0%
Downstream cost saving	15%	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	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%	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	5%
Cost (£M)	£ 2.85	£ 0.49	£ 3.21	£ 2.54	£ 1.92	£ 2.31	£ 0.81	£ 1.21	£ 2.68	£ 0.79	£ [illegible]	£ 0.60
ROI Index (100=average)	106	358	109	33	78	[illegible]	148	107	10	152	202	174

Strategy Impacts on Objectives

"Benefits"

Cost

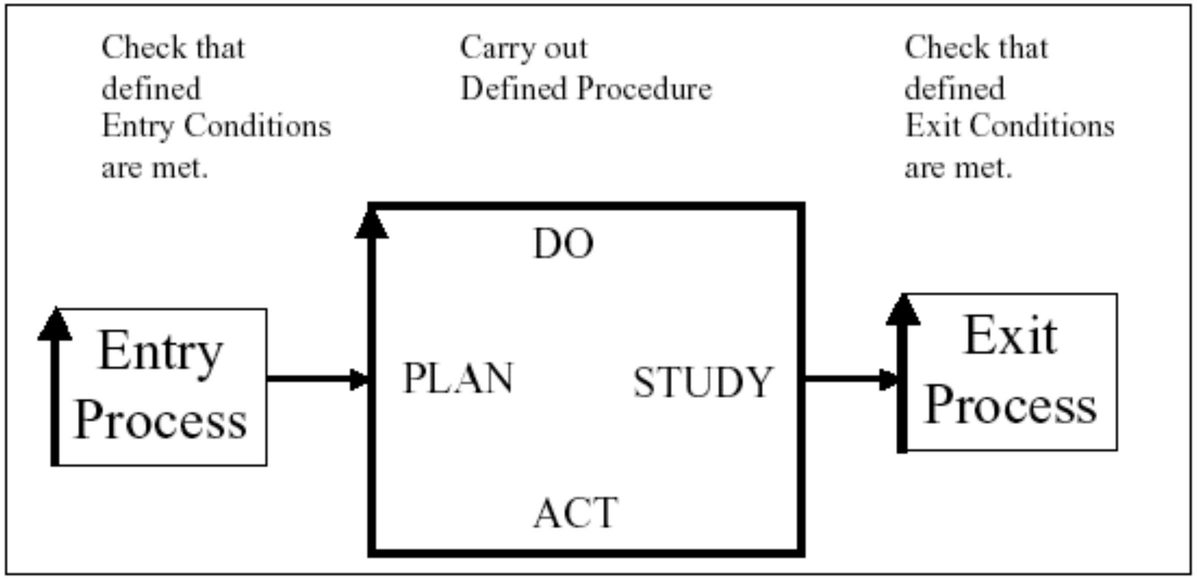
Benefit/Cost ratio

358 !

Cures: Result and Value Orientation
(which leads to cost and time predictability)

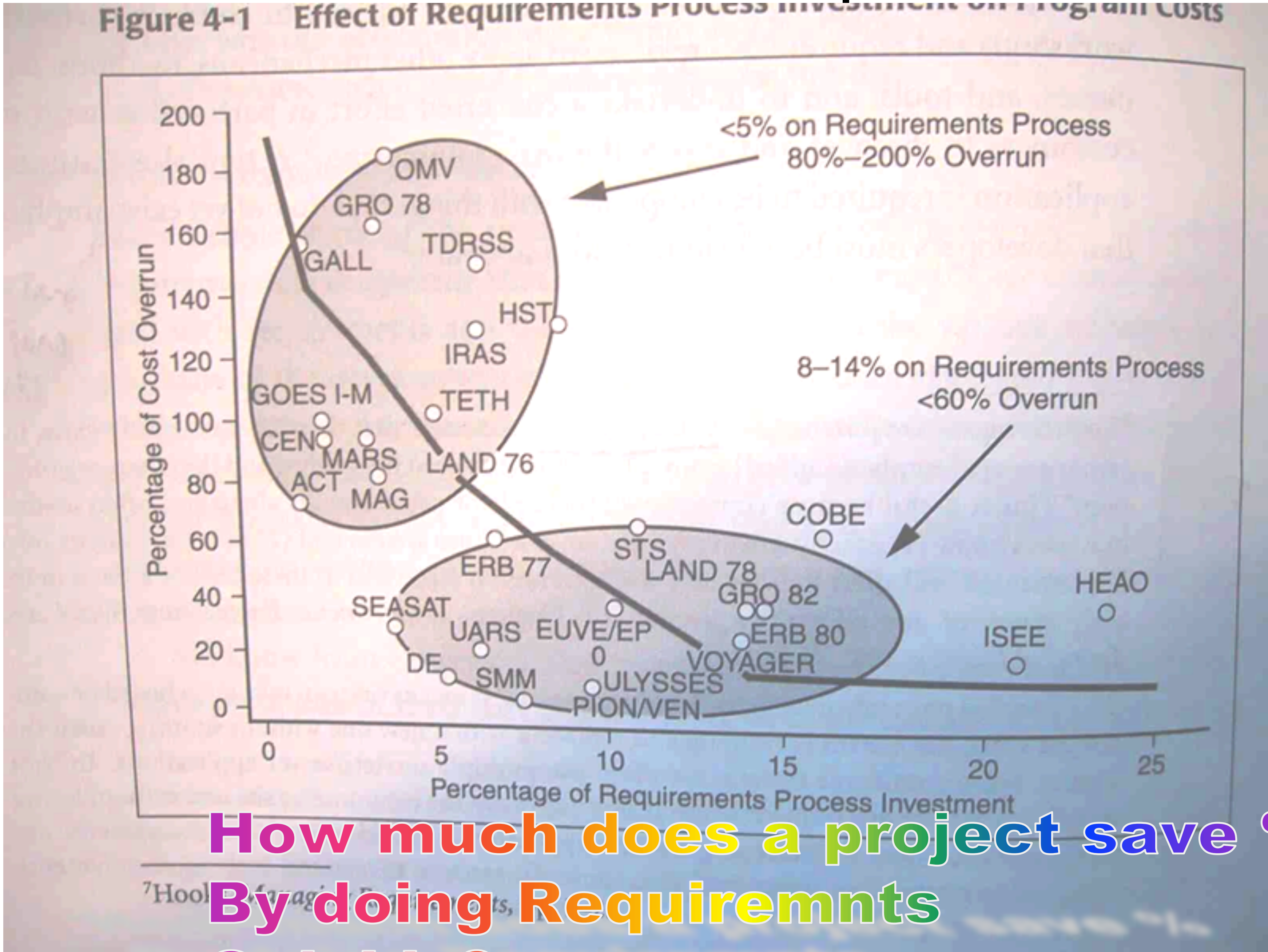
- **Technical** Cures
  - *Quantified* Exit and Entry to all Tech Processes
    - Max 1.0 Major defects/page allowed, not ‘180’!
  - The use of Agile (sampling) Spec QC to easily and quickly measure the quality level of tech work
  - The use of a smarter specification language (‘Planguage’) to deal with quality, risks, priorities in a much more intelligent ways
  - Architecture closely related to quantified quality and constraint requirements
    - Architect to Cost!
  - Real Systems Engineering (design to Quality and Cost)
    - Get cost control by ‘design to cost’
  - Much faster value delivery cycles
    - For feedback and adjustment to reality
    - For early payment
    - To avoid surprising costs getting out of hand

The quantified Exit and Entry controls



- Entry and Exit Condition example:
- Maximum estimated 1.0 Major defects per logical page remaining.
- This was the MOST important lesson IBM learned about software processes (source Ron Radice, co-inventor Inspections, Inventor of CMM)

Investment in Requirements



How much does a project save %
By doing Requiremnts
Quickly?

Requirements Monday Erieye 1 0f 3

USAB:USABILITY:

Ambition: Operator ease of learning & doing tasks under <all conditions> should be maximum possible ease & performance with minimum training & minimum <unchecked error> possibility.

TRAINED: DEFINED : Command and Control Operator onboard, who has been through approved training course of at least 200 hours duration.

RARETASKS: DEFINED: types of tasks performed by a single operator less than one a week average.

TASKS DONE: DEFINED: Onboard operator distinct tasks carried out.

ERICSSON
TAKING YOU FORWARD

GREEK EriEye

Saab Argus 100H during its short service with HAF

ERICSSON
TAKING YOU FORWARD

http://hafcphotos.cs.net/view/gp.cfm?photoid=141961&type=3

Erieye 2 of 3, Intuitiveness Requirement

INTUITIVE: USAB.INTUITIVENESS

Ambition: High probability in % that operator will <immediately> within a specified time from deciding the need to perform the task (without reference to handbooks or help facility) find a way to accomplish their desired task.

Scale: Probability that an <intuitive>, TRAINED operator will find a way to do whatever they need to do, without reference to any written instructions (i.e. on paper or on-line in the system, other than help or guidance instructions offered by the system on the screen during operation of the system) within 1 second of deciding that there is a necessity to perform the task. <-- MAB "I'm not sure if 1 second is acceptable or realistic, it's just a guess"

Meter: To be defined. Not crucial this 1st draft ←- TG

Past [GRAPES] 80% ? ← LN

Record [MAC] 99%? ← TG

Fail [**TRAINED, RARETASKS** [{<1/week,<1/year}]] 50 - 90%?← MAB

Goal [TASKS DONE [<1/week (but more than 1/Month)]] 99% ?← LN

[**TASKS DONE** [<1/year]] 20% ? ←- JB

[Turbulence, **TASKS DONE** [<1/year]] 10% ? ←- TG

ERICSSON
TAKING YOU FORWARD

Erieye 3 of 3 Intelligibility Requirement

INTELL: USAB.**INTELLIGIBILITY:** "synonym tags, USAB is defined above"

Ambition : High ability to <correctly> interpret meaning of a [set] of <inputs> by the operator.

Scale: Probability in % of <objectively correct> interpretation(s) of a defined [set] of information within [defined time limits]

Meter [**ACCEPTANCE**] X (10) trained operators, Y (100) <representative> sets of information per operator within 15 minutes. ? ← MAB

"Not sure if the 15 minutes are realistic"

"this is a client & contract determined detail"

M1: Past : [XXX, 20 trained operators, 300 data sets in 30 minutes] 99.0% <-- Acceptance test report from XXX. MAB

Record [XXX] 99.0% "None other than XXX known by me" ←MAB

Fail [**DELIVERY CYCLE** [1]] 99.0% ? ←MAB

Fail [**ACCEPTANCE**] 99.5% ? ←MAB

Goal [M1 "parameters as above"] 99.9% ←LN

ACCEPTANCE: DEFINED**:** formal acceptance test, as defined by our contract with a particular customer.

DELIVERY CYCLE: DEFINED**:** Evolutionary result delivery cycle. Integrated, useful.

ERICSSON

TAKING YOU FORWARD

Executive Action and Options

- Develop New Policies
  - Quality Management
  - Cost Management
  - Design Management
  - Risk Management
  - Prioritization
  - Project – Evolutionary Method
  - Predictability Management
- Make a Corporate Improvement Plan
  - Top Objectives Quantified (deviation. Predictability)
  - Identify top ten most powerful strategies, esp. meta-strategies
  - Decompose strategies into small implementable (2%) delivery steps
  - Begin ‘next week’ with implementation in practice
  - Measure progress and adjust

Predictability Management Policy

- Root Causes of Deviations from numeric plans will be tackled
- Degrees of deviation of all critical qualities and costs will be kept for all projects and analyzed for change

Examples of Objectives

- Ericsson Case: Engineering Productivity x 2
  - More detail in
  - Top level Critical Project Objectives
  - at:
  - http://www.gilb.com/tiki-download_file.php?fileId=180

ERICSSON
TAKING YOU FORWARD

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**:

ERICSSON
TAKING YOU FORWARD

ERICSSON
TAKING YOU FORWARD

‘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

Examples Strategies

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?

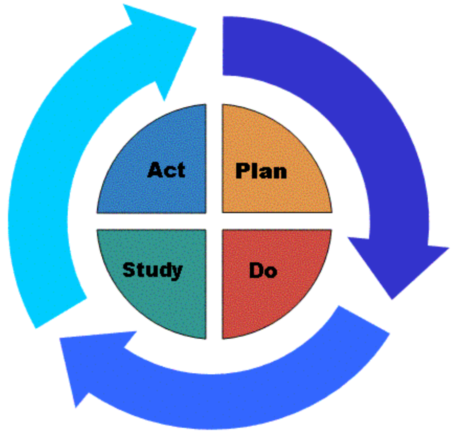
ERICSSON
TAKING YOU FORWARD

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+]

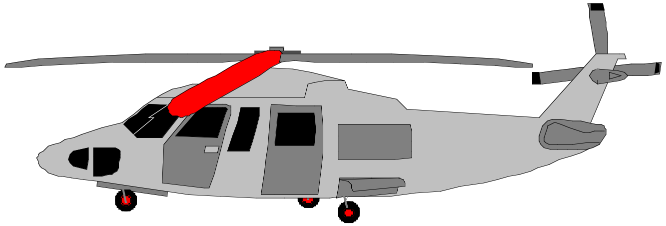
DPP/EVO, Master them and Spread them on priority basis.

Short Term Goal [Next Weeks]

DPP [RS?]
EVO [Package C ?]

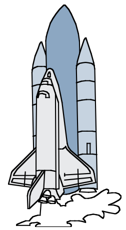
ERICSSON
TAKING YOU FORWARD

Decision: {Go, Fund, Support}



Harlan Mills on **Predictability** and he used *Inspection*!

- "Software Engineering began to emerge in FSD" (IBM Federal Systems Division, from 1996 a part of Lockheed Martin Marietta) "some ten years ago [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 [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.
  - **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."** Harlan Mills



[IBM Systems Journal No. 4, 1980, p. 415], Reprinted IBM SJ Vol. 38 1999, 289-295

See note for Flight software. http://history.nasa.gov/sts1/pages/computer.html Case Study, "The Space Shuttle Primary Computer System," *Communications of the ACM* 27, No. 9 (September 1984): 871–900.
See note for Weinberg history FSD via Mercury project

GOOD QUALITY RESULTS > 90% SUCCESS RATE

- Formal Inspections (Requirements, Design, and Code)
- Joint Application Design (JAD)
- Software Six-Sigma methods (tailored for software projects)
- Quality Metrics using function points
- Quality Metrics using IBM's Orthogonal classification
- Defect Removal Efficiency Measurements
- Automated Defect tracking tools
- Active Quality Assurance (> 5% SQA staff)
- Utilization of TSP/PSP approaches
- => Level 3 on the SEI capability maturity model (CMM)
- Formal Test Plans for Major Projects
- Quality Estimation Tools
- Automated Test Support Tools
- Testing Specialists
- Root-Cause Analysis

http://www.gilb.com/tiki-download_file.php?fileId=250

Impact Estimation

- Case US DoD
  - More detail
  - Top level Critical Project Objectives
  - at:
  - http://www.gilb.com/tiki-download_file.php?fileId=180

DoDef. Persinscom Impact Estimation Table:

Designs

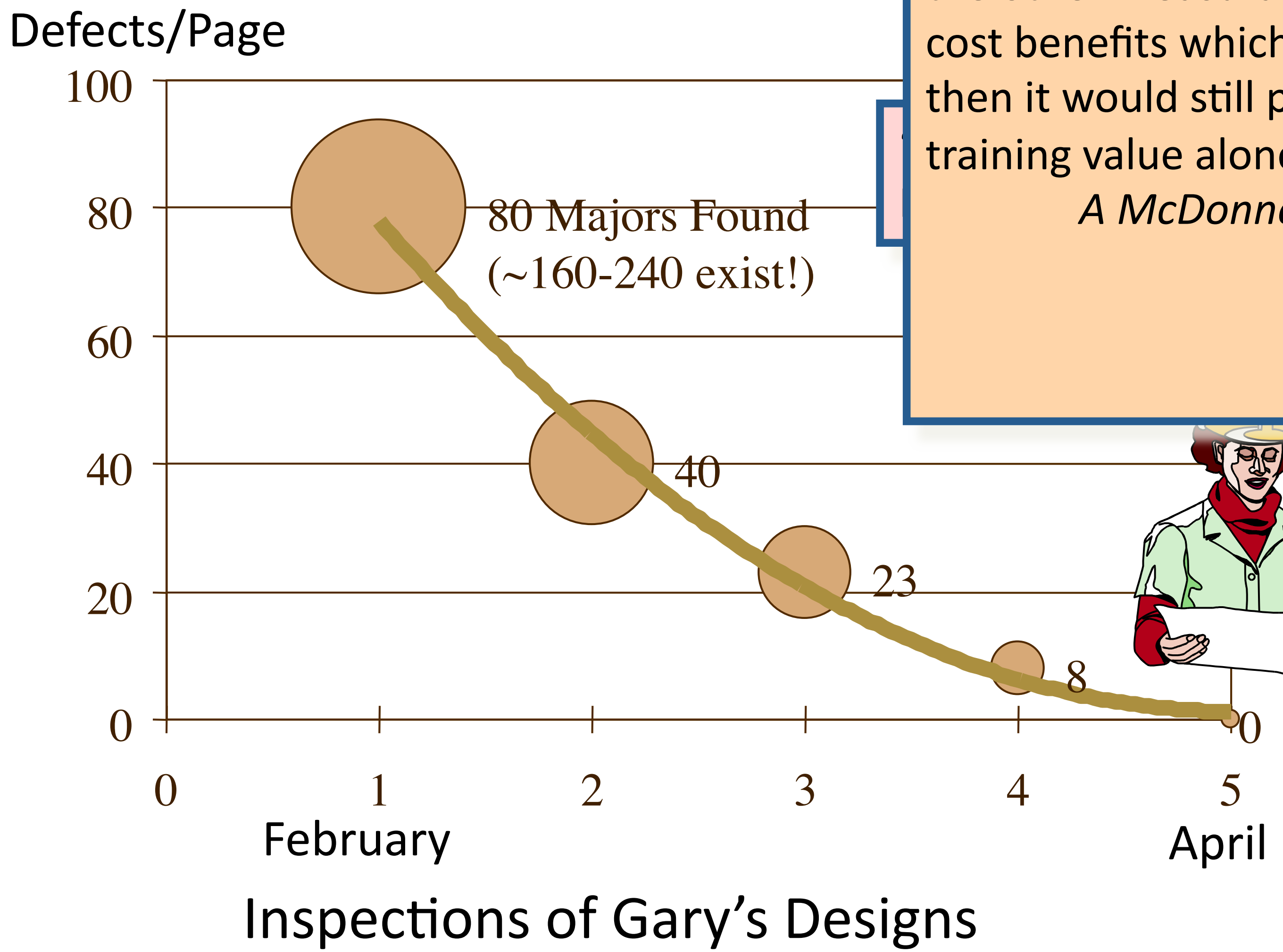
Requirements

R→ D Impacts

Design Ideas ->	*Technology Investment*	*Business Practices*	*People*	*Empowerment*	*Principles of IMA Management*	*Business Process Re-engineering*	*Sum Requirements*
	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%				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%
Sum of Performance	*482%*	*280%*	*305%*	*390%*	*315%*	*649%*	
Money % of total budget	15%	4%	3%	4%	6%	4%	36%
Time % total work months/year	15%	15%	20%	10%	20%	18%	98%
Sum of Costs	*30*	*19*	*23*	*14*	*26*	*22*	
Performance to Cost Ratio	*16:1*	*14:7*	*13:3*	*27:9*	*12:1*	*29:5*	

Evolutionary improvement

Positive Motivation:
Personal Improvement
The value of **inspection** as Training
at MD/Boeing

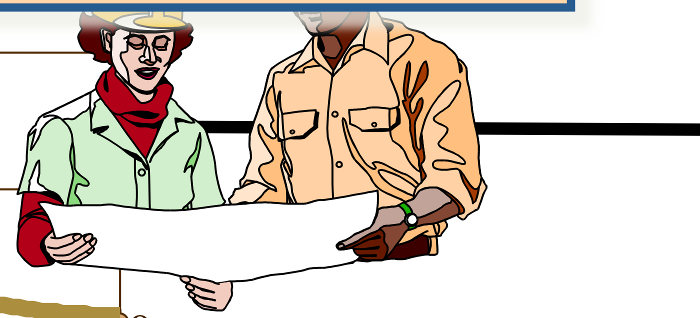


“We find an hour of doing Inspection is worth ten hours of company classroom training.”

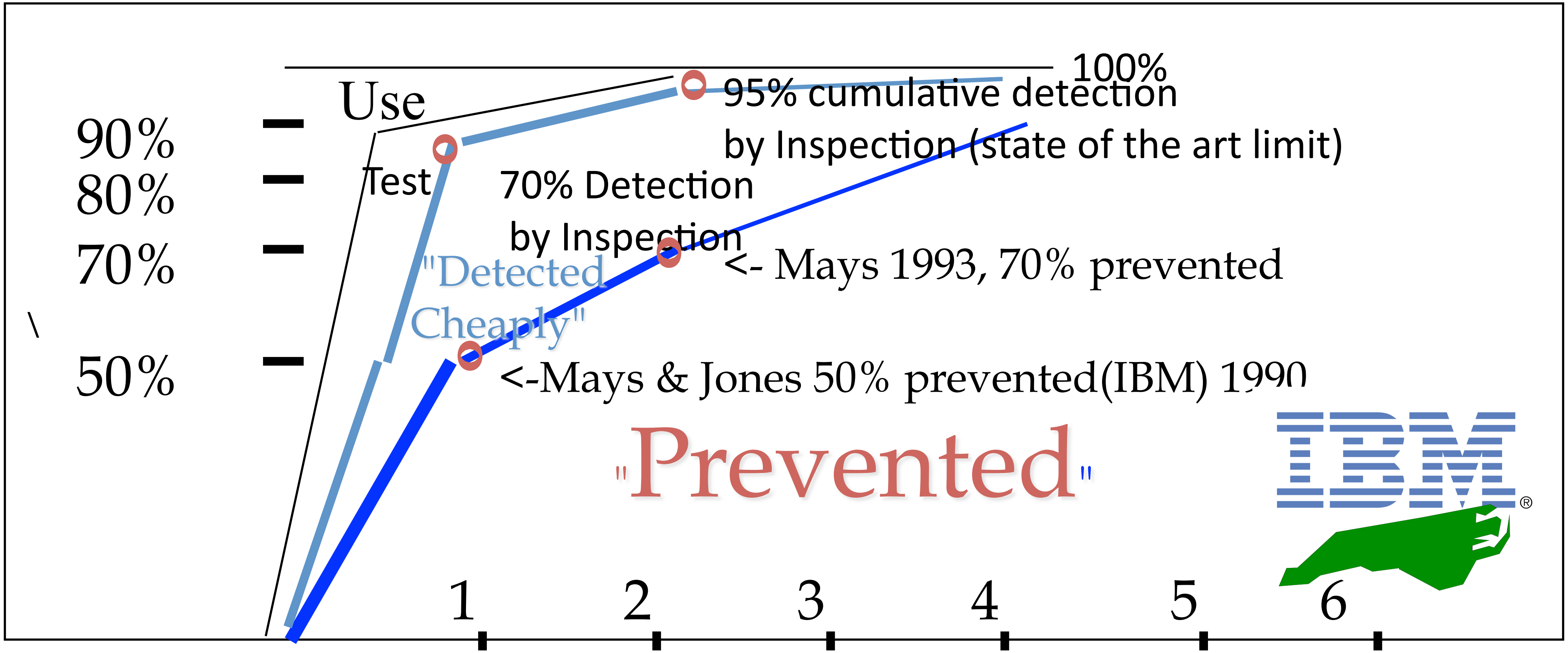
A McDonnell-Douglas line manager

“Even if Inspection did not have all the other measurable quality and cost benefits which we are finding, then it would still pay off for the training value alone.”

A McDonnellDouglas Director



The evolution of defect prevention



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

Gilb's Evo Method Used Widely at HP and Studied 'Scientifically'

RAPID AND FLEXIBLE PRODUCT DEVELOPMENT: AN ANALYSIS OF SOFTWARE PROJECTS AT HEWLETT PACKARD AND AGILENT

by

Sharma Upadhyayula

M.S., Computer Engineering
University of South Carolina, 1991

Submitted to the System Design and Management Program in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Engineering and Management

at the
Massachusetts Institute of Technology

January 2001

Signature of Author ....
System Design and Management Program
December 13, 2000

http://www.gilb.com/tiki-download_file.php?fileId=65

Rollout: 2 pronged approach

- **BROAD PROOF**: Choose 1 (or few projects) to prove case on
  - Prove case
  - Then scale up and out
  - Do 'most everything' on these projects
  - But make sure you have expert coaches
  - And real commitment to do things properly
- **EARLY LEVERAGE**: Select one strategy, and spread it widely in 2010
  - I suggest
    - **Quantified Top Level Objectives** – FOR ALL PROJECTS
      - Supported by
        - » 1 day director-level management training in quantification
        - » Standards: Policy, rules, exit and entry levels
        - » Quality Control of Objectives according to standards
        - » Top Management Breathing Down Neck
        - » Qualified Coaching, especially first time.

Descartes On Small

- "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

1.1.1.1.1.1.1 or 7x1 or The Power of 1
Principle for Evolutionary Steps Decomposition

- Find
- 1 Stakeholder Type, and from them derive
- 1 Value of that stakeholder type and then
- 1 Quality that delivers that value, and then
- 1 Design that delivers that Quality to a
- 1 Real Instance of the Stakeholder Type, in
- 1 Week, and plan this within
- 1 Hour

Examples of what NOT to do

- ‘Go Lean’
- Get to CMMI Level x
- ‘Go Agile’
- More Formal Processes
- Centralize Control, micromanage
- Demand Deadlines and Cut budgets
  - Without controlling necessary qualities

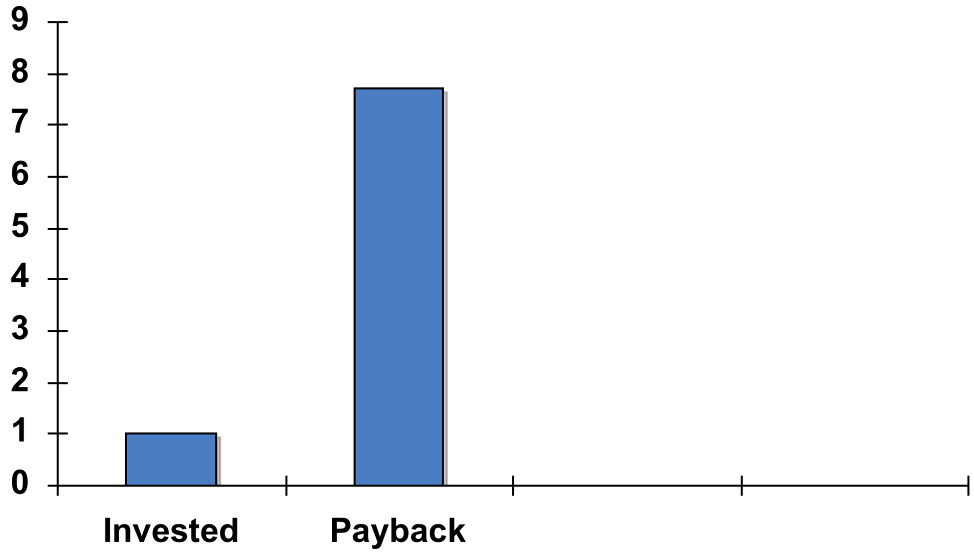
Popular Myths of Management Control

- CMMI
  - CMM(I) is a stupid bureaucracy in practice
    - Which will inflate costs – it does not even try to deal with costs
    - And not control quality – it does not even try
    - See David Rico for data on how bad it is! (lowest ROI)
- Process
  - Processes themselves do not control costs
  - There are some intelligent processes
    - Like dynamic design to cost (Cleanroom!, Mills)
    - Like Agile Spec QC applied to judge Exit level
    - Like evolutionary value delivery
- Agile
  - Current 'agile' methods (Scrum) bear no relationship to cost and quality
- Lean
  - Lean principles are intelligent, but don't 'go lean'
  - Measure improvements in organizational behavior, in relation to your own quantified organizational objectives
- Inspection: in 'clean-up-bad-work' mode, before test
  - Costs too much
  - Is ineffective

“Selling” the Change

- Sell the ROI (see next slides for real examples)
- Sell the Measurable results
- Sell the short term, and long term result
- Sell the low risk – no big up front investment
  - You invest as you see the payoff is *real*

Return On Investment at Raytheon
about $10,000 per programmer/year
Remember Inspections and DPP= Main Driver and Engine''

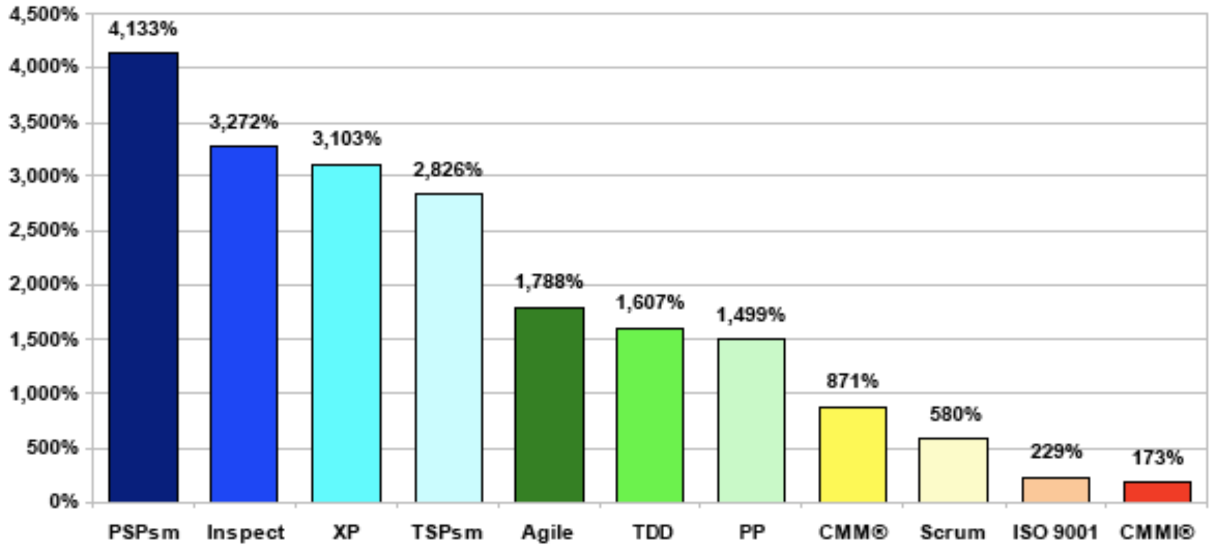


- $7.70 per $1 invested at Raytheon
- Sell your improvement program to top management on this basis
- Set a concrete target for it
  - Goal [Our Division, 2 years hence] 8 to 1

David Rico Studies
(many at gilb.com)

ROI of Individual Methods

- Data for all methods was used for comparison
- Best Agile and Traditional Methods had top ROI
- Agile Methods better than big Traditional Methods

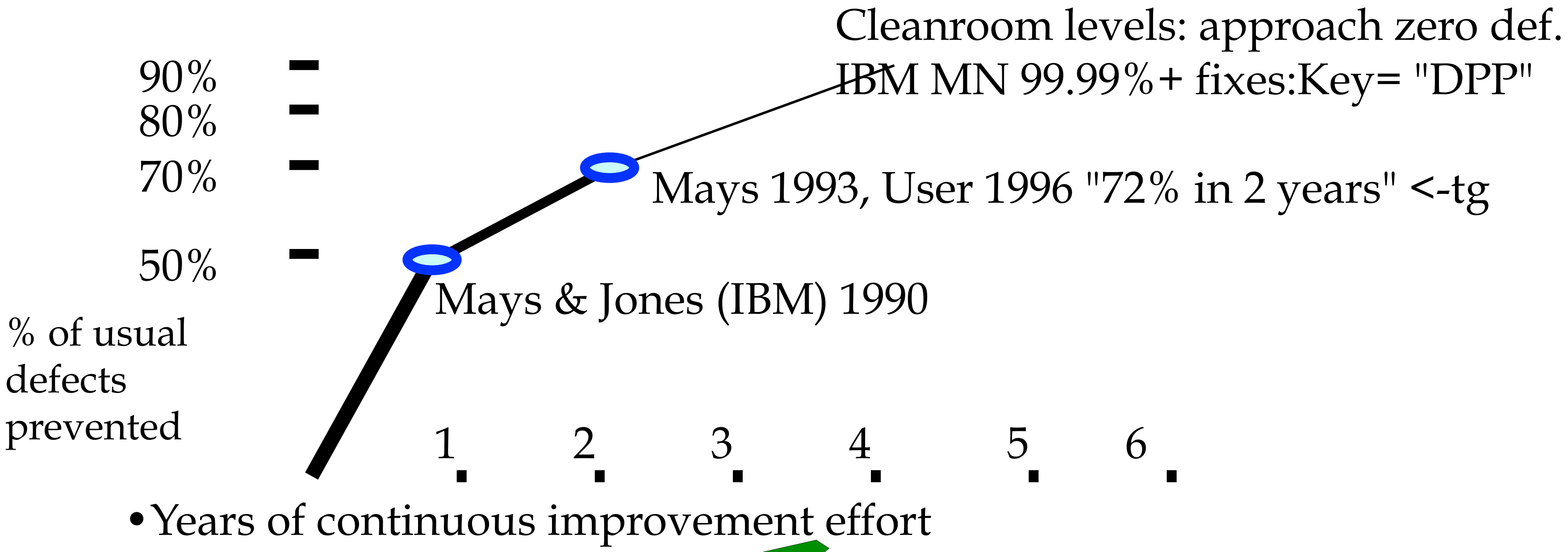


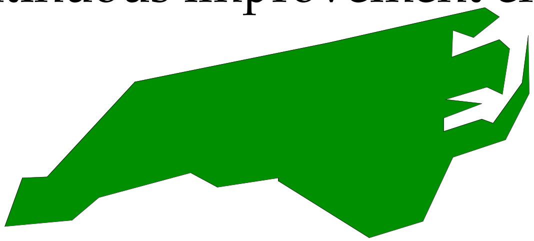
Rico, D. F. (2008). *What is the ROI of agile vs. traditional methods?* Retrieved September 3, 2008, from http://davidfrico.com/rico08b.pdf

CEO AND Board Support

- Set quantified objectives for the changes
- Select Strategies, show effects as Impact Estimations
- Show graphs of projected change over time
  - See next slide for real example
- Take Top Management (COO?) Responsibility
- Form a change team

Defect Prevention Experiences:
THERE IS A PREDICTABLE IMPROVEMENT IN DEFECT INSERTION RATES USING 'DPP' (CMMI 5)




North Carolina

IBM Research Triangle Park Networking Laboratory

Principles of Professional Change

(© Gilb, 12.2009)

1. You have to define your critical organizational improvements quantitatively
2. You have to judge all organizational strategies in relation to these critical objectives
3. You have to roll out change early and often, and measure the effect immediately
4. You have to prioritize change strategies that really work, and kill off those that don't, before scaling up
5. Focus on Meta-Strategies: those that allow decentralized feedback and change during projects (like Evo and Spec QC)
6. Project Architecture must explicitly address quantified project objectives
7. All cost-driving specification must be quality controlled and have high quality (1 Major defect/page) exit levels
8. Projects must deliver stakeholder value incrementally and measurably
    1. Thorough stakeholder value analysis must be used to prevent cost surprises later
9. Always prioritize the most efficient strategies next: high value/cost wrt risk
10. Sub-contract for no cure no pay, not for fixed cost or time and materials

Meta Policy*

- Quantify all aspects of Critical Qualities
- Use Impact Estimation Tables to evaluate all ends-means (Objectives-Strategies) situations
- Use Spec QC sampling to measure Major defects
- Develop simple short but powerful standards for specification (rules), like quantify quality, and entry exit levels to work processes.
- Evolve everything: delivery value early, learn and change early
- Reward real systems level planned stakeholder value delivery – not component completion
- Drive projects based on critical few requirements, treat all supporting detail as 'design' to meet these requirements
- Engineer quality and low cost into the systems by design

- * meta = **denoting something of a higher or second-order kind *: metalanguage | metonym.***

The simplest and most powerful initial changes.

- Exit:
  - Formal numeric work process release control
- Critical Quality Quantification
  - Extreme clarity of central purpose
- No cure no pay
  - Mange results, not effort
- Competition:
  - prove ideas internally by measured competition

If I were COO.

- Prove out this advice quickly
- And scale up as fast as it proves useful.
- Find champions for the changes
- Publish to policy asap

End - Finis

- Gilb.com