



**Tom  
Gilb**

## Competitive Engineering: A totally metrics-based system-development method.

3 hour tutorial on the 13th afternoon

**Presented at**

The International Workshop on Software Auditing and Metrics (SAM 2004)

**at The International Conference on Enterprise Information Systems (ICEIS) 2004**

**ICEIS 2004 is organized and hosted by Universidade Portucalense - Porto (Portugal)**

<http://www.iceis.org/>

# Detailed Tutorial Outline:

Planguage: a quantified planning language.

Integrating benchmarks and requirement targets

Quantified Quality Control of specifications

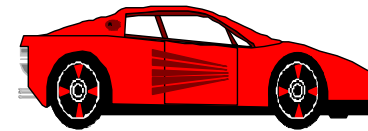
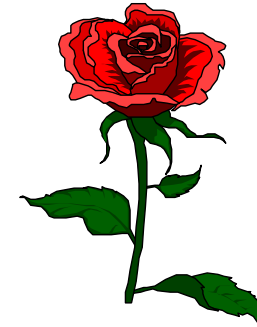
Impact Estimation Tables for quantified evaluation of design

Evolutionary Project Management

*Consider the Performance of :*

A flower

- fragrance
- attractiveness
- pollen quantity
- toxicity
- bloom frequency



A car

- comfort
- safety
- speed
- capacity

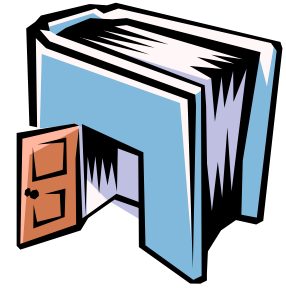
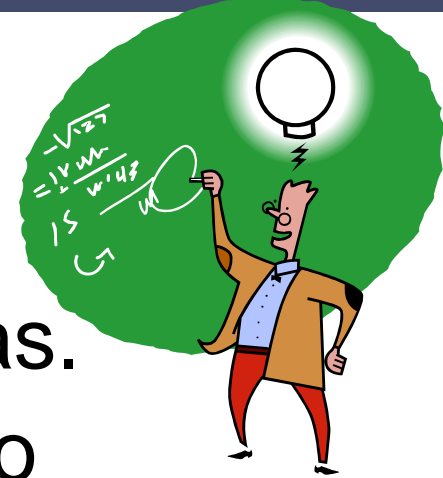
A person

- balance
- intelligence
- courtesy
- helpfulness

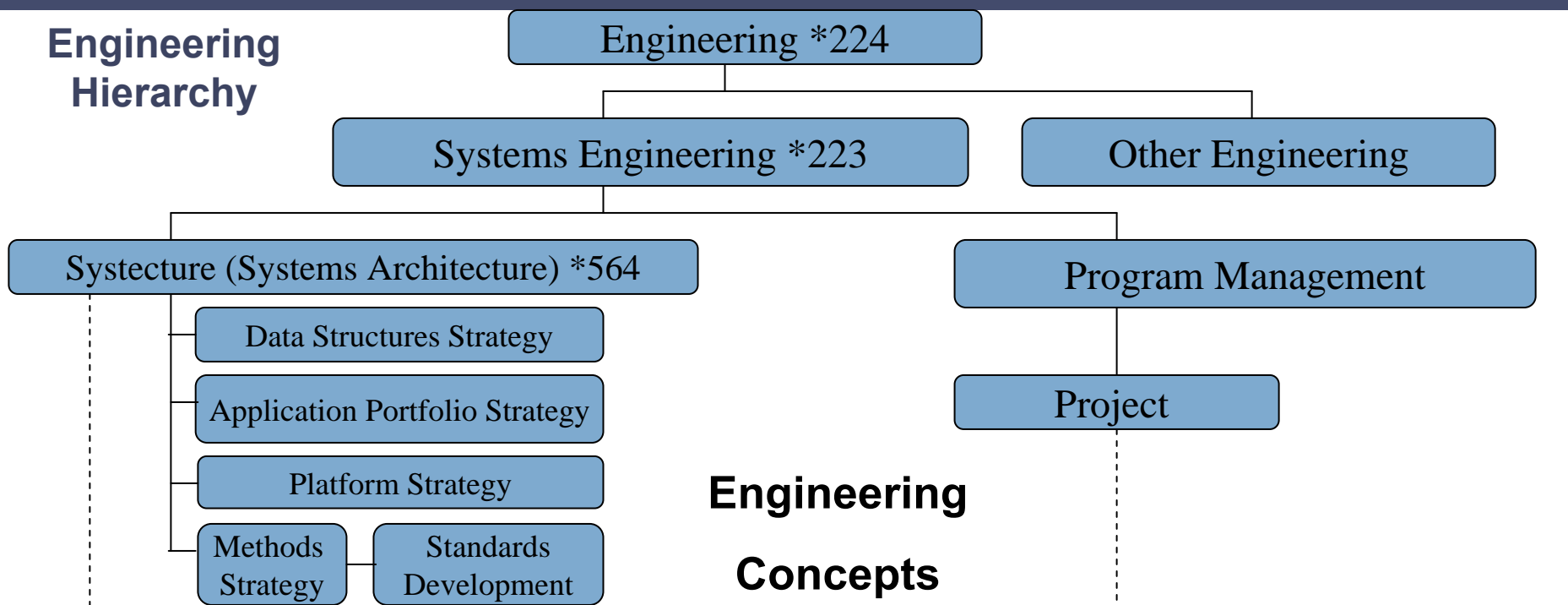


# Tutorial Objectives

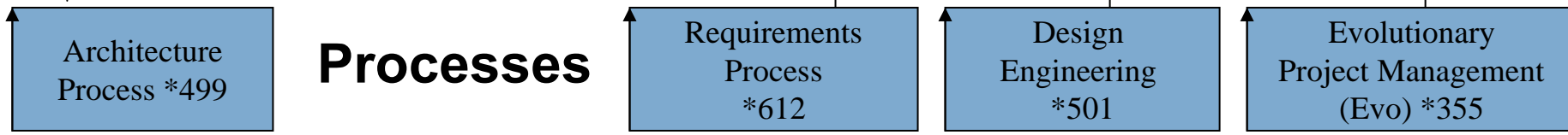
1. Become aware of entirely new ideas.
2. Be able to evaluate if these apply to participant's work.
3. Be aware of how to get more detailed information on the subjects.
4. Enthuse participants with the attractiveness of the ideas presented.



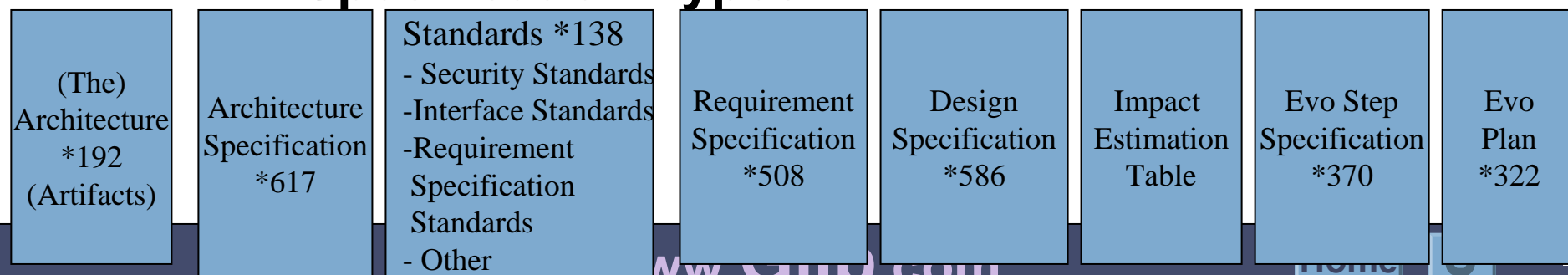
# Engineering Hierarchy



## Engineering Concepts



## Specification Types



# Part 1: Planguage: a quantified planning language.

A Planning Language - an engineering language

A systems engineering language (software, management)

Concept Glossary

Graphical Language

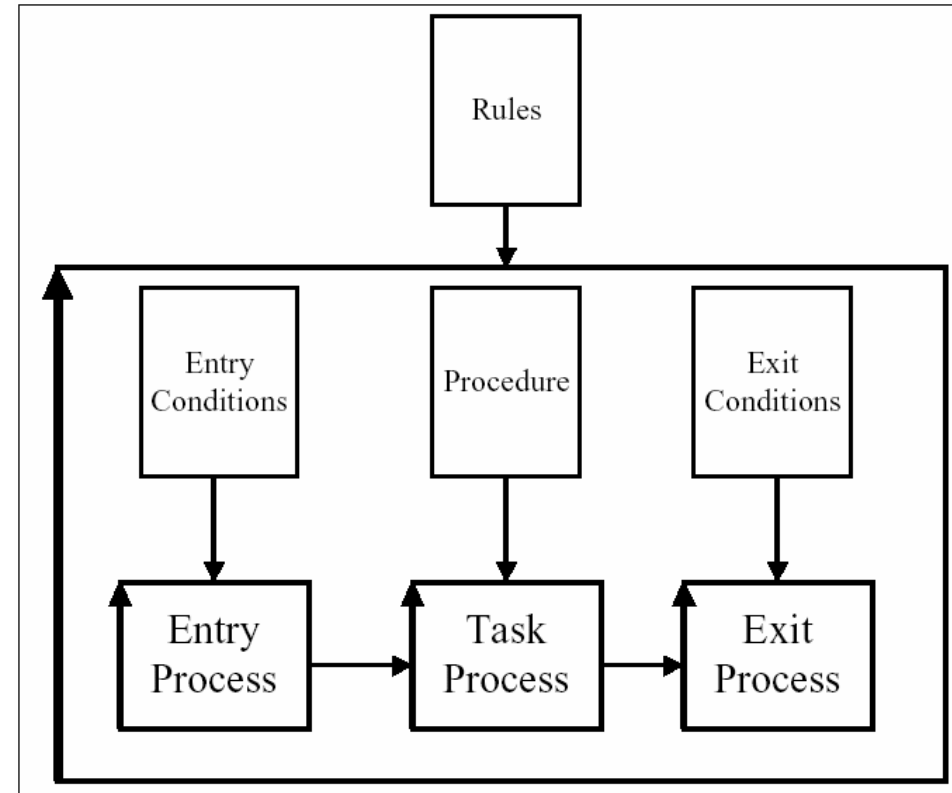
Control of Multiple dimensions:  
Performance, Costs,  
Constraints

Extendible, Tailorable, Open

Rich views, traceability,  
configuration management

Risk Management

Priority Management



# A Planning Language - an engineering language

Uses =

Systems Analysis

Requirements

Contracting specs

Design -

Architecture

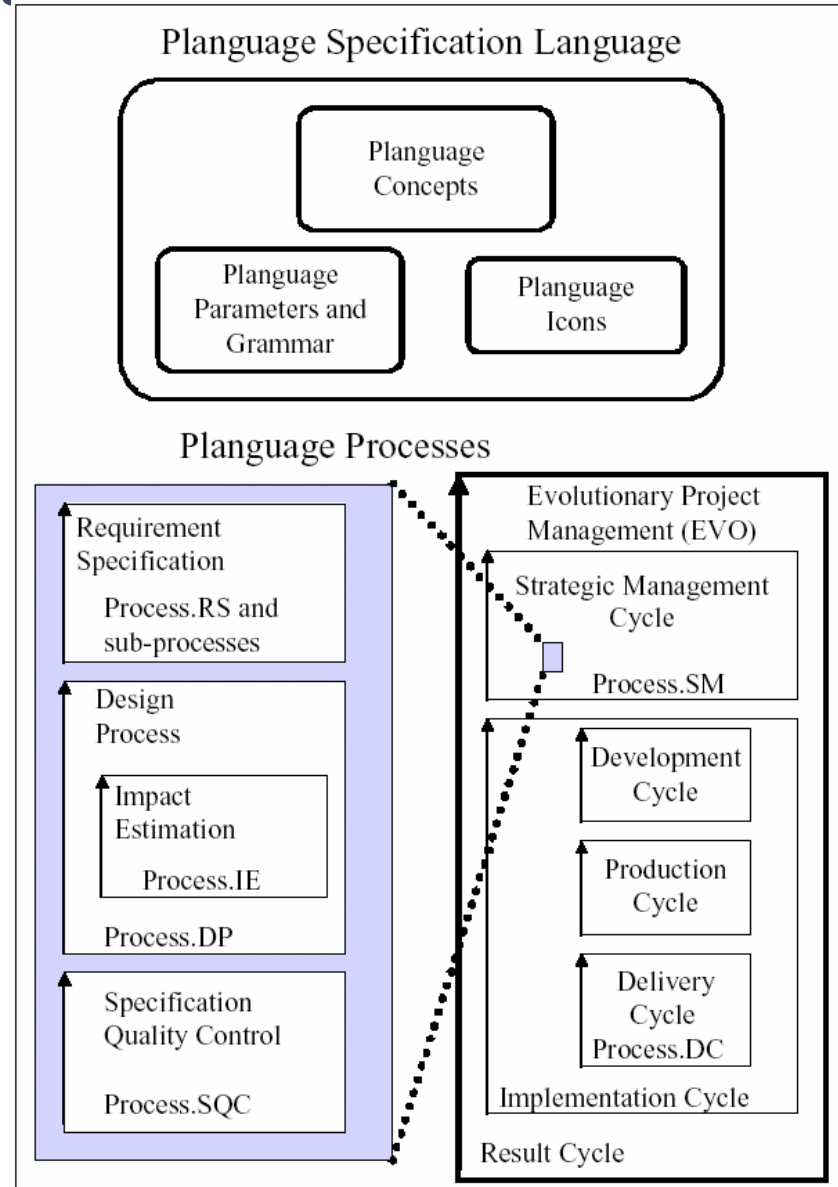
Presentation

Spec Quality

Control

Project

Management



# A systems engineering language

(also software, management)

Generic Ends-Means process

Well-defined standards

Specification rules

Requirements and design processes

One page - modules

Reuse of generic standards

Suitable for

Top management strategy

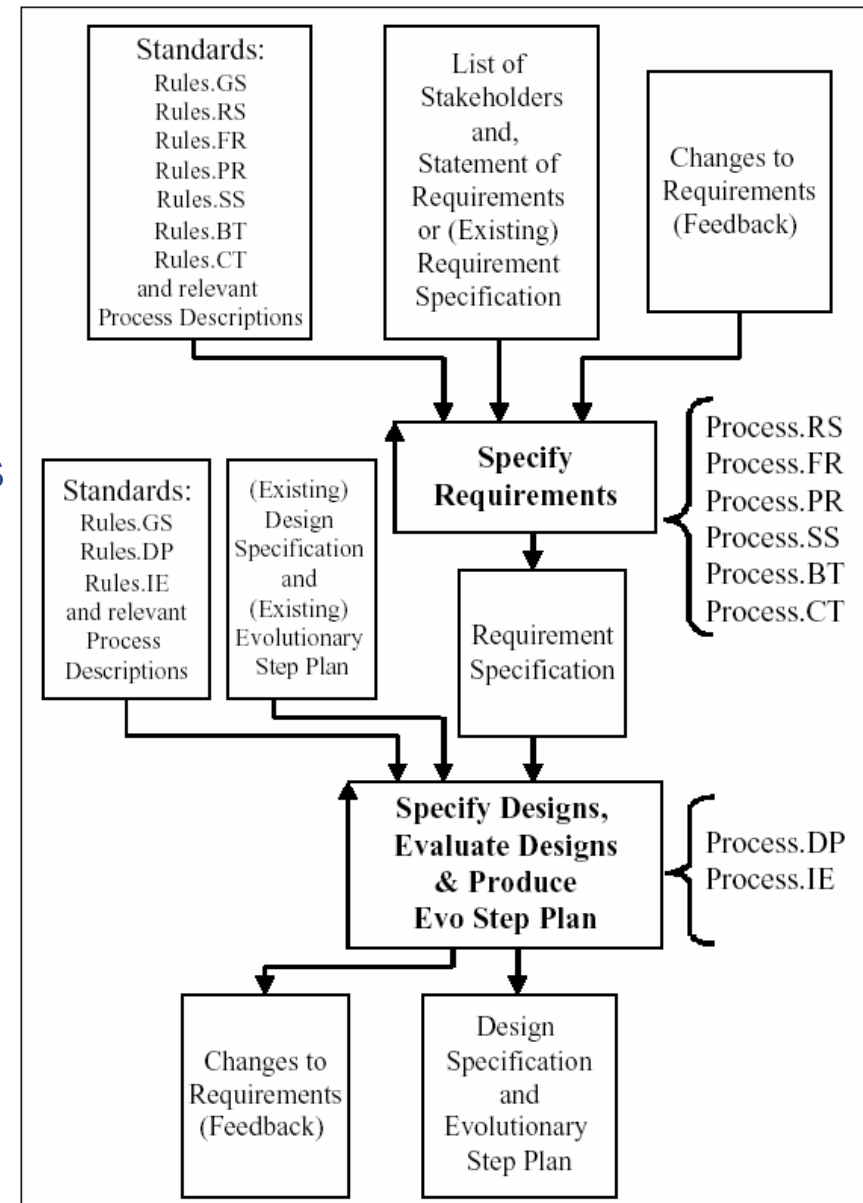
Marketing product plans

Software engineering

Systems engineering

Specific engineering

Aircraft for example



Language standards

# Concept Glossary

## Glossary Purpose.

The central purpose of this Planguage glossary is to define 'Concepts' – not words.

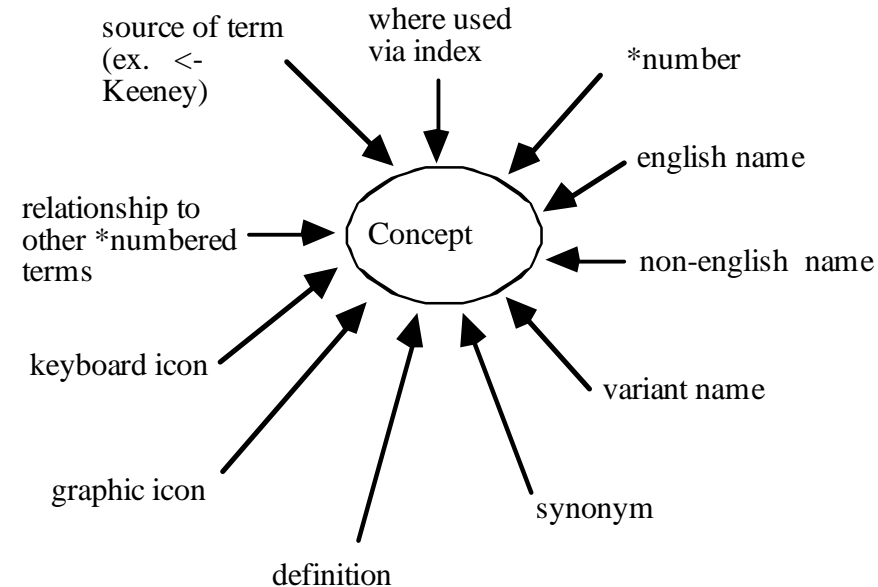
These concepts have many 'names' (or 'tags' in Planguage) and attributes.

The 'names' function as 'pointers' to the concept, but *names do not change or determine the concept itself*.

Names, numbers and icons merely *cross-reference* the concept.

The central, universal identification tag of a concept is its unique number, prefaced by an asterisk (\*001 etc.).

This device is designed to allow and enable full or partial translation to various international languages, and to corporate dialects.





# Graphical Language

For many concepts we have defined graphical symbols

Keyed Icons:

So that symbols can be keyed in combination with text specification

Similar to corresponding drawn icons

Drawn icons:

Suitable for graphical presentation

Why?

International language

Avoids debates over word choice

Short notation

PLANGUAGE TERM	Keyed ICON
<i>Planguage Concept</i>	
Gist:	$\Sigma$
Ambition Level:	@.Σ
Scale:	- - -
Meter:	- ? -
<i>TARGETS</i>	
Goal:	>
Stretch:	>+
Wish:	>?
<i>CONSTRAINTS</i>	
Fail:	>>
Survival Limit:	[ ]
<i>SYSTEM SPACE CONDITIONS</i>	
<i>Time, Place &amp; Event</i>	[qualifier conditions]
<b>Background Information:</b>	
Source:	<-
Comment:	"text."
<i>BENCHMARKS</i>	
Past:	<
Record:	<<
Trend:	?<

# Control of Multiple dimensions: Performance, Costs, Constraints

Planguage specializes  
in

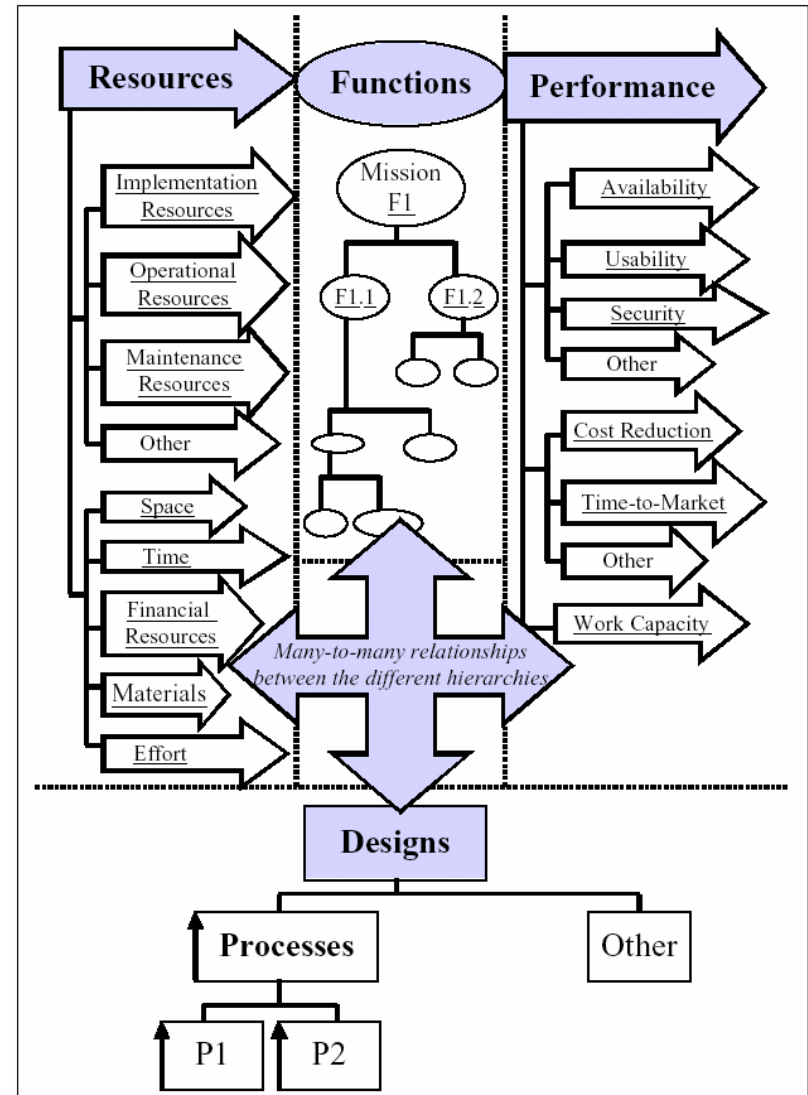
trying to get control  
over

multiple and  
dynamically changing  
critical system  
attributes,

through quantified

requirement  
specification,  
design impact analysis  
and

measurement tactics.



# Extendible, Tailorable, Open

## Planguage:

*Free of cost, & royalties*

*Easy to extend*

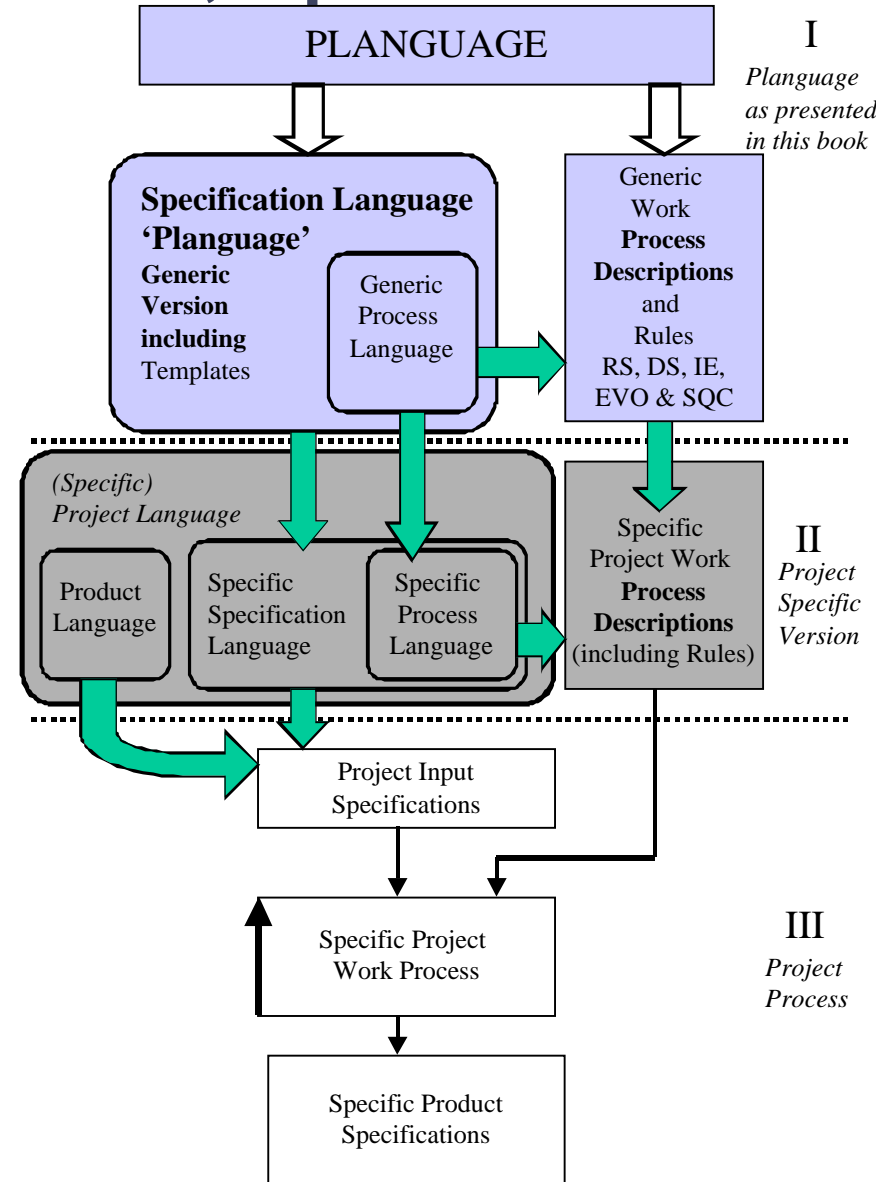
*Easy to modify locally*

Corporate

Project level

National language

*Designed for re-use and tailoring of reused elements*



# Rich views, traceability, configuration management

Some Planguage parameters which define **relationships**.

Authority

Source

Owner

Author

Implementer

Impacts

Supports

Supported By

Version

Derived From

Sub-component of

Sub-components {list}

Dependencies

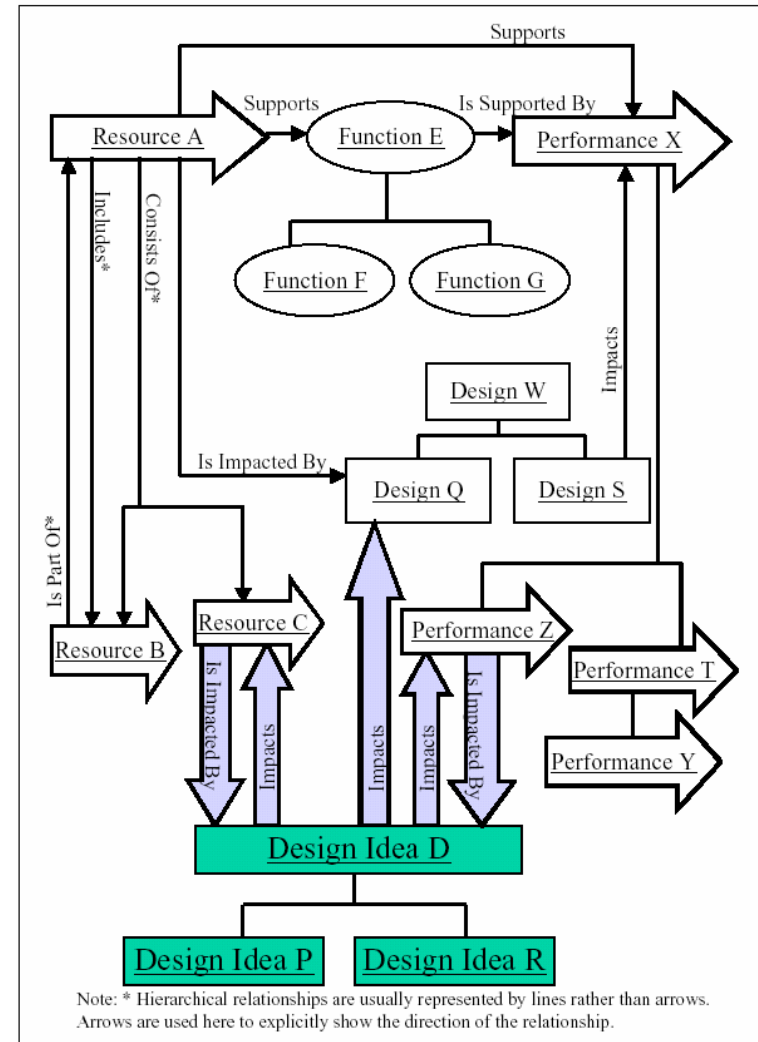
Contract

Test Case

Scenario

Model

And more!



# Risk Management

Planguage integrates  
specific tools for risk  
specification

with more general  
tools for risk  
*recognition* and risk  
*analysis*

in a *single integrated*  
specification  
language.

## TEMPLATE FOR FUNCTION SPECIFICATION <with hints>

**Tag:** <Tag name for the function>.  
**Type:** <{ Function Specification,  
Function (Target) Requirement,  
Function Constraint}>:

Note: By default, a 'Function Requirement' is assumed to be a 'Function Target'.

### Basic Information

**Version:** <Date or other version number>.  
**Status:** <{Draft, SQC Exited, Approved}>.  
**Quality Level:** <Maximum remaining major defects/page, sample size, date>.

**Stakeholders:** <Name any stakeholders with an interest in this specification>.  
**Owner:** <Name the role/email/person responsible for changes and updates to this specification>.

**Gist:** <Give a 5 to 20 word summary of the nature of this function>.  
**Description:** <Give a detailed, unambiguous description of the function, or a tag reference to someplace where it is detailed. Remember to include definitions of any local terms>.

### Relationships

**Supra-functions:** <List tag of function/mission, which this function is a part of. A hierarchy of tags, such as A.B.C, is even more illuminating. Note: an alternative way of expressing supra-function is to use Is Part Of>.  
**Sub-functions:** <List the tags of any immediate sub-functions (that is, the next level down), of this function. Note: alternative ways of expressing sub-functions are Includes and Consists Of>.  
**Is Impacted By:** <List the tags of any design ideas or Evo steps delivering, or capable of delivering, this function. The actual function is NOT modified by the design idea, but its presence in the system is, or can be, altered in some way. This is an Impact Estimation table relationship>.  
**Linked To:** <List names or tags of any other system specifications, which this one is related to intimately, in addition to the above specified hierarchical function relations and IE-related links. Note: an alternative way is to express such a relationship is to use Supports or Is Supported By, as appropriate>.

### Measurement

**Test:** <Refer to tags of any test plan or/and test cases, which deal with this function>.

### Priority and Risk Management

**Rationale:** < Justify the existence of this function. Why is this function necessary? >.  
**Assumptions:** <Specify, or refer to tags of any assumptions in connection with this function, which could cause problems if they were not true, or later became invalid>.  
**Dependencies:** <Using text or tags, name anything, which is dependent on this function in any significant way, or which this function itself, is dependent on in any significant way>.  
**Risks:** <List or refer to tags of anything, which could cause malfunction, delay, or negative impacts on plans, requirements and expected results>.  
**Priority:** <Name, using tags, any system elements, which this function can clearly be done *after* or must clearly be done *before*. Give any relevant reasons>.

### Specific Budgets

**Financial Budget:** <Refer to the allocated money for planning and implementation (which includes test) of this function>.

# Priority Management

Priority is

Claim on scarce or limited resources

Is a function of

Constraint type (Survival, ..)

Target type (Goal, ..)

Remaining gap to constraint or target level & [qualifiers]

Remaining budgeted resources; and their constraint and target levels

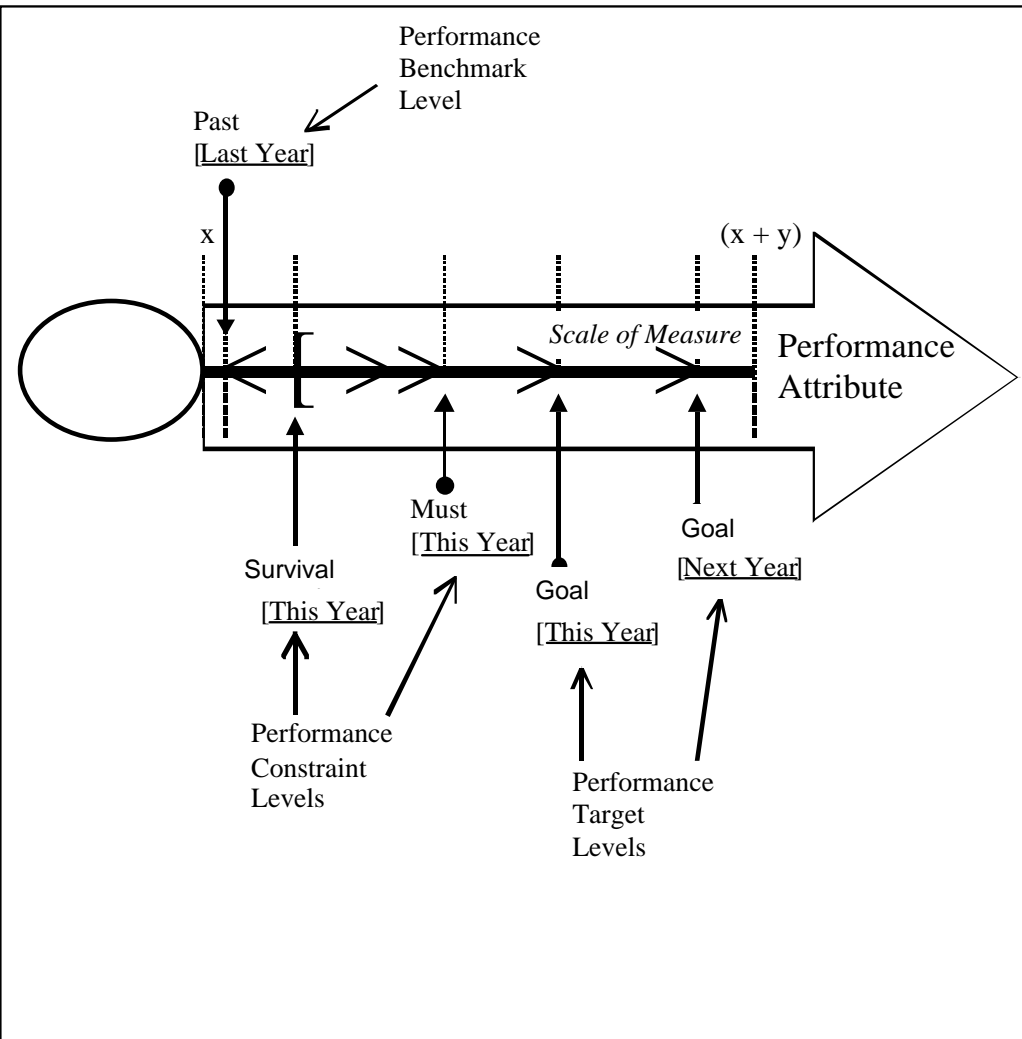
Priority is dynamically computable!

Priority is also related to other specification parameters such as

Authority

Sponsor

Source



# Part 2: Integrating benchmarks and requirement targets

Systems analysis benchmarks are integrated with setting future requirements.

This improves Competitive Analysis and Competitive Engineering Specification

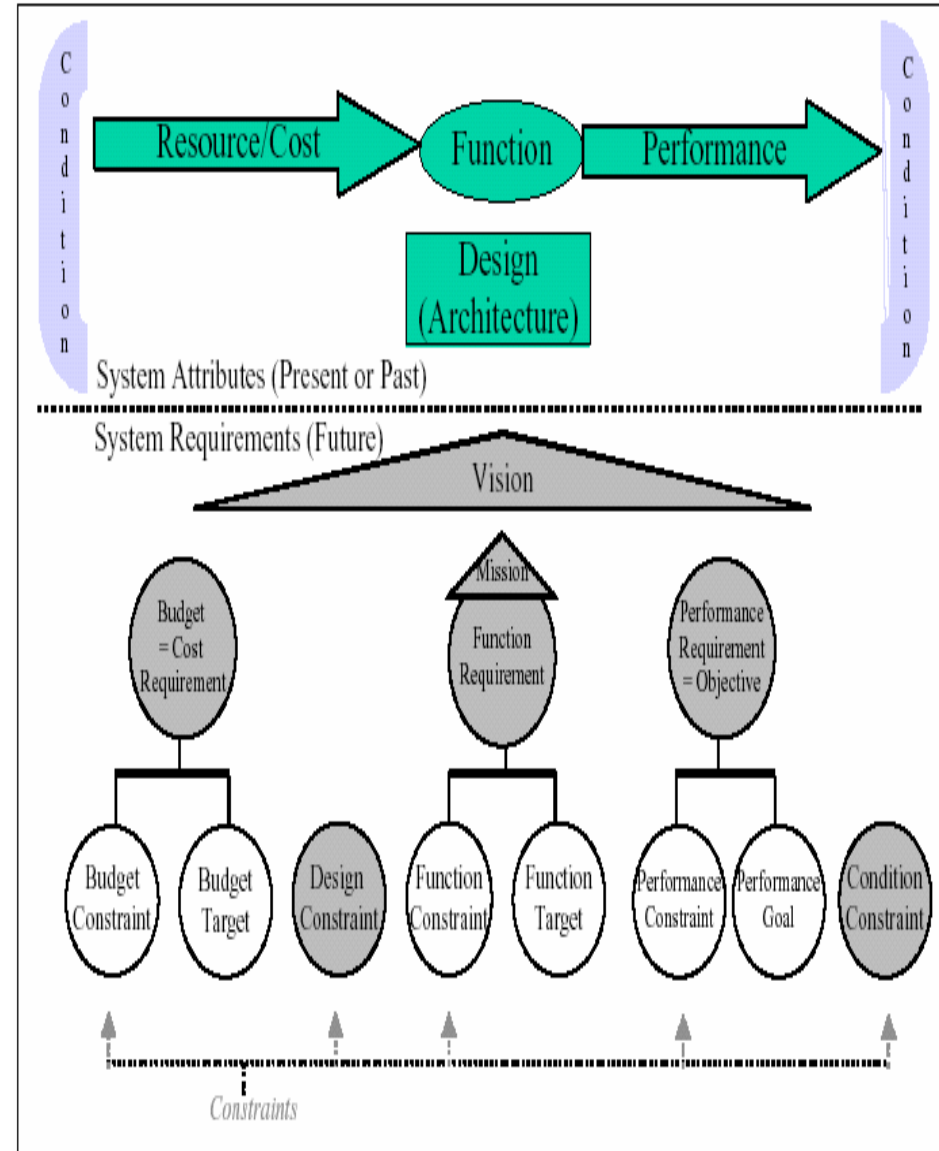
Scales: powerful flexible measures to compete with

Meters: practical ways to measure performance levels

Benchmarks: Past, Record, Trend

Targets: Goal, Stretch, Wish, Ideal

Constraints: Fail, Survival



# Systems analysis benchmarks are integrated with setting future requirements.

## Adaptability:

Type: Quality Requirement.

Scale: the calendar time in hours needed to re-configure the defined [Base Configuration] to any other defined [Target Configuration] using defined [Methods] and defined [Reconfiguration Staff].

## Expert Reconfiguration: Defined As:

{Base Configuration = Novice Setup,  
Target Configuration = Expert Setup,  
Methods = Selection of Library Reconfiguration Process,  
Reconfiguration Staff = Qualified Expert}.

## ===== Benchmarks =====

Past [Expert Reconfiguration, Version 0.3, Asian Market]: < 1 hour.

## ===== Goals (Performance Targets)=====

Authority [Goals]:Federal Drug Administration.

Goal [Expert Reconfiguration, Deadline = Version 1.0]: < 0.5 hours.

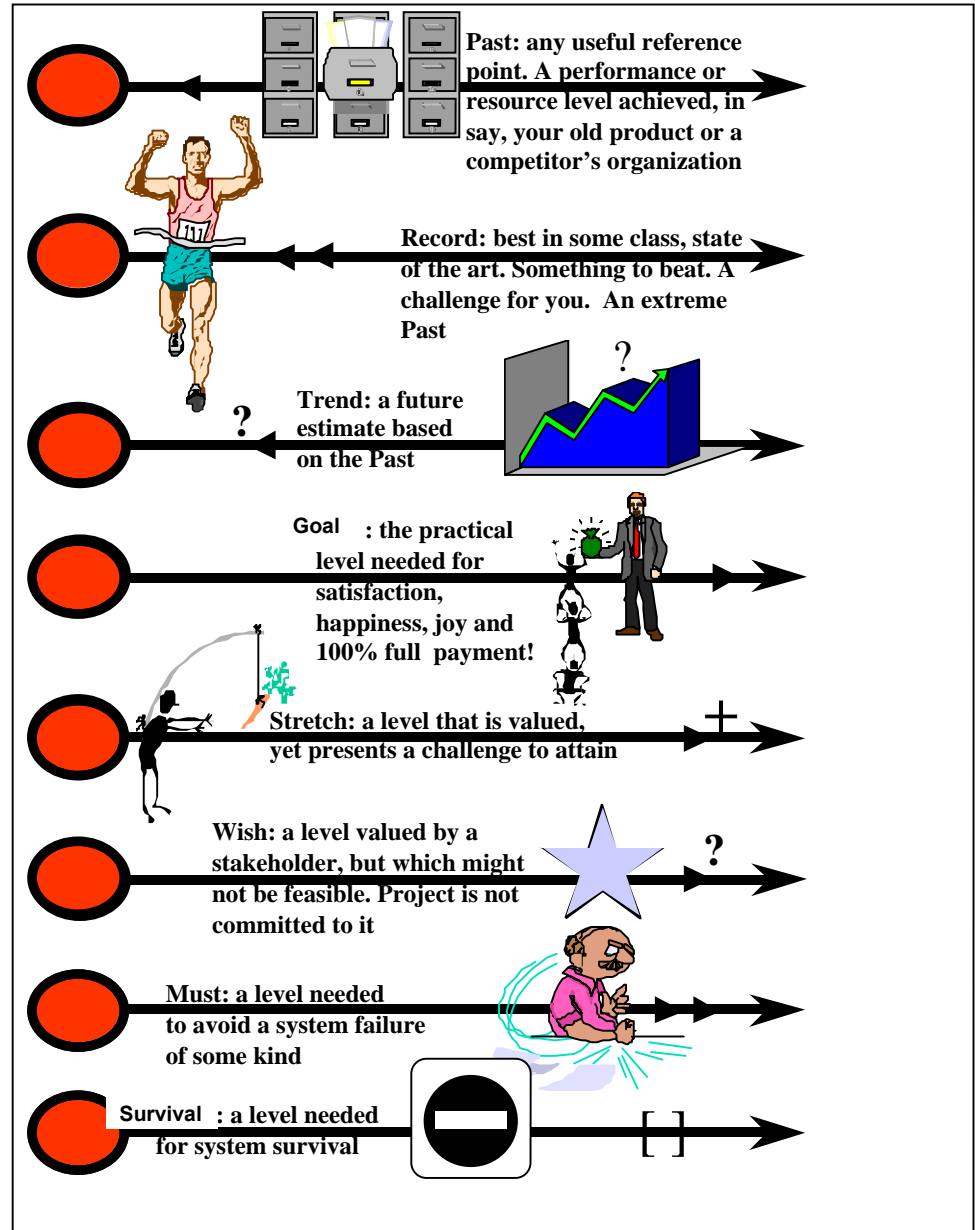
Goal [Expert Reconfiguration, Deadline = Version 2.0]: < 0.1 hours.

## ===== Constraints =====

Fail [All USA Products]: < 0.7 hours.

Fail [Expert Reconfiguration, Deadline = Version 2.0]: < 0.5 hours.

Survival [Expert Reconfiguration, European Market]: < 1 working day.



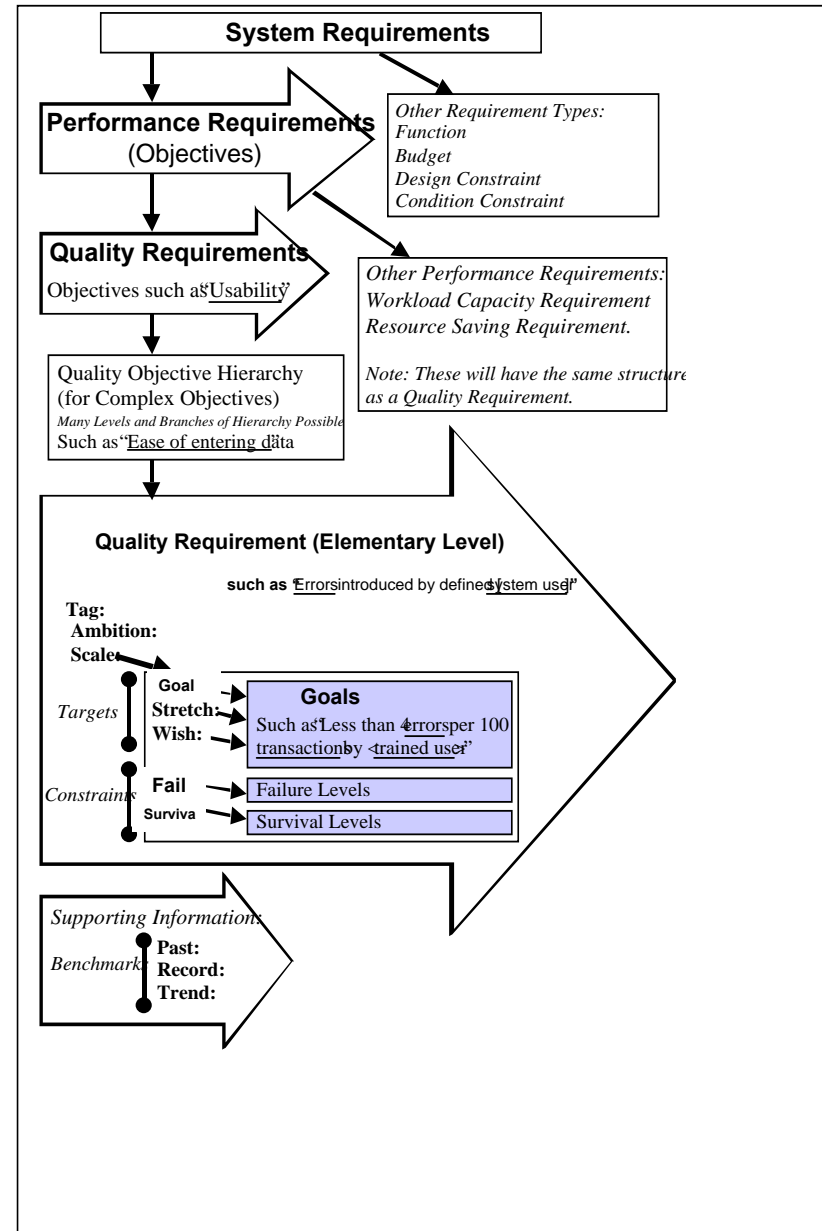


## Competitive Analysis

Make sure your own and competitor levels are analyzed and specified together with future requirements

## Competitive Engineering

Make sure you not only specify the balanced 'goal' but that marketing information about 'wishes' is captured. Make sure that the engineer is challenged by a 'stretch' goal



# Scales: Units of measure - a foundation

Scale -/-/

Concept \*132 January 9, 2003

A 'Scale' *parameter* is used to define a 'scale of measure'.

All elementary scalar attribute definitions require a defined Scale.

A Scale states the fundamental and precise *operational definition* for a specific scalar attribute.

It is used as the *basis* for expressing many of the *parameters* within the scalar attribute definition (for example, Meter, Goal and Budget):

all scalar estimates or measurements are made with reference to the Scale.

The Scale states the units of measurement, and any required scalar qualifiers.

User Friendly:

Type: Quality Requirement.

Ambition: To consistently exceed Competitor's ease of learning.

**Scale: Time to Master  
a defined [Task]  
by defined [Learner].**

Meter: <Use good academic practice, do at least 10 Tasks, with at least 5 Learner Types and at least 50 people>.

Record [Competitor AA, Product XYZ, Task = Dial Out, Learner = Novice]: 2 minutes <- Our current tests.

Goal [Our Company, Product ABC, Task = Dial Out, Learner = Novice]: < 10 seconds <- Marketing Requirement 4.5.7.

Master: Defined as: ability to pass a suitable approved test.

# Meters: practical ways to measure performance levels

*Meter* -/?/-

*Concept \*093 April 18, 2003*

A Meter parameter is used to identify, or specify, the definition of a practical measuring device, process, or test that has been selected for use in measuring a numeric value (level) on a defined Scale.

~~Meters are not the primary focus of this document~~

~~We are in the process of developing a new set of metrics~~

## Repair:

**Ambition:** Improve the speed of repair of faults substantially, under given conditions.

**Scale:** Hours to repair or replace, from fault occurrence to when customer can use faultlessly, where they intended.

**Meter [Product Acceptance]: A formal test in field with at least 20 representative cases,**

**[Field Audit]: Unannounced field testing at random.**

## ===== Benchmarks

Past [Product = Phone XYZ, Home Market, Qualified Dealer Shop]:  
{0.1 hours at Qualified Dealer Shop +  
0.9 hours for the Customer to transit to/from Qualified Dealer Shop}

Record [Competitor Product XX]: 0.5 hours average.  
"Because they drive a spare to the customer office."

Trend [USA Market, Large Corporate Users]: 0.3 hours. "As on-site spares for large customers."

## ===== Targets

Goal [Next New Product Release, Urban Areas, Personal Users]: 0.8 hours in total,

[Next New Product Release, USA Market, Large Corporate Users]: 0.2 hours

<-Marketing Requirement, 3 February This Year.

## ===== Constraints

Fail [Next New Product Release, Large Corporate Users]: 0.5 hours or less on average

<-Marketing Requirement, 3 February This Year.

# Benchmarks: Past, Record, Trend

**Past:** A relevant benchmark level already achieved by an existing system (our own, competitive, or any other system) that is worth consideration.

**Record:** A 'Past', which is the best known result [in some defined area]. A 'state-of-the-art' value.

**Trend:** An extrapolation of past data, trends and emerging technology to a defined [time and place].

*Aside from our own project's plans to improve this level, what future levels are likely to be achieved by others?*

*What will we be competing with?*

Usability [New Product Line, Major Markets]:

Ambition: To achieve a low average time-to-learn to use our telephone answerer, under various conditions.

Scale: Average number of minutes for defined [representative user and all their household family members over 5 years old] to learn to use defined [basic daily use functions] correctly.

Meter [Product Acceptance]: A formal test in field with at least 20 representative cases,

[Field Audit]: Unannounced field testing at random.

===== Benchmarks =====

**Past [Product XYZ, Home Market, People between 30 and 40 years old, in homes in Urban Areas, <For one explanation & demo>]: 10 minutes.**

**Record [Competitor Product XX, Field Trials]: < 5 minutes?> <- one single case reported,**

**Trend [USA Market, S Corporation, By Initial Release]: 10 seconds <- Public Market Intelligence Report.**

===== Constraint =====

Must [Next New Product Release, Children over 10]: 5 minutes  
<- Marketing Requirements 3 February Last Year.

===== Targets =====

Plan [Next New Product Release, Urban Areas, Personal Users]: 5 minutes total,

[Next New Product Release, USA Market, Large Corporate Users]: 5 minutes <- Marketing Requirements 3 February Last Year.

Stretch [Next Year]: (Record - 10%).

# Targets: Goal, Stretch, Wish, Ideal

**Goal:** A future required level under [defined conditions], which at least has to be achieved to claim success in meeting a requirement. A signal to stop investing in levels better than this level; because the value gained is insufficient to justify additional costs.

**Budget:** a 'Goal' level for costs.

**Stretch:** A future desired and valued level, under [defined conditions], which is designed to challenge people to exceed Plan levels.

**Wish:** A future desired level, which is valued by a stakeholder. The requirement is not planned or promised yet; due to technical or cost reasons – or lack of evaluation, but it is recorded, and kept in the requirement database (even if not acceptable now), so that it can be borne in mind as a future competitive opportunity.

**Ideal:** a future desired level which is perfect.

Usability [New Product Line, Major Markets]:  
 Ambition: To achieve a low average time-to-learn to use our telephone answerer, under various conditions.  
 Scale: Average number of minutes for defined [representative user and all their household family members over 5 years old] to learn to use defined [basic daily use functions] correