How Good Is A Process?

Evaluating Engineering Processes' Efficiency

Tom Gilb
RPL
Iver Holtersvei 2, NO-1410 Kolbotn, Norway

Abstract. What is 'best practice' for an engineering process? How good is your current set of development, maintenance and service processes? How can we decide exactly which processes we are going to adopt in our organization, for example in a CMMI implementation?

It is the assertion of this paper that such questions are often dealt with without explicit and quantified regard to the full set of real, and well-defined business needs, as well as often not taking into consideration the current processes and the issues of changing them. We too often carry out and change processes because we are told to, not because there is a clearly defined need to do so.

Introduction

A rational evaluation process would continuously match our continuously evolving set of business, technical and engineering objectives to a set of engineering processes. It would do so on a multi-dimensional and numeric basis:

- We would *quantify* our engineering process objectives.
- We would *estimate* the impacts we can expect from new and changed engineering processes.
- We would measure in practice the impact of new processes.
- We would decide which processes are good and which are bad, based entirely on how they worked in our particular environment.

I am sure the reader agrees with the desirability of numeric principles, even if they are in doubt about how to practice them. However, how many people can show that their organization operates on this rational principle? What is often missing is the ability to articulate engineering organizational objectives as quantified goals. My observation of a large number of multinational engineering organizations convinces me that even the best and most senior managers are not trained in how to do this. The good news is that given a little help and some examples, they are willing to define their needs quantitatively. The aim of this paper is to outline how to quantify such objectives and to outline how to use such quantification to achieve better processes.

Ten Principles for Processes

Here are ten principles for evaluating engineering processes:

1. Processes are 'good' to the degree in practice that they satisfy the organizational objectives.

2. When organizational objectives change, or are satisfied by other means, the usefulness of a process may decline or disappear.

3. Processes that are equivalent in their performance effects can be distinguished by their use of the limited, budgeted resources (for example, human resources, financial costs and time).

4. We can estimate the efficiency (value to cost ratio) of a process based on experience with it, or similar processes; but we cannot be certain of the process impacts until we measure them in place within our organization.

5. Just because we have measured a process as being efficient once does not mean that its efficiency will not change for better or for worse over time or in different circumstances.

6. If the process efficiency does not in practice meet the estimated levels of efficiency, then one possible cause is malpractice.

7. Processes should be implemented early in small evolutionary steps and their effect measured before scaling up.

8. Process impacts will always be on multiple critical organizational performance and cost characteristics; so we must not evaluate them in a single dimension alone.

9. The entire justification of any process should rationally be the efficient effects on the organizational objectives. A process should never be mandated as 'best practice', but should forever be monitored for its justification.

10. Before implementing any new process, the resources to implement and to maintain it should be created by conscious and specific removal of the less efficient processes that they will replace (Conner 1998).

These principles shall now be discussed in greater detail in the following sections.

1. Processes are 'good' to the degree in practice that they satisfy the organizational objectives.

There is an implication in this principle's statement. It implies that all organizational performance objectives can be expressed quantitatively with a defined scale of measure, a target level and a deadline.

Here are some interesting examples of potential organizational objectives:

- Time To Market
- Predictability of Time To Market
- Lead Time
- Productivity
- Quality Levels
- Transportability (Outsourceability)
- Competitiveness
- Risk Avoidability/Controllability
- Prioritization Ability
- Customer Satisfaction

Here are some examples of defining some corresponding scales of measure for these concepts using Planguage (Gilb 2005):

Time To Market:

Scale: Time from Product Concept Approval to availability on defined [Market].

Predictability of Time To Market:
Scale: Percentage (%) overrun of actual Project Time compared to planned Project Time.
Project Time: Defined: Time from the date that Toll Gate 0 (TG0) passed, or other defined [Start Event] to the Planned- or Actually- delivered Date of all Specified Requirements.
Specified Requirements: Defined: Written approved quality requirements for products with respect to planned and constraint levels with qualifiers [when, where, conditions]. And, other function, resource and design requirements.
Meter: The Productivity Project or Process Owner will collect numeric data, concerning the required levels on this Scale, from all projects, or make appropriate estimates, and put them in the Productivity Database.

Lead Time:
Scale: Months from TG0, to successful first use for major workstation package.
Scale: Average Time from defined [Inception Point] until defined [Delivery Point].

Productivity:
Scale: Net Profit per financial year derived from defined [New Products or Services].

Quality Levels:
Scale: Percentage (%) +/- deviation from defined [Agreed Quality Attributes].

Process Transportability:
Scale: The cost as a percentage (%) of affected persons Gross Annual Cost, for successfully learning to deploy a defined [Process] to a defined [Capability Level].

Competitiveness:
Scale: Average percentage (%) impact on a defined set of [Competitiveness Measures] within a year from First Deployment in a defined [Organization].

Risk Controllability:
Scale: The percentage (%) probability that defined [Project or Product Requirements] can be delivered within defined [Percentage of Target Levels] under conditions of defined [Risks].

Prioritization Ability:
Scale: The average speed in Days that a new [Priority Item] can be effectively acted upon.

Customer Satisfaction:
Scale: Average survey result on a scale of 1 to 6 (best).

There is, in addition, a second implication to this principle; it introduces the idea that we can evaluate the degree of expected impact on organizational performance characteristics, and on resource budgets. This can be estimated initially using Impact Estimation (IE) tables (Gilb 2005). Impact Estimation is a method that forces us to ask, “Exactly how much will this design (in this case, a process or process change) impact my unfulfilled objectives?”

Table 1: Example of a client's initial draft setting the objectives that their engineering processes must meet. The right-hand side of the table shows how these objectives relate to defined senior management financial planning (The table has been modified for confidentiality purposes)

Business Objective	Measure	Goal [200X]	Stretch [200X]	Volume	Value	Profit	Cash
Time to Market	Normal project time from TG0 to TG5	< 9 months	< 6 months	X		X	X
Product Range	Minimum BoM	< 90$	< 30$	X		X	X
Platform Technology	# shipping	4	6	X		X	X
Units	# shipping	> 4M	> 5M	X		X	X
Operator Preference	Operator issues	1	2	X		X	X
Commoditization	Switching time	< 2 years	< 1 year		X	X	X
Duplication	Share of code in best selling device	> 90%	> 95%		X	X	X
Competitiveness	Major feature comparison	Same	Better	X		X	X
User Experience	Key use cases superior vs. competition	5	10	X	X	X	X
Downstream Cost Saving	Project ROI	> 33%	> 66%	X	X	X	X
Other Country	Shares of sales	> 50%	> 60%	X		X	X

Then, using the best available basis for estimation (such as experience, pilots or guesses) we estimate the degree that we will move towards the unfulfilled target levels for our objectives. To give an idea of the kind of information that IE requires, here is a simplified IE statement:

> Process A will deliver a 10% increment towards Objective B bringing us to 90% of our target level with uncertainty ±2%, based on Evidence X.

We need to create an IE table when we want to evaluate a set of processes against a set of objectives by making a set of statements like this (See Table 1). This IE table gives us an overview of the totality of our process change environment. It can also be used to track actual results after implementation (reality versus estimates).

Table 2: A set of 12 proposed engineering processes, for about $100,000,000 of investment projected over time, are evaluated theoretically for their impact on 13 business objectives (as defined in Table 1 above). This real example is altered substantially to protect confidentiality. It appropriately ignited the imagination of top management to really plan their engineering business in a quantified manner. Notice the overall impact to cost ratio (ROI Index) is estimated for each process. The actual definitions of the processes are elsewhere and are confidential. But that detail would be needed to estimate and to check these estimates

		.............................Deliverables						
		Telephony	Modularity	Tools	User Experience	GUI & Graphics	Security	Enterprise
Business Objective								
Time to Market		10%	10%	15%	0%	0%	0%	5%
Product Range		0%	30%	5%	10%	5%	5%	0%
Platform Technology		10%	0%	0%	5%	0%	10%	5%
Units		15%	5%	5%	0%	0%	10%	10%
Operator Preference		10%	5%	5%	10%	10%	20%	10%
Commoditization		10%	-20%	15%	0%	0%	5%	5%
Duplication		10%	0%	0%	0%	0%	5%	5%
Competitiveness		15%	10%	10%	10%	20%	10%	10%
User Experience		0%	20%	0%	30%	10%	0%	0%
Downstream Cost Saving		5%	10%	0%	10%	0%	0%	5%
Other Country		5%	10%	0%	10%	5%	0%	0%
Total Contribution		90%	80%	55%	85%	50%	65%	55%
Cost (£M)		0.49	1.92	0.81	1.21	2.68	0.79	0.60
Contribution to Cost Ratio		**184**	42	68	70	19	82	92

2. When organizational objectives change, or are satisfied by other means, the usefulness of a process may decline or disappear.

Organizational objectives (called business objectives in the tables) are subject to pressures that demand constant tuning, updating and even radical change as soon as possible. It is simple enough to change target number and due dates in a set of objectives. But stopping the process change ‘ship in mid Atlantic’ is quite another problem. Major investments in contracts and training may have been set in motion. Maybe they are now obsolete!

This argues for implementing processes with the following considerations:

- Highest value-to-cost processes first (See ‘Contribution to Cost’ in Table 2, which measures a notion of ROI (Return On Investment))
- Highest risk-of-obsolescence processes last

You would have to understand the volatility of the objectives target levels to determine that risk.

Large and costly processes need to be decomposed into smaller, early implementations, and high-value low-volatility sub-processes need to be prioritized. This is the Evolutionary Project Management method (Evo) as described in (Gilb 2005 Chapter 10).

3. Processes that are equivalent in their performance can be distinguished by their use of the limited, budgeted resources (for example, human resources, financial costs and time).

The primary consideration for an engineering process is its ability to help us reach our target levels. In other words, we are interested in its contribution to achieving the goal and stretch levels for the business objectives (see Table 1). If it does not do that, it does not matter how cheap it is. The second consideration is that the costs for all types of resource are within budgets, or profitability limitations.

In addition, a single process should not steal resources from more profitable processes. Decisions about what to spend on process implementation cannot be made in isolation from all the other processes that use concurrent resources. The Impact Estimation table helps us get a view of all of these considerations.

4. We can estimate the efficiency (value to cost ratio) of a process based on experience with it, or similar processes; but we cannot be certain of the process impacts until we measure them in place within our organization.

Estimations are guesses, and we all know they are not for sure. Consequently we cannot bind ourselves (in contracts, and corporate plans) to *full* implementation of a particular process until it is proven to deliver to expectations in practice.

This requires evolutionary implementation, for example on a project-by-project basis, or even in small groups within larger projects. If the estimates are validated by practical experience, we can ramp up. Otherwise we may have to drop the new engineering process, replace it with another or tune it to work properly.

5. Just because a process has been measured as being efficient once does not mean that its efficiency will not change for better or for worse over time or in under a different set of circumstances.

We should have a commitment to long-term measurement that the processes are still working with the impacts they initially were validated to have. Processes may well have to be reinforced (for example, with training, motivation, and management support), and they may well have to be retuned.

Such process monitoring does not have to be expensive or frequent. For example, sampling should be sufficient. And we can at least measure new people using the process (for example, new hires and new projects). This measurement is the basic cost of making sure we get value for process money. This cost has to be planned and budgeted as a part of understanding whether we should use the process at all. If we can't afford to check that it works, then we can't afford to do it. You can consider it in the same light as financial auditing.

6. If the process efficiency does not in practice meet the estimated levels of efficiency, then one possible cause is malpractice.

Many processes are complicated, and malpractice of an apparently small detail, sometimes known as 'cutting corners', will be tempting for those who do not know the consequences, or do not care. So, just because a highly valued practice does not appear to give the results experienced elsewhere, does not mean it will not work for you. You might have to bring in expertise on successful use of the process (or read the process 'recipe' more carefully).

My favorite example is within the Specification Quality Control method, also known as Inspection (see Gilb 2005 Chapter 8). Most people who claim they are using the method do not practice checking at the optimum checking rate, which is about one page per hour for technical documents such as requirement specifications. They try to check 50-page documents in an hour or two! This has the inevitable consequence that the defect detection rate can fall by at least an order of magnitude. They then falsely conclude that the process is no good at finding defects! Or worse, that their documents are of sufficient quality (see Figure 1).

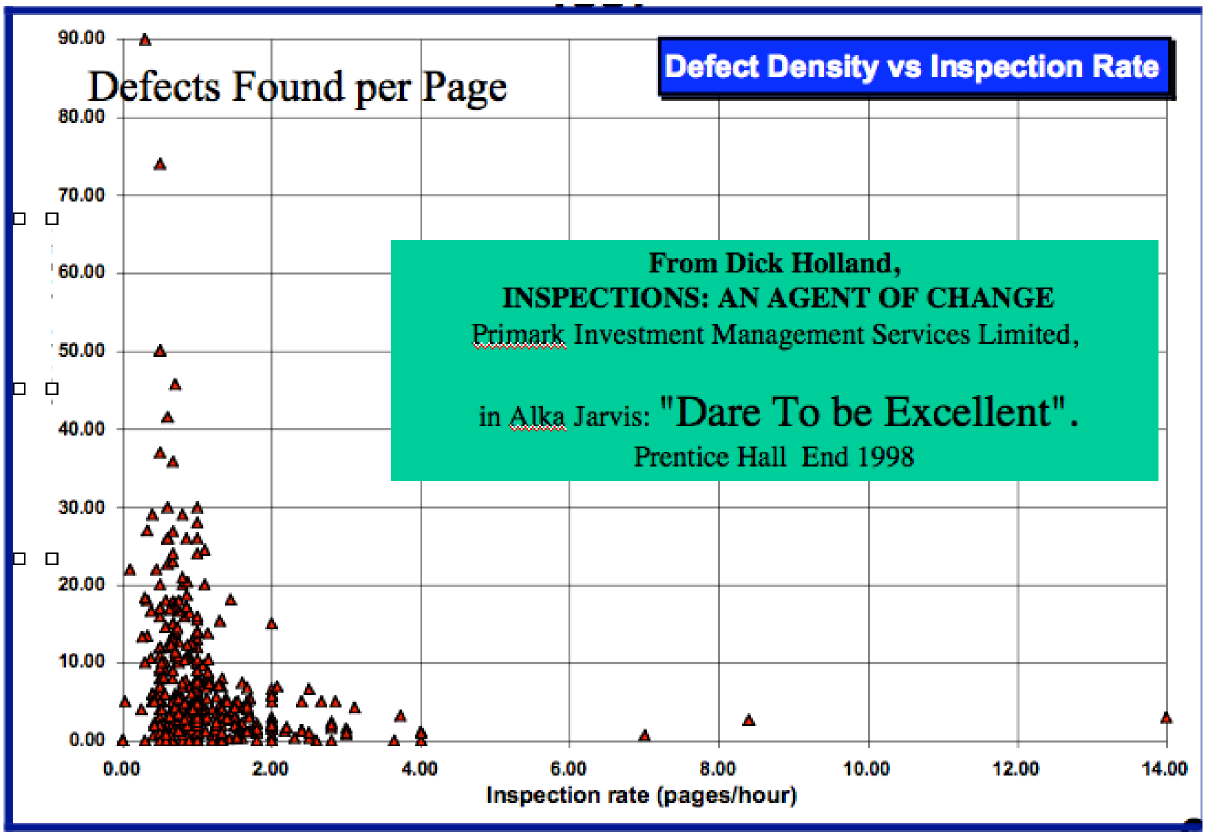


Figure 1. A report from a client, who took engineering process control seriously. Most document inspections are run at an appropriate rate for discovering defects. The few that are not (such as at a rate of 14 pages per hour) prove the point that there are limits to the speed of checking things! (Shah-Jarvis et al. 1998)

7. Processes should be implemented early in small evolutionary steps and their effect measured before scaling up or attempting to combine with other processes.

There are always pressures and temptations to install exciting new engineering processes all at once in an organization. I have had clients pressure us into training hundreds of engineers in a new process within a year, when there were no instances where the process was working as it ideally should do. Our audit showed that after a year. They had no fully successful model instance to follow! Corners were being cut, pressured by managers who did not take the week's training that their engineers got from us. Ultimately the client reported about $10 million effort savings from the use of our process teachings. But I am convinced they could have done an order of magnitude better as HP did (Bob Grady, papers and books) if they had followed out persistent advice to master the process locally, and spread the correct process, and its measurements. They actually argued that massive training and bottom up process change was their corporate culture. Maybe it was, but it felt wrong. Brute force can work, but I do not admire its efficiency!

8. Process impacts will always be on multiple critical organizational performance and cost characteristics; so we must not evaluate them in a single dimension alone.

It is a general characteristic of any design or architecture, including any process design, that when you install it in a system (organization) it tends to have some impacts on *many* critical performance and cost attributes, and significant impacts on more than one.

So, evaluation of any major engineering process based on a single dimension (for example, productivity) is doomed to be incomplete and risky. The additional unmeasured side effects can be positive, but they can just as well be negative. In fact, they can be intolerable effects. You therefore need to estimate all side effects, and you need to measure them in your evolutionary pilot installations of the process. Impact Estimation tables are a good practical tool to use to keep track (as they can record both the initial estimates and the feedback from actual practice).

9. The entire justification of any process should rationally be the efficient effects on the organizational objectives. A process should never be mandated as 'best practice', but should forever be monitored for its justification.

We need to stop the dogmatic culture of mandating processes because they are 'known best practices' or because they are in some set of key practices in some standard, such as CMMI. This is primitive and people who do it are not real engineers! There needs to be a clear corporate policy such as the following:

> "All engineering practices must always prove themselves numerically in terms of our plans and needs."

Another approach is to be clear that engineering communities are charged with well-defined engineering results (for example, delivering specific quality levels on time) and they are free to use any engineering process they want that gets them there, and they are (with some unfortunate political exceptions) free to avoid using any engineering processes which prevent them from reaching their objectives.

The best CEOs and CTOs make this abundantly clear to their engineers. For example, John Young, CEO at HP in his '10X' policy stated that engineers would be supported in getting ten times better qualities within the decade with whatever methods worked! They got 9.95 X better by the end of the decade. This is an excellent example of inspiring 'the troops' and supporting them in finding the right technical solutions. None of this "You will get CMMI Level 3 this year" that I have seen some CEOs guilty of. I know of one CEO who realized his mistake and changed to giving his CTO a bonus based on the measurable engineering productivity that the Level 3 was supposed to bring about!

10. Before implementing any new process, the resources to implement and maintain it should be created by conscious and specific removal of the less efficient processes that they will replace (Conner 1998).

Conner makes a major point in his excellent book that we have to get pretty formal about reducing the load of process change (overload) on most engineer's shoulders. The world's greatest process for you will not be successfully implemented unless senior management clearly removes the burdens of past process failures. We need to create a human capacity for people to prioritize the best changes, the ones that are needed now, and will really work.

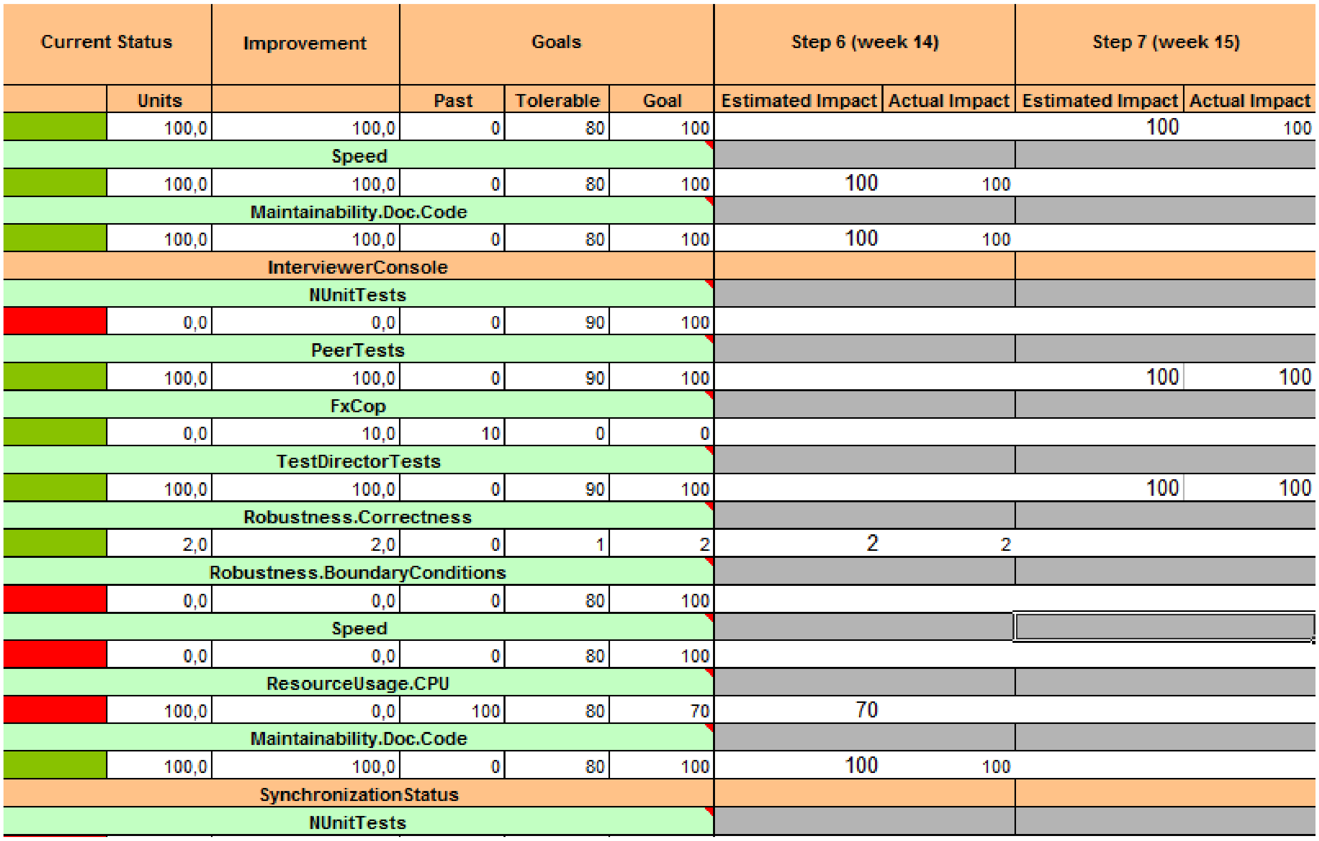
We are going to have to unceremoniously dump masses of process baggage that cannot prove its necessity based on measured facts, and with relation to current objectives.

A Case Study

One client, Future Information Research Management (FIRM) demonstrates putting the process principles given in this paper into practice. They develop the product Confirmit, a successful Web Opinion Survey Product. They use agile methods. See (Gilb and Johansen 2005) for further details. I shall describe here one practice they use that I particularly like. They use one of every month, and shift their normal customer/user focus to measurable improvement (by any means they see fit) of *the internal product qualities*, as viewed by the internal stakeholders (that is the developers, maintainers and testers). They call it their ‘Green Week’.

They are completely driven by 12 long-term product quality goals (for example, testability and maintainability). They can use any process or technology that in fact delivers the engineering goals. The ‘grass roots’ people themselves can discover and try out the techniques. Management does not want to tell them what to do. Developers love it and have “empowered creativity”.

Table 3: An Impact Estimation table used at the FIRM Company to specify requirements (Goals), to estimate impact of engineering process changes, to record actual impact, and to keep track of long-term progress towards those goals, cumulatively and incrementally. A ‘Green Week’ is carried out once a month to serve the internal stakeholders (the developers and maintainers) (Gilb and Johansen 2005)



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

Conclusions

We can and should think about *engineering processes* in terms of their multidimensional contribution to our defined engineering process objectives .

We need to be continuously aware of necessary changes in our objectives, and the corresponding need to change engineering processes to satisfy them.

We need to be fact-driven by what the engineering process changes actually deliver, and we need to let the 'grass roots' engineers find out what works in practice.

Give your engineers a fair chance to implement ideas, give them the time and money necessary to do it.

References

Conner, Daryl R., *Leading at the Edge of Chaos, How to Create the Nimble Organization*, Wiley, 1998. ISBN 047129557.

Gilb, Tom, *Competitive Engineering: A Handbook For Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage*, Elsevier Butterworth-Heinemann, 2005. ISBN 0750665076.

Gilb, Tom and Johansen, Trond, "From Waterfall to Evolutionary Development (Evo): How we rapidly created faster, more user-friendly, and more productive software products for a competitive multi-national market", INCOSE Proceedings, July 2005. The 'Green Week' slide is from a corresponding presentation (without paper) by Trond Johansen, at ROOTS Conference, Bergen, Norway, May 2005.

Grady, Robert B., "Successful Software Process Improvement" Prentice Hall PTR; 1st edition (May 1997), Page 253

Grady, Robert B., and Tom Van Slack. Hewlett-Packard. July 1994. Key Lessons in Achieving Widespread Inspection Use. *IEEE Software*. Pages 46-58. This paper also appears in (Wheeler, Brykczynski and Meeson 1996). This paper is available for a fee at http://csdl.computer.org/comp/mags/so/1994/04/s4046abs.htm

Grady, Robert B. August 1996. Software Failure Analysis for High Return Process Improvement Decisions. *Hewlett-Packard Journal*. Vol. 47. No. 4. Pages 15-24.
Available as Adobe Acrobat pdf file at http://www.hpl.hp.com/hpjournal/96aug/aug96a2.htm
****

Shah-Jarvis, Alka, Hayes, Linda and Jarvis, Alka, *Dare to Be Excellent: Case Studies of Software Engineering Practices at Work*, Prentice Hall, 1998. ISBN 0130811564.

Biography

Tom has been an independent consultant, teacher and author since 1960. He works mainly with multinational clients; helping improve their organizations, and their systems engineering methods.

Tom's latest book is 'Competitive Engineering: A Handbook For Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage' (Summer 2005).

Other books are (with Dorothy Graham) 'Software Inspection' (1993) and 'Principles of Software Engineering Management' (1988). His 'Software Metrics' book (1976, Out of Print) has been cited (Radice, IBM) as the initial foundation of what is now CMMI Level 4.

Tom's key interests include business metrics, evolutionary delivery, and further development of his planning language, 'Planguage'. He is a member of INCOSE and is an active member of the Norwegian chapter NORSEC.

Email: Tom@Gilb.com

URL: http://www.Gilb.com

Thanks for editing help to Lindsey Brodie, Middlesex University, London, UK

Version 14th April 2006 05:38