INTRODUCTION: I made this collection primarily by cutting quotations from 'Competitive Engineering'. Not least I hope that those who currently enjoy quoting principles on Twitter, will have better access to these ideas. ☺
I also hope to put a focus on the concept of useful principles.
In my 1988 book, Principles of Software Engineering Management (I suspect one of the few books with 'principles' in the title, that actually has some principles in it?), someone once counted that I had about 120-140 specific principles. I hope to, one day (soon? Encourage me), publish that collection on the web. I can tell the reader that when writing CE, I did not 'peek' at the PoSEM principles, hoping to enhance my own creativity.
Have fun!
Tom

See after end of the 100 principles for © and quotation rights.

Principle **Concept *208 May 28, 2003**

A principle is a short basic statement, which summarizes and teaches basic philosophy or the pragmatics of a method.

Notes:
An excellent collection of systems engineering heuristics are found in 'The Art of Systems Architecting' by Mark Maier and Eberhardt Rechtin [MAIER02].
2. An excellent discussion of the importance of the heuristic inthe engineering process will be found in KOEN03.
Related Concepts [Principle *208]:
• Heuristics
Type [Principle *208]: Guidance.

from 'Competitive Engineering' Book 2005

1.6 Principles: Generic Project <- CE pages 23-24

Principles are teachings, which you can use as guides to sensible action. Here is a set of fundamental principles:

1. The Principle of 'Controlling Risk'

There is lots of uncertainty and risk of deviation from plans in any project. You cannot eliminate risk. But, you can document it, plan and design for it, accept it, measure it and reduce it to acceptable levels. You may want to avoid risk, but it doesn't want to avoid you.

2. The Principle of 'Storage of Wisdom'

If your people are not all experienced or geniuses, You need to store their hard-earned wisdom in your defined process.
Capture wisdom for reuse,
Fail to write it, that's abuse!

3. The Principle of 'Experienced Geniuses' If you do have any experienced geniuses, don't just let them save projects; They should share their wisdom with colleagues, on how to avoid failures.
Those who learn the hard way, Should share their easy way.

4. The Principle of 'Grass Roots Experience'

Your grass roots people will know what is wrong with your work standards, So let them suggest improvements, every day.
The soldier who has the boot on knows where it pinches.

5. The Principle of 'Short and Sweet'

Keep your standards short and sweet, A single page will do the feat.
Brevity is the soul of wit, All essentials, a page do fit.

6. The Principle of 'Don't Refuse to Reuse'

Reuse good specifications, and don't repeat them,
Once said suffices, no repetitious vices.
Write once, use many.

7. The Principle of 'High Standards'

Have high standards for your work process entry, to save yourself grief,
Have high standards for your work process exit, to your friends' great relief.
Note work standard conditions for success, Respect them; even in duress.

8. The Principle of 'Quality In, From the Beginning'

Quality needs to be designed into processes and products, Cleaning up bad work is a loser, but cleaning early is better than late.
A stitch in time still saves nine, But an ounce of prevention is still worth a pound of cure.
9. The Principle of ‘No Simpler’
The optimum guidance lies somewhere between anarchy, And too much bureaucracy. Things should be as simple as possible, But no simpler. *See footnote 5*

- *Footnote:5 ‘’Physics/theories/things should be as simple as possible, but no simpler’’. Reputed quote of Albert Einstein. Nobody seems able to prove he actually said it, but it is acknowledged to be in his spirit. Calaprice, Alice [Editor]. 2000. The Expanded Quotable Einstein. Princeton University Press. ISBN 0-691-07021-0.*

10. The Principle of ‘Intelligent Insubordination’
A work process ‘standard’ isn’t a law, just good advice, Ignore it, if you’ve better ‘words from the wise’.
Rules were made to be broken wisely.

2.6 Principles: Requirement Specification <- CE pages 64-65
1. The Principle of ‘Results Beat All’
The top strategy is ‘getting the stakeholder results’.
Meeting requirements is more fundamental than any other process or principle.
2. The Principle of ‘Goodies Control beats Bean Counting’
Focus on getting the Goodies. Their costs will be forgiven.
The main point of any project, or change effort, is to improve stake-holder benefits. The benefits must be at least as well-controlled as the resources needed to get them. Otherwise the benefits will lose out, at the hands of the always limited, clearly budgeted resources.
3. The Principle of ‘Reasonable Balance’
Reach for dreams, but don’t let one of them destroy all the others.
You cannot require an arbitrary set of requirements. There must be balance between performance requirement levels, resources available and available design technology.
4. The Principle of ‘Unknowable Complexity’

You must feed a lion to find out how hungry it is.
You cannot have correct knowledge of all the interesting requirements' levels for a large and complex system in advance. You cannot know which requirements are needed, and which are realistic, until you have some practical experience with a real system with real people using it.
5. The Principle of 'Specification Entropy'
Even gourmet decays.
Any requirement or design specification, once made, will become gradually less valid, as the world, for which they were intended, will change over time.
6. The Principle of 'Critical Values'
If you don't find the critical requirements, they will find you!
You must identify all potentially requirements for all stakeholders or you risk losing profitability, or even system failure.
7. The Principle of 'How Good' and 'How Much' before 'How'
All performance requirements and resource requirements must be stated before any design idea can be fully and properly evaluated.
8. The Principle of 'Gap Priorities'
The least fulfilled requirement attributes become our current priorities.
By calculating the 'gap' between current real levels of performance delivered and the required levels, we can assume that the biggest unfilled 'gap' in meeting our targets is our current greatest priority. For example, you cannot know now if you will be hungrier, thirstier or more tired a week from now. But wait a week and you will know which need has priority.
9. The Principle of 'Stop the World, I Want to get Off'
There is no final set of real-world requirements; freezing the specifications will make your real problems worse than any problems caused by updating them.
10. The Principle of 'Eternal Projects' Survival is a lifetime project.,
The process of delivery of results has no end, if you are in competition for survival. Note 6
Note 6 Based on the wisdom of W. Edwards Deming.

3.6 Principles: Function and Function Requirement Specifications

These are principles for recognizing what is, and is not, a function and also for working with functions.
1. The Principle of 'What Function?'

Function is ‘what’ a system does, never ‘how well’ it does it or ‘how it does it so well.’

2. The Principle of ‘Thing with Attributes’

A function is the thing, which has the performance and resource attributes attached to it.

3. The Principle of ‘Living Map’

Function specification is sometimes best done by declaring the existing system to be a living map.

4. The Principle of ‘Part of Totality’

Functions are always part of some larger function and can probably be described by their own sub-functions.

5. The Principle of ‘Each to their Own’

Different functions require different performance and resource attributes; so, one reason we specify the functions is to identify and distinguish their required attributes.

6. The Principle of ‘Timing’

Different functions can be delivered to customers at different times, so another reason to specify functions is to know ‘what to do when.’

7. The Principle of ‘Conditional Function’

Some functions may not be necessary, except under specified conditions or events, and these conditions should be specified and exploited in project planning. You don’t have to do what is not yet required!

8. The Principle of ‘Room with a View’

A function definition is not absolute; it is a viewpoint, and many overlapping function views can be made and used fruitfully to satisfy different needs.

9. The Principle of ‘Terrain does not change with the Maps’

The real system does not change just because you document function viewpoints and function hierarchies: correctly or incorrectly.

10. The Principle of ‘False Function Foils Fruits’

If you mistakenly request a design, as basic functionality, you will limit your ability to improve the design to give better competitive attributes.

Alternatively,

Don’t request ‘functions’ which are really ‘designs for performance’, You might not get the performance you really want.

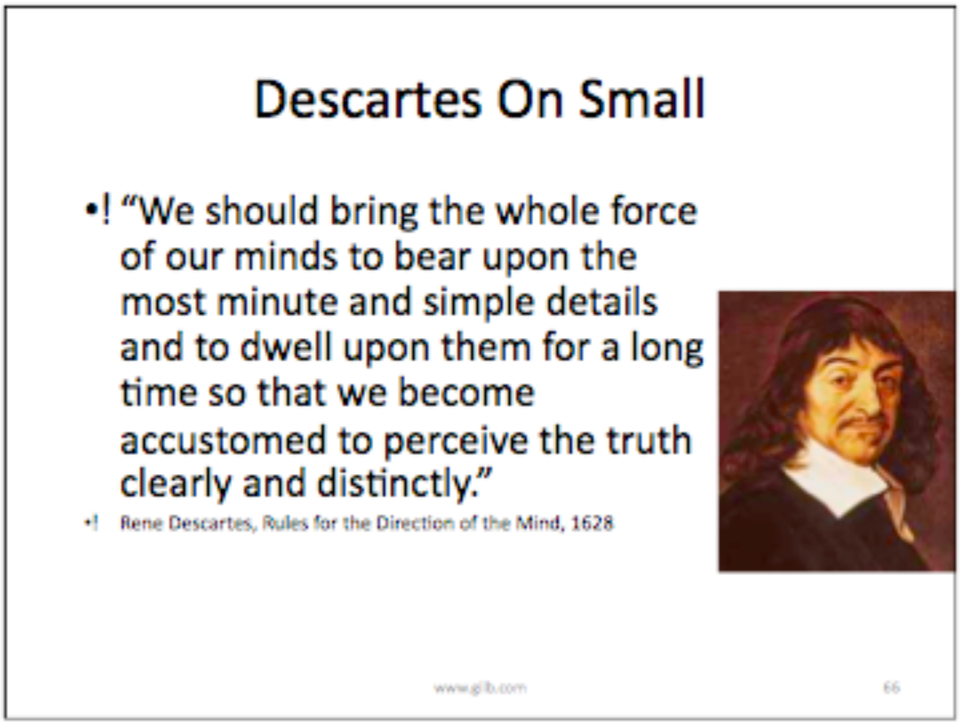
<u>4.6 Principles: Performance Requirements</u> <- CE pages 126-7

1. The Principle of ‘Bad numbers beat good words’

Poor quantification is more useful than no quantification; at least it can be improved systematically.
2. The Principle of ‘Performance quantification’
All performance attributes can be expressed quantitatively, ‘qualitative’ does not mean unquantifiable.
3. The Principle of ‘Threats are measurable’
If the lack of a performance attribute can destroy your project, then you can measure it sometime; the only issue will be ‘‘how early?’’
4. The Principle of ‘Put up or shut up’
There is no point in demanding a performance requirement, if you cannot pay or wait for it.
5. The Principle of ‘Deadline or die’ There is no point in demanding a performance requirement, if you would always give priority to something else, for example, a deadline.
6. The Principle of ‘Dream, but don’t hold your breath’ There is no point in demanding a performance requirement, if it is outside the state of your art.
7. The Principle of ‘Benchmarks and targets’
Numeric past ‘history’ levels and numeric future requirement levels together complete the performance requirement definition of relative terms like ‘improved’.
8. The Principle of ‘Scalar priority’
In practice, the first priority will be survival, The second priority will be avoiding failure, The third priority will be success, And the required levels for all of these will be constantly changing.
9. The Principle of ‘Many-splendored things’
Most performance ideas are usefully described by several measures of goodness.
10. The Principle of ‘Limits to detail’
There is a practical limit to the number of facets of performance you can define and control. It is far less than the number of facets that you can imagine might be relevant. (Try a limit of just the Top Ten!)

5.6 Principles: Scale Definition <- CE pp 151-152
1. The Principle of ‘Defining a Scale of Measure’ If you can’t define a scale of measure, then the goal is out of control. Specifying any critical variable starts with defining its scale.
2. The Principle of ‘Quantification being Mandatory for Control’

If you can’t quantify it, you can’t control it.
If you cannot put numbers on your critical system variables, then you cannot expect to communicate about them, or to control them.
3. The Principle of ‘Scales should control the Stakeholder Requirements’
Don’t choose the easy Scale, choose the powerful Scale.
Select scales of measure that give you the most direct control over the critical stakeholder requirements. Choose the Scales that lead to useful results.
4. The Principle of ‘Copycats Cumulate Wisdom’
Don’t reinvent Scales anew each time – store the wisdom of other Scales for reuse. Most scales of measure you will need will be found somewhere in the literature or can be adapted from existing literature.
5. The Cartesian Principle Divide and conquer said René – put complexity at bay.
Most high-level performance attributes need decomposition into the list of sub-attributes that we are actually referring to. This makes it much easier to define complex concepts, like ‘Usability’, or ‘Adaptability,’ measurably.



6. The Principle of ‘Quantification is not Measurement’
You don’t have to measure in order to quantify!
There is an essential distinction between quantification and measurement. ‘‘I want to take a trip to the moon in nine picoseconds’’ is a clear requirement specification without measurement.’’ The well-known

problems of measuring systems accurately are no excuse for avoiding quantification. Quantification allows us to communicate about how good scalar attributes are or can be – before we have any need to measure them in the new systems.

7. The Principle of 'Meters Matter'

Measurement methods give real world feedback about our ideas. A 'Meter' definition determines the quality and cost of measurement on a scale; it needs to be sufficient for control and for our purse.

8. The Principle of 'Horses for Courses'

Different measuring processes will be necessary for different points in time, different events, and different places.5

9. The Principle of 'The Answer always being 42' (Hitchhikers Guide!)

Exact numbers are ambiguous unless the units of measure are well-defined and agreed. Formally defined scales of measure avoid ambiguity. If you don't define scales of measure well, the requirement level might just as well be an arbitrary number.

6.6 Principles: Resource Requirements <- pages 176-177 CE

1. The Principle of 'Many Critical Risks' There are many resource, performance and condition dimensions critical to any system, not just one or a few.

2. The Principle of' You Can't Have It All, Trade-offs are a Necessity'

Fixing the required level of one resource dimension arbitrarily can only be done at the probable expense of other attributes.

3. The Principle of 'You Get What You Pay For' It is really the availability of resources, which limits the levels of performance that can be delivered in practice

4. The Principle of 'Attribute Balance'

Once you have found a balance between performance and costs, management cannot cut the financial budget, people or time without negative consequences.

5. The Principle of 'The Cost of Perfection'

Perfect quality costs infinity.

6. The Principle of 'The Rolls Royce'

Near-perfect performance levels cost more than most people would pay.

7. The Principle of 'Natural Ambition'

The pressure on resources will always be at a 'level of discomfort', not to say downright intolerable – this is a natural management strategy to find out how far they can push!

8. The Principle of 'The Traffic Bottleneck Illusion'
Increasing your allocated resources will not relieve the pressure on you, but only raise that sponsor's expectations. Removing one bottleneck serves mainly to discover others.
9. The Principle of 'Really Useful Resource Management'
The only practical way to control costs and performance in large complex dynamic systems is by early, frequent realistic evolutionary feedback on costs, and consequent adaptation to realities.
10. The Principle of 'Shifting Conflicts'
Conflicts amongst budget targets, performance targets and design ideas are natural; there's no blame. You just keep resolving them: it's the name of the game. Budget constraints will always exist and, will always be subject to change

7.6 Principles: The Design Engineering Process <- CE 210-211

1. The Principle of 'Design Ideas are only as Good as the Requirements Satisfied'
Design ideas cannot be correctly judged or validated except with respect to all the performance and cost requirements they must satisfy.
2. The Principle of 'The Best Chess Move'
You should try with each increment of design specification or design implementation, to get the best possible satisfaction of your unsatisfied performance requirements, from your unused cost budgets.
3. The Principle of 'Results Beat Theory'
Design ideas are only as good as their real results, not their intent.
4. The Principle of 'Early Surprises'
You never know how it works, until you have actually tried out a design idea in practice. Get surprised as early as possible.
5. The Principle of 'It's Not Just What You Do, It's How You Do It'
Design ideas must try to exercise control over both design content and design implementation. The devil is in the details!
6. The Principle of 'Good is Not Always Good Enough'
A 'good' design idea might not be good enough to meet all your targets on time.
7. The Principle of 'Designs should have Good Return on their Investment'
'Good' design ideas might cost too much, sooner or later.
8. The Principle of 'Sneaky Gremlins'

Apparently 'good' design ideas might have subtly-hidden nasty side effects. Estimate them, know when you don't know them, measure them, and don't assume they won't hurt you! They will show you no sympathy!

9. The Principle of 'Design Beats Test'

Design performance 'in', and design 'to control' costs: You cannot test quality into a badly designed system.

10. The Principle of 'Eternal Vigilance for the Butterfly Effect'

You never finally know about a design idea's effects; Tomorrow's slightest change might ruin your whole project. Even initially successful designs might have to be adjusted for growth and change.

8.6 Principles: Specification Quality Control (SQC) <- page 246 CE

1. The Principle of 'Illegality'

'Defects' are objective violation of accepted written rules.

2. The Principle of 'Majors are the pay off'

Major defects are the only economically interesting defects.

3. The Principle of 'Keen to be seen clean'

The main purpose of SQC is to measure that the specification is clean enough: not to clean up a specification that isn't.

4. The Principle of 'Cleanup your own mess'

Specification cleanup is the writer's responsibility, before SQC.

5. The Principle of 'Prevention is better than cure'

There are many effects of SQC, but the most useful are learning to avoid defects caused by bad process, and committed by the writer.

6. The Principle of '50% effectiveness'

History shows that you can only expect to find and fix about half the defects that are there.

7. John Craven's Principle (within Hewlett Packard)

The team is there to make the "writer look like a hero."

8. The Principle of 'Magnificent Profitability'

The expected return on investment for SQC is at least 'ten to one.'

9. The Principle of 'Client-Server'

The writer is the client and the checkers serve as advisors.

10. The Principle of 'The Pilot in Command'

The team leader is responsible for the SQC process.

Good execution of a badly executed specification will tend to execute you!

9.6 Principles: Impact Estimation. <- CE pp 278-279

1. The Principle of 'Words being difficult to weigh'

Non-numeric estimates of impact are difficult to analyze and improve upon. A design idea described as ‘excellent’ could actually be worse than another merely described as ‘good.’

2. The Principle of ‘Doubtful digits are better than none’

A bad numeric estimate, and its definition, can still be systematically criticized and improved. In fact, a random number is a better starting estimate than flowery, descriptive words.

3. The ‘Evident’ Principle

Estimates without sources, evidence and credibility are not evident.

4. The Principle of ‘Uncertainty in no uncertain terms’

The uncertainty estimate is at least as important as the main estimate.

5. The Principle of the ‘Seat Belt’ A safety margin is as necessary with uncertain estimates, as a seat belt is with uncertain traffic.

6. The Principle of ‘Profitable Proposals’

The value of an idea is how well it meets objectives. The net value considers the costs too.

7. The Principle of ‘the Swiss Army Knife’

Impact Estimation is a multi-purpose method. It can help you in many situations: to evaluate, to compare, to present, to argue, to destroy, to find weaknesses, to cut fat, to see risk, to prioritize, to sequence and more.

8. The Principle of ‘Always Useful’

Impact Estimation can assist a project throughout its lifecycle – from identifying requirements to assessing feedback data from implemented systems.

9. The Principle of ‘Multiplicity’

When stakeholders have multiple requirements, then we need to evaluate multiple design options against all those requirements including considerations of value, in order to make a reasonable choice.

10. The Efficiency Principle

When real life has many stakeholder values, and many cost constraints, then evaluation of designs (strategies) must be done with respect to both the values and the costs.

10.6 Principles: Evolutionary Project Management <- CE p. 310

1. The Principle of ‘Capablanca’s next move’

There is only one move that really counts, the next one.

2. The Principle of ‘Do the juicy bits first’

Do whatever gives the biggest gains. Don’t let the other stuff distract you!

3. The Principle of ‘Better the devil you know’

Successful visionaries start from where they are, what they have and what their customers have.

4. The Principle of ‘You eat an elephant one bite at a time’

System stakeholders need to digest new systems in small increments.

5. The Principle of ‘Cause and Effect’

If you change in small stages, the causes of effects are clearer and easier to correct.

6. The Principle of ‘The early bird catches the worm’

Your customers will be happier with an early long-term stream of their priority improvements, than years of promises, culminating in late disaster.

7. The Principle of ‘Strike early, while the iron is still hot’

Install small steps quickly with people who are most interested and motivated.

8. The Principle of ‘A bird in the hand is worth two in the bush’

Your next step should give the best result you can get now.

9. The Principle of ‘No plan survives first contact with the enemy’ (*2)

A little practical experience beats a lot of committee meetings.

10. The Principle of ‘Adaptive Architecture’

Since you cannot be sure where or when you are going, your first priority is to equip yourself to go almost anywhere, anytime.

* 2 This saying is attributed to Prussian general staff and the elder Von Moltke: ‘‘They did not expect a plan of operations to survive beyond the first contact with the enemy. They set only the broadest of objectives and emphasized seizing unforeseen opportunities as they arose . . . Strategy was not a lengthy action plan. It was the evolution of a central idea through continually changing circumstances’’ (From Von Clausewitz in his ‘On War’, quoted by General Electric’s CEO, Jack Welch in a speech December 8, 1981, in Slater, 2000: 194).

THE PRINCIPLES GAME

I have a ‘game’ I play, designed to show the power of each individual principle.

I ask for a random number of choice between 1.01 (1) and 10.10 (100th principle), Or maybe easier 2 numbers, a chapter 1-10 and a principle 1-10. Then we look up the principle in question and I ask people to reflect on the following:

1. Is the principle deep (profound) non-trivial?
2. Is it widely taught? WHERE?
3. Is it widely practiced, for example amongst your own staff? Where?
4. Should serious engineers and professionals learn, and be motivated to apply such principles? Why?
5. Do you think the other 99 principles are less powerful? Take another random sample!

Try this yourself!

I have written a paper (originally for NTNU, Trondheim professors – not that it helped ☺)

Undergraduate Basics for Systems Engineering (SE),
using The Principles, Measures, Concepts and Processes of Planguage.

Held at INCOSE 2007 I believe.
I argue for the academic adoption of Planguage/CE ideas academically. I fear that rational argument is not a useful tool in the face of individual, and academic tradition. But I have hope for the next centuries!

Here is a sample:

<u>Principles</u>

Some Principles of Useful Knowledge

UNIVERSALITY: 1. Knowledge is more useful when it applies to more circumstances
ETERNALITY: 2. Knowledge is more worth learning if it can be applied for a long time after learning it
VALUE: 3. Knowledge is more useful if there is a high value from applying it
SHARING: 4. Knowledge is more useful if it can easily be shared with others
PROOF: 5. Knowledge is useful when early feedback can prove its usefulness in practice
SYNCHRONOUS: 6. Knowledge is more useful when it can be used together with a larger body of knowledge
MEASURABIILITY: 7. Knowledge is more useful when the results of its application can be measured
ACCEPTANCE: 8. Knowledge is more useful when it is widely accepted in your culture.
COST: 9. Knowledge is more useful when the cost of applying it is low.

GENERATION: 10. Knowledge is more useful when is can be used to generate even more useful knowledge.

And

The Notion of Usefulness of Principles:

A principle is a short statement that guides people to take certain decisions or action. It is condensed wisdom. Principles are useful if they remind or teach us to act in a better way than we otherwise would do.

For example:

"There is lots of uncertainty and risk of deviation from plans in any project.

You cannot eliminate risk. But, you can document it, plan and design for it, accept it, measure it and reduce it to acceptable levels. You may want to avoid risk, but it doesn't want to avoid you." Source: CE, page 23.

This principle tries to warn about the inevitability of risk. It also is specific about what you can do about risk. It teaches that you cannot eliminate risk, but you can try to manage it in various ways.

From the departure point of this principle, the teacher can then be more specific on how to identify, specify and mitigate risks.

<u>The Notion Of Half Life of Principles</u>

If a principle became obsolete in a few years – perhaps because of new technology or new economics, then it would be less valuable to learn, and might even be dangerous to continue to practice beyond its true lifetime. So I prefer principles that we can imagine 'always were true', and we can so no clear reason why they 'will not be true for the foreseeable future".

It takes decades from when a principle is stated, until it becomes taught in any substantial way. The student has decades of their future in which to apply a principle. So it makes good sense that the principle is something we can rely on in the long term.

The Notion of Fundamentality of Principles

Principles should be *fundamental*. They should be basic tools for everyday use in planning, engineering, discussing, decision-making, and reasoning. We should be able to use them as the basis for all our more-detailed actions and thinking processes. For example:

"8.The Principle of 'Quality In, From the Beginning'

Quality needs to be designed into processes and products,

Cleaning up bad work is a loser, but cleaning early is better than late.

A stitch in time still saves nine,

But an ounce of prevention is still worth a pound of cure.

"

Source CE, page 24.

The above principle applies to all engineering and management planning work. WE humans seems to have a strong natural tendency to clean up our faulty work when it is discovered, rather than to consciously discover how we can prevent the faults from getting into our work in the first place.

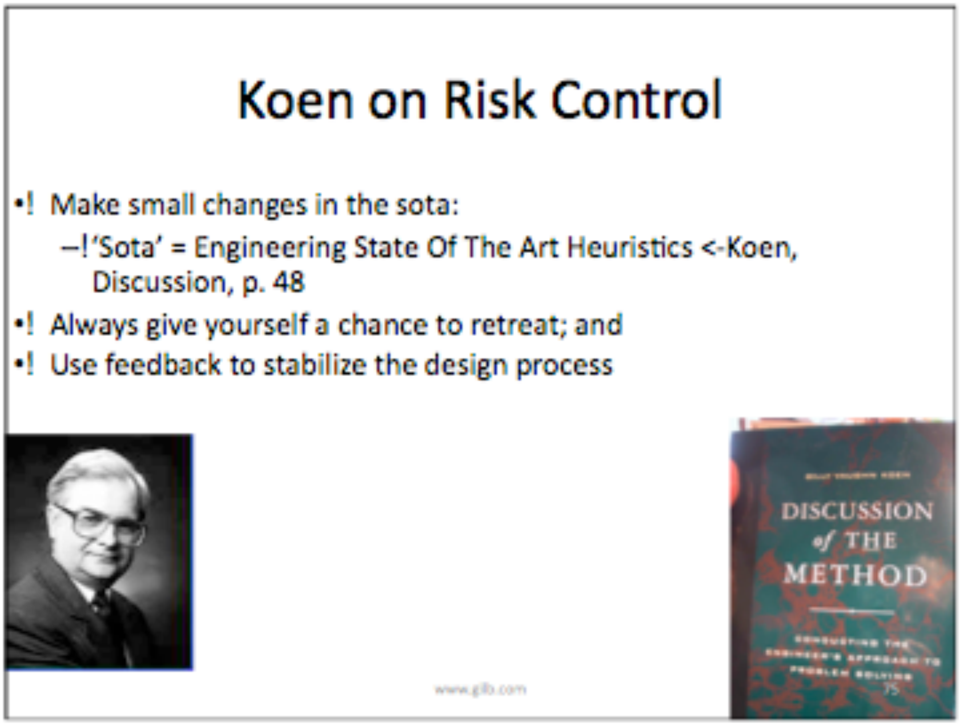
This principle is at the heart of CCI Level 5 (Defect Prevention).

This principle is fundamental. It is at the basis of all improvement efforts in a systems engineering process. It is the basis for a paradigm shift for many professionals I deal with; the shift from ‘fix problems’, to ‘prevent problems’. Students should be taught such profound principles before they waste years discovering them, if at all.

In Competitive Engineering I have offered 100 such principles. I have 'brainstormed' many more in other books and papers, including this one. I am sure my many systems engineering, and other discipline's colleagues have and will continue to develop principles that deserve to be taught formally. My concern is that we place far too little emphasis on selecting and teaching these principles. My concern is that students do not even get a dozen good principles to base their professional work on. I think we need a course called something like "The Most Important Systems Engineering Principles".

http://www.gilb.com/tiki-download_file.php?fileId=98

The Principles of Tao Teh Ching (500 BC)
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.
See http://zapatopi.net/Kelvin/quotes.html (corrected to http://zapatopi.net/kelvin/ Oct 2009), which reads: ''In physical science the first essential step in the direction of learning any subject is to find principles of numerical reckoning and practicable methods for measuring some quality connected with it. I often say that when you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meagre and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the state of Science, whatever the matter may be.''
Lord Kelvin
Evo is, above all, the application of the Shewhart process control cycle, 'Plan-Do-Study-Act'. It is learning from doing and acting on that learning. It is adapting to the complex and changing realities of a project. Evo is systematic engineering work (of the type described by Koen (Koen 1984).
Koen, Billy V. 1984. Toward a Definition of the Engineering Method. Proceedings of the ASEE-IEEE Frontiers in Education. 14th Annual Conference, Philadelphia, PA. 3-5. October 1984. Pages 544–549. The paper also appeared in Engineering Education. December 1984. Pages 150–155. Also in Spring 1985 in The Bent of Tau Beta Pi. Pages 28–33. Reprinted there with permission from Proceedings of the ASEE-IEEE Frontiers in Education. A full page extract is in Gilb (1988). An extended and very interesting comment on the paper's ideas is in Koen (2003).



Koen, Billy Vaughn. January 2003. Discussion of the Method: Conducting the Engineer's Approach to Problem Solving. Oxford University Press. ISBN 0-195-15599-8. Pages 260.
http://www.me.utexas.edu/faculty/people/ koen.shtml/.
http://www.me.utexas.edu/~koen/REVIEWS/Kloman(http://www.me.utexas.edu/~koen/REVIEWS/Kloman(RiskManagementReports).pdf

Summer Reading

The warm and slothful days are upon us, and what better way to respond than to take several hours to expand our minds with new ideas and challenging perceptions. At the top of this year's list of summer reading is an engineer's approach to tackling the problems of uncertainty. It isn't conventional beach reading but it is worth the effort. It came to me through Mike Murphy, who heads the eclectic Informal Risk Management Association in Toronto, a global group of kindred risk management spirits. The book is
Billy Vaughn Koen's *Discussion of the Method* (Oxford University Press, Oxford 2003),
a profound agglomeration of philosophy, mathematics, Eastern mysticism and very practical engineering. As he explains, "engineering is a risk-

taking activity." Mike wrote his own review for Amazon.com and here is his conclusion:
"This book came recommended to me by a colleague who is a professional engineer. It wasn't the cheapest offering from Amazon, and when I took it out of the shipping box, frankly, I was a little disappointed - seeing a slim, paperbound volume, I thought it might be a thin and expensive read on a penny-per-page basis.
Well, was I ever wrong! As I started turning the pages during a first skim read, it struck me that Koen has brought together a huge amount of experience on engineering with a deep understanding of philosophy (to his credit, both Western and Eastern) plus a range of subjects from classical literature and world religion to the vicissitudes of world languages, and forged them into a brilliant synthesis of remarkable clarity and originality.
His central thesis is 'All is heuristic' (All is rule of thumb). He has surrounded this argument with a phalanx of other heuristics (59 in total) that range from the practical ('at some point in the project, freeze the design') to the metaphysical ('sincerity of belief and the inability to disbelieve are poor justifications for claiming that a belief is true') to the paradoxical ('if a concept produces paradoxes, unexplained complexities or unexpected departures from expected results, better consider it a heuristic').
In writing this book, Koen has both mastered and melded a number of seemingly immiscible disciplines - philosophy, linguistics, theology - with his own professional field of engineering (he is professor of Mechanical Engineering at University of Texas at Austin and a fellow of the American Nuclear Society). It is reminiscent of the way that Thomas Aquinas reconciled Christianity with Philosophy.
This is no mean feat, and Koen's book, unpretentiously entitled 'a discussion', is an intellectual tour of the first order. Of course, his many references mean so much more if you are familiar with them. If you aren't, be sure to try them - your life will be immeasurably enriched. In any event, Koen illuminates a path to greater understanding.
His prose is very engaging and the book is well suited for general audiences. It is a book that begs to be read and re-read.
(I heartily agree Tom Gilb Oct 2 2009)
PS I cannot find a download link to his 1984 paper but I have a digital copy I can send on request. TomsGilb@gmail.com

Descartes Principle

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

www.gilb.com 66

> ‘Interesting’ results are our ‘values.’
>
> *Keeney (1992)*

> The real price of everything, what everything really costs to the man who wants to acquire it, is the toil and trouble of acquiring it.
>
> *Adam Smith (1723–90) Scottish economist,* The Wealth of Nations *(1776)*

from CE p. 83

Overview of Planguage Methods for Controlling Costs

The prerequisites for effective control over a project are tight integration of cost and performance considerations, 'design to cost' and using feedback on actual costs to modify plans. Planguage methods ensure these prerequisites by demanding:

- detailed, numeric, measurable performance specifications that adequately capture the performance requirements: the qualities (stakeholder-related objectives) as well as the workload capacities and resource savings (the resource-related objectives)
- resource requirement specifications for the resources allocated, and for any known restrictions on resource expenditure
- design specifications with detailed expected cost *and* performance attributes of the design
- impact estimates of the abilities of the various designs to meet both the performance goals and the resource budgets
- selection of evolutionary steps according to their stakeholder value, and their performance to cost ratios
- feedback from live systems of the actual progress towards achieving the performance levels, and the actual resource expenditure after implementing each evolutionary step
- action being taken on the feedback to adjust specifications, or the future evolutionary steps, to ensure realistic plans (revision of budgets or tradeoffs).

from CE p. 83, some 'principles of controlling costs'

We can control costs if we get early warnings of unexpected costs and we are able to react to these warnings. We must have early, frequent, feedback mechanisms in our planning, our systems engineering and our project management. We can get this degree of control:

- by budgeting resources in *small* (say, 2%) increments
- by *designing* to stay within the budget
- by *reacting to experience* with cost expenditure (changing designs or requirements as far as it is realistic to do so)
- by monitoring a *multiplicity* of resource budgets and a *multiplicity* of performance goals
- by specifying all the *constraints* that apply to the problem, in advance of solving it.

CE p.

You only get what you pay for.

Folk wisdom

> If you think you know something about a subject, try to put a number on it. If you can, then maybe you know something about the subject. If you cannot then perhaps you should admit to yourself that your knowledge is of a meagre and unsatisfactory kind.
>
> *Lord Kelvin, 1893*

Love's Many Attributes

- Trust
  - Truthfulness
  - Broken Appointments
  - Late Appointments
  - Gossiping to Others
- Respect
- Friendship
- Sharing
- Attention
- Understanding
- Support
- Care
- Comfort
- Kissed-ness
- Sex
- Passion
- *other attributes?*

Love.Trust.Truthfulness:
Ambition: No Lies.
Scale: Average Black Lies/Month.
Meter: Confidential Log of Lies.
Past Lies: Past [Ex-Spouse, Two Years Ago]: 42.
Goal [Current Spouse, This Year]: (Past Lies)/2.

Black Lies: Defined As: Non-White Lies.

Figure 5.2
Love is a many-splendored thing! Another example of decomposing a complex subject into its component attributes. This is from a classroom exercise, which was done in two stages. First, we decomposed the complex concept, 'Love' into many aspects. Then we took one attribute at random to see if a reasonable quantified specification could be achieved.

http://www.gilb.com/tiki-download_file.php?fileId=335
L Day's paper on Love Quantification, inspired by the principle of Decomposition for quantification of quality variables.

On Quantification

- No matter how complex the situation, good systems engineering involves putting value measurements on the important parameters of desired goals and performance of pertinent data, and of the specifications of the people and equipment and other components of the system.
- It is not easy to do this and so, very often, we are inclined to assume that it is not possible to do it to advantage.
- But skilled systems engineers can change evaluations and comparisons of alternative approaches from purely speculative to highly meaningful.
- If some critical aspect is not known, the systems experts seek to make it known. They go dig up the facts.
- If doing so is very tough, such as setting down the public's degree of acceptance among various candidate solutions, then perhaps the public can be polled.
- If that is not practical for the specific issue, then at least an attempt can be made to judge the impact of being wrong in assuming the public preference.
- Everything that is clear is used with clarity: what is not clear is used with clarity as to the estimates and assumptions made, with the possible negative consequences of the assumptions weighed and integrated.
- We do not have to work in the dark, now that we have professional systems analysis.

Simon Ramo

Wisdom from founder of TRW (R=Ramo) corporation, on quantification.

===================================Wisom from Lewis CarollWW

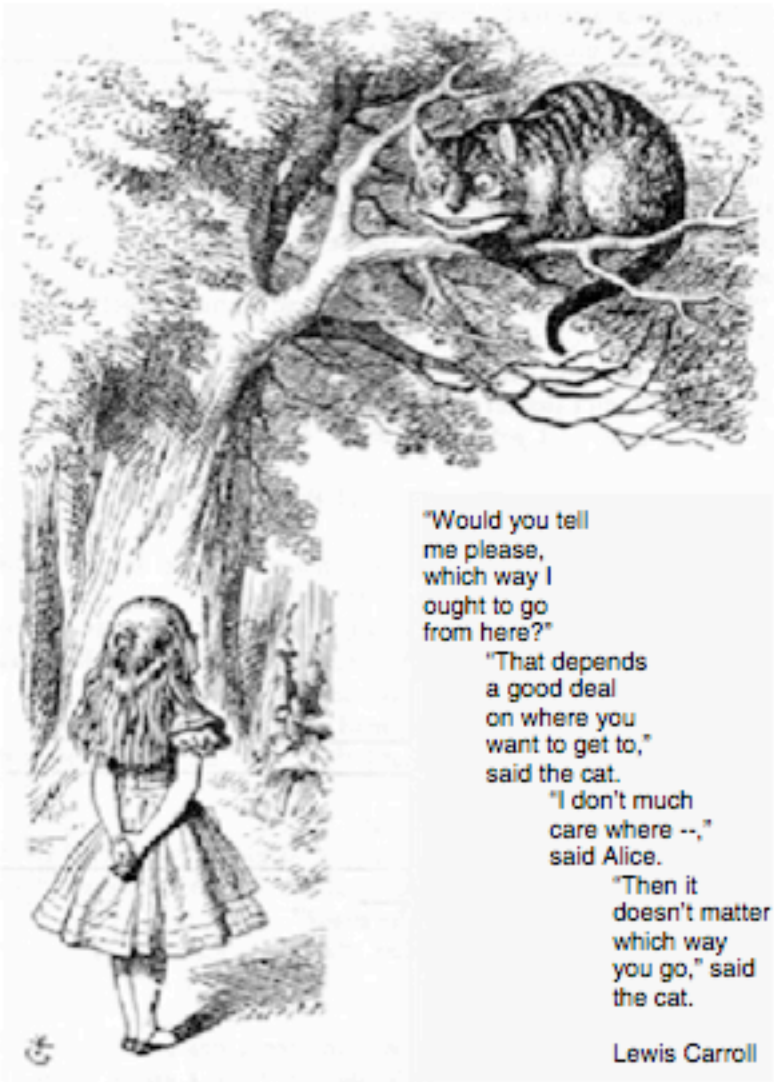


Figure 2.10
Alice and the Cheshire Cat. Illustration by John Tenniel, wood-engraving by Thomas Dalziel. From Chapter 6, *Alice in Wonderland* by Lewis Carroll.

Wisdom from Lewis Carroll, You need to know your requirements clearly, or you cannot decide on the means to achieve them! A main point of Planguage!

=====================================

Vision

At the highest level, there should be a vision statement for a system. A vision or vision statement is a specific, long-range, overall category of requirement. That means it can concern itself with future mission and/or targets and/or constraints. It is a leadership statement for focus and motivation. Visions are often defined in broad summary terms. For example, 'become world class.' But there is no reason to be so vague. Great practical visions[1] are extremely concrete:

"*I believe that this nation should commit itself to achieving the goal, before this decade is out, of landing a man on the moon and returning him safely to the earth.*"

John F. Kennedy.
Delivered before a joint session of Congress, May 25, 1961.[2]

"*I believe that we must improve the numeric level of all critical product and service qualities by an order of magnitude by the end of the decade in order to remain competitive.*"[3]

John Young,
CEO Hewlett Packard Company, April 1986.
Known at the '10X' policy.

"*We shall go on to the end, we shall fight in France, we shall fight on the seas and oceans, we shall fight with growing confidence and growing strength in the air, we shall defend our island, whatever the cost may be, we shall fight on the beaches, we shall fight on the landing grounds, we shall fight in the fields and in the streets, we shall fight in the hills; we shall never surrender.*"

Churchill, June 4, 1940.[4]

A vision will ultimately need to be decomposed into specific requirements such as measurable objectives with quantified goals. Using qualifiers, these requirements can, as necessary, be tied by specification to specific times, locations, components and events of the system.

Wisdom from great people. State objectives in terms of the end state, and don't tell the troops HOW to do it.

Figure G2
Alice meets Humpty Dumpty[3].

[3] Illustration by John Tenniel to Chapter 6 of 'Through the Looking-Glass' by Lewis Carroll. Wood-engraving by Thomas Dalziel. Illustration from http://www.scholars.nus.edu.sg/landow/Victorian/graphics/tenniel/lookingglass/6.1html/. Additional Lewis Carroll text was added in.

More wisdom from Lewis Carroll: A primary principle of Planguage is that WE determine the meaning of the words. We do not leave them to accidental and random interpretations by individuals.

"The aim must include plans for the future."

<-(*Deming 1993 Page 51*)

"It is important that an aim never be defined in terms of activity or methods. It must always relate to how life is better for everyone."

<-(*Deming 1993 Page 52*).

"The aim precedes the organizational system and those that work in it. Workers, for example, can not be the source of the aim, for how would one know what kind of workers to choose?"

<-(*Deming 1993 Page 52*)
Attributed to Deming by Carolyn Bailey.

<- CE p. 324
Principles for setting requirements. Wisdom from my friend Deming.

Similar to the Vision wisdom : keep it at a high result level – not the 'means'
(who I managed to take to Ballet in London on 2 occasions! ☺)

"Assumptions are suppositions, conjectures, and beliefs which lack verification at the time of writing, or requirements and expectations that are not within our power to control, but which have been used as part of the basis for planning future actions. We identify for each the degree of risk involved and possible consequences if the assumption is erroneous."

<- *Don Mills, NZ 2002 (Personal e-mail)*

Notes:

"To leave [soft considerations] out of the analysis simply because they are not readily quantifiable or to avoid introducing 'personal judgments,' clearly biases decisions against investments that are likely to have a significant impact on considerations as the quality of one's product, delivery speed and reliability, and the rapidity with which new products can be introduced."

<- *R. H. Hayes et al.* Dynamic Manufacturing, *Free Press 1988 NY Page 77,* quoted in Mintzberg (1994 Page 124)

"Aligning Rewards with Measurements 'You have to get this one right. . . . a universal problem: What you measure is what you get – what you reward is what you get. Static measurements get stale. Market conditions change, new businesses develop, new competitors show up. I always pounded home the question 'Are we measuring and rewarding the specific behavior we want?'"

<- *Jack Welch, former CEO* General Electric *(Welch 2001 Page 387)*

Principles of top level objectives, Quantify the 'soft' ones. Define the measurable goals you really want to achieve.

"The other beauty ('truths I've learned to challenge') goes something like this: A team comes in with a proposal to leapfrog the current position of its leading competitor. The implicit assumption is the competition will be sleeping. Doesn't usually happen that way . . . It was tough, but we tried like hell to look at every new product plan in the context of what the smartest competitor could do to trump us. Never underestimate the other guy."

Jack Welch, former CEO General Electric (Welch 2001 Page 391)

Jack Welch on the principle of analyzing the future. Something we suggest with Trend parameter CE p. 433

.

(Company Communication) Bill of Rights <- CE book page 74, sourced from PoSEM 1988, developed by Gilb for ICL CEO Robb Wilmot 1982 (who adopted it officially, plaques on the wall etc.)

. You have a right to know precisely what is expected of you.
. You have a right to clarify things with colleagues, anywhere in the organization.
. You have a right to initiate clearer definitions of objectives and strategies.
. You have a right to get objectives presented in measurable, quantified formats.
. You have a right to change your objectives and strategies, for better performance.
. You have a right to try out new ideas for improving communication.
. You have a right to fail when trying, but must kill your failures quickly.
. You have a right to challenge constructively higher-level objectives and strategies.
. You have a right to be judged objectively on your performance against measurable objectives.
. You have a right to offer constructive help to colleagues to improve communication.

Original version in (Gilb 1988, Principles of Software Engineering Management, Page 23)
Figure 2.5 (CE Page 74) The author suggested these 'rights' for a multinational client. Of course it is a sneaky way to tell people what their 'duties' are!

Gilb's Law of Reliability

http://www.gilb.com/tiki-download_file.php?fileId=338
Gives the original 1975 Datamation article. With my commentary on the 'Laws'. I drafted them one evening for a course I was going to hold the next day; and later wrote the article for Datamation (Laws of Unreliability)

Gilb's Laws of Unreliability : Computers are unreliable, but humans are even more unreliable.

Gilb's Laws of Unreliability Corollary : At the source of every error which is blamed on the computer you will find at least two human errors, including the error of blaming it on the computer.

Gilb's Laws of Unreliability : Any system which depends on human reliability is unreliable.
Gilb's Laws of Unreliability : The only difference between the fool and the criminal who attacks a system is that the fool attacks unpredictably and on a broader front.
Gilb's Laws of Unreliability : A system tends to grow in terms of complexity rather than of simplification, until the resulting unreliability becomes intolerable.
Gilb's Laws of Unreliability : Self-checking systems tend to have a complexity in proportion to the inherent unreliability of the system in which they are used.
Gilb's Laws of Unreliability : The error-detection and correction capabilities of any system will serve as the key to understanding the type of errors which they cannot handle.
Gilb's Laws of Unreliability : Undetectable errors are infinite in variety, in contrast to detectable errors, which by definition are limited.
Gilb's Laws of Unreliability : All real programs contain errors until proved otherwise -- which is impossible.
From this website: http://www.sanjeev.net/murphys-laws/murphys-laws-065.html

---

Demarco and Gilb's Law of Measurability
Gilb Measurability Principle
PeopleWare page 59 (second edition) provides the most helpful and concise version of this principle, which De Marco and Lister call Gilb's Law:
Anything you need to quantify can be measured in some way that is superior to not measuring it at all.
Source: http://c2.com/cgi/wiki?GilbMeasurabilityPrinciple
I was sitting on an outdoor bench in 1986 at Imperial College, London, at a Conference with Tom DeMarco, and we discussed this, which led to his formulation in his Peopleware book. <-TsGilb

---

Practical Strategies for Risk Management

1. **Quantify requirements**: All critical performance and resource requirements must be identified and quantified numerically.
2. **Maximize profit, not minimize risk**: Focus on achieving the maximum benefits within budget and timescales rather than on attempting to eliminate all risk.
3. **Design out unacceptable risk**: Unacceptable risk needs to be 'designed out' of the system consciously at all stages, at all levels in all areas, for example, architecture, purchasing, contracting, development, maintenance and human factors. This means selecting lower-risk options.
4. **Design in redundancy**: When planning and implementing projects, conscious backup redundancy for outmaneuvering risks is a necessary cost.
5. **Monitor reality**: Early, frequent and measurable feedback from reality must be planned into your development and maintenance processes, to identify and assess risks before they become dangerous.
6. **Reduce risk exposure**: The total level of risk exposure at any one time should be consciously reduced to between 2% and 5% of total budget.
7. **Communicate about risk**: There must be no unexpected risks. If people have followed guidelines, and are open about what work they have done, then others will have the opportunity to fight risks constructively. Where there are risks, then share that information.
8. **Reuse what you learn about risk**: Standards, and other forms of work process guidance, must capture and assist good practice. They must be subject to continuous process improvement.
9. **Delegate personal responsibility for risk**: People must be given personal responsibility in their sector for identification and mitigation of risks.
10. **Contract out risk**: Make vendors contractually responsible for risks. They will give you better advice and services as a result.

Figure 1.2
Practical strategies for risk management.

Gilb's Risk Principles (aka strategies) from page 6, Competitive Engineering book, 2005
http://www.gilb.com/tiki-download_file.php?fileId=20
for Risk Paper

http://www.gilb.com/tiki-download_file.php?fileId=155
for Clinical Risks slides London lecture

http://www.gilb.com/tiki-download_file.php?fileId=227

4slidePERPage Gilb Critical Factor Dynamic Risk Control- the use of Planguage tools – requirement quantification, Impact Estimation and Evolutionary Feedback – in discovering risks and mitigating them MASTER.pptx_compressed.pdf (3.45 Mb)
You can download this file using: http://www.gilb.com/tiki-download_file.php?fileId=227
Especially and originally for The Gilb Annual London Seminar June 23-27 2008

Figure 1.10 <- CE page 32-33
Example of a corporate policy standard.

GILB'S INTERPRETATION OF ERICSSONS CORPORATE QUALITY POLICY

Their policy was about 52 pages, and one evening at their site I decided to see if I could condense the essentials. So you could say these are Gilb-formulated principles, inspired by Ericsson (about 1995±3).

Notes Supporting the Example of a Corporate Policy Standard

1. Quantify Critical Success Factors: All critical success factors (function, performance and resource) for any activity (planning, systems engineering and management) shall be expressed clearly, unambiguously, measurably and testably at all stages of consideration: presentation, evaluation, construction and validation.
2. Evaluate Risk: In any planning or systems engineering work we shall explicitly document all notion of suspected or possible elements of risk or uncertainty, so nobody reading it can be in the least doubt as to the state of our certainty and knowledge.
3. Assess Change Impact – To Exercise Control over Multiple Dimensions of Performance and Budget: All design ideas (strategies, system components, processes or other devices) shall be evaluated with regard to their effects on all the critical objectives and budgets. Initially, this should be by estimates, which are based on facts and experience. On delivery, the design ideas shall then be evaluated by actual measurements taken as early and as frequently as possible.
4. Ensure Change Control – Configuration Management and Traceability: All statements of objectives, budgets, design ideas, and estimates and measures of the impact of design ideas on objectives and budgets shall be captured with explicit detailed information as to their sources, so that detailed change control is made effective and efficient.

5. Perform Evolutionary Project Management: All projects whether concerning organizational issues or product development, shall be controlled by a Plan-Do-Study-Act process control cycle. They shall have small increments of cost and time (in the 2% to 5% range normally) before attempting to deliver useful customer increments of function and/or performance improvement (at least some sort of field trial). Where there is any choice of incremental step content we shall choose the increment which gives the greatest quantified impacts in total on all critical customer or project objectives, with least resource expenditure.

6. Ensure Continuous Work Process Improvement: Practical priority will be given to measurable continuous improvement of all work processes in systems engineering, management and other company activities. Plans for type and degree of improvement will be budgeted; and progress towards improvement objectives will be measured. The ambition level will be world-class levels and to be the leader in any area. As a practical matter all employees are expected to participate in analysis of current defects found by quality control (for example, specification quality control (SQC) and test) and to spend effort improving the current work environment to eliminate 50% of the current defects every year over the next few years.

7. Evaluate Specification Quality: All documents, capable of producing a significant impact on our performance levels, must be evaluated using the best available quality control process. These documents must meet an appropriately high quality standard (that is a low numeric value for the 'maximum possible remaining major defects/page' as specified in our written standards and policies) before being released to any internal or external customer for serious use. The ultimate release level shall be state of the art (between 0.3 and 3.0 remaining major defects/page).

How to decompose systems into small evolutionary steps: (a list of practical tips)

1. Believe there is a way to do it, you just have not found it yet![4]
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 stakeholders: users, salesperson, installer, testers or customer. However small the positive contribution, something is better than nothing.
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 stakeholder. Think 'tomorrow' or 'next week.' Think of one interesting improvement.
6. Focus on the results. (You should have them defined in your targets. Focus on moving *towards* the goal and budget 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, not style.
9. Don't be afraid that the stakeholders won't like it. If you are focusing on the results they want, then by definition, they should like it. If you are not, then do!
10. Don't get so worried about "what might happen afterwards" that you can make no practical progress.
11. You cannot foresee everything. Don't even think about it!
12. If you focus on helping your stakeholder 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/stakeholder community.
15. When some cycles, like a purchase-order cycle, take a long time, initiate them early (in the 'Backroom'), 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', stakeholders, or end user. They probably have dozens of suggestions.
19. Talk with end users and other stakeholders 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 people overlook it.

[4] Working within many varied technical cultures since 1960 I have never found an exception to this – there is always a way!

Figure 10.6
Ideas to assist identifying steps.

http://www.gilb.com/tiki-download_file.php?fileId=350
For decomposition slides

And
http://www.gilb.com/tiki-download_file.php?fileId=41
my 2008 INCOSE Paper on Decomposition

==

8 Competitive Engineering

Twelve Tough Questions

1. **Numbers**
   Why isn't the improvement quantified?
2. **Risk**
   What is the degree of risk or uncertainty and why?
3. **Doubt**
   Are you sure? If not, why not?
4. **Source**
   Where did you get *that* information? How can *I* check it out?
5. **Impact**
   How does your idea affect *my* goals and budgets, measurably?
6. **All critical factors**
   Did we forget anything critical to survival?
7. **Evidence**
   How do you *know* it works that way? Did it 'ever'?
8. **Enough**
   Have we got a complete solution? Are *all* requirements satisfied?
9. **Profitability first**
   Are we planning to do the 'profitable things' first?
10. **Commitment**
    Who is responsible for failure, or success?
11. **Proof**
    How can we be sure the plan is working, during the project, early?
12. **No cure, no pay**
    Is it 'no cure, no pay' in a contract? Why not?

© Tom Gilb 2000–5

A full paper on this is available at www.Gilb.com

http://www.gilb.com/tiki-download_file.php?fileId=24

12 Tough questions are Gilb principles for analysis of suggested improvement ideas.. The link above is a more-detailed paper on the subject. The paper has never been published – but I always felt it should be, in a management journal – HBR anyone ? ☺

==

The dominating influences behind the creation of Planguage include:

The works of Deming, Juran, Crosby, Jevons (The Principles of Science, Dover Edition, 1960, originally published 1875), Boehm, Weinberg, Lord Kelvin, Keeney, Koen and Peters. See also the Bibliography and the citations in this book.

Underlying Principles of Evolutionary Project Management

The underlying principle of Evo is the Plan-Do-Study-Act cycle (PDSA cycle). In other words, the 'process control cycle' as taught by Walter Shewhart of AT&T from the 1920s onwards and, by his pupils, W. Edwards Deming from the late 1940s to the 1990s (Deming 1986) and Joseph Juran (1974). It is one of nature's great laws; learn, adapt and survive.

Evo expands on the Statistical Process Control 'Plan-Do-Study-Act' (PDSA) cycle concepts since it demands:

1 In 1988, the author taught Evo to an HP project team, which included Todd Cotton, who later went on the spread the method widely at HP (Cotton 1996), (May and Zimmer 1996).

The evolutionary development methodology has become a signifi- cant asset for Hewlett-Packard software developers. Its most salient, consistent benefits have been the ability to get early, accurate, well-formed feedback from users and the ability to respond to that feedback. Elaine May and Barbara Zimmer, Hewlett-Packard (May and Zimmer 1996, Page 44).//INTEGRA/ELS/PAGINATION/ELSEVIER UK/OMP/3B2/FINALS/0750665076-CH010.3D – 291 – [291–320/30] 29.6.2005 12:44PM

. Early delivery of project results to stakeholders (for example, 'next week'!)

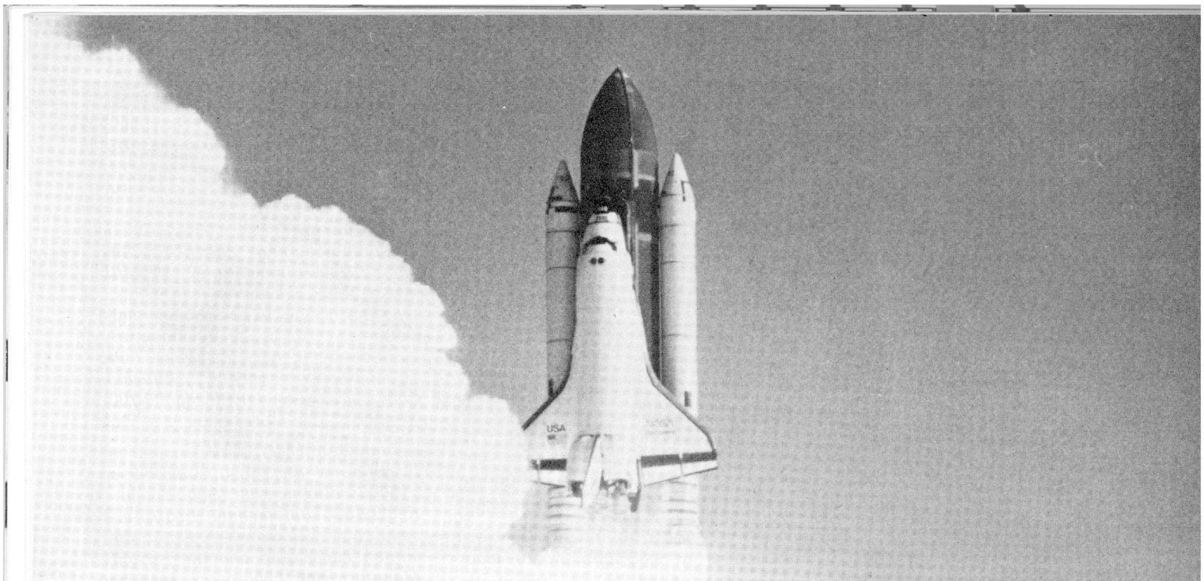
. Frequent releases to stakeholders (for example, 'every Friday') . Small increments ('steps') (for example, no more than 2% of total project) . Useful-to-stakeholder steps (benefit delivered, value experienced) . Selection and sequencing of steps according to degree of stakeholder

benefit; usually but not always, high-profit steps first (using dynamic priority determination).

Who could be against such an idea? It is a powerful competitive weapon. In practice, the main problem for project management is usually 'how?' How is a major project divided up into a succession of say, monthly improvements to be delivered into the hands of the users? Some people don't see any difficulty. Many, however, are unable to envision such small step decomposition for their projects, and usually claim it is impossible. In my experience, there are always ways of achieving such decomposition. It is a question of training, being determined to find the answer and having the right technical knowledge and/or sufficient insights into the stakeholder environment.

As our entire political, technological and economic world now has a greater rate of change and is much more unpredictable and complex than ever before, adaptive methods, such as Evo, must

From page xi – endorsements CE book
“Systems engineers should find Competitive Engineering widely useful, with or without the additional framework provided by Planguage. Even without adopting Planguage as a whole there are numerous important principles and techniques that can benefit any system project. And those who dip in looking for solutions to one problem or another may come to appreciate the full framework of Planguage.”
Dr Mark W. Maier
Distinguished Engineer at The Aerospace Corporation and Chair of the INCOSE Systems Architecture Working Group. Co-author of The Art of Systems Architecting, Second Edition (CRC Press).



Toward a Definition of the Engineering Method

Billy Vaughn Koen
University of Texas at Austin

Can you name one thing in the room in which you are sitting (excluding yourself, of course) that was not developed, produced, or delivered by an engineer?

(Typically rhis question is greeted with a bewildered silence.)

Can **you** name a profession that is affecting **your** life more incisively than is the engineer's?

(The effectof the atomic bomb, telephone, computer, airplane—all undeniably the products of the engineer's method—on our health, stress level and happiness makes this a difficult question to answer)

Since engineering is evidently very important, can **you** now define the engineering method for solving problems?

(Were you asked this about the scientific method, you could easily answer)

Lacking a ready answer, can you then name a nationally known engineer who is wise, well-read and recognized as a scholar in the field of engineering—one to whom I can turn to find out what engineering really is?

(We can all name economist?, lawyers, doctors, and religious leaders with similar qualifications, bur to which engineer can Peter Jennings turn to put engineering in perspective in thecase of an emergency in a nuclear reactor, the dumping of chemical wastes at Love Canal or the collapse of rhe walkway in rhe Hyatt Regency' The only answer to no one)

No profession affecting our world to rhe extent that engineeringdoes can claim thisisolation.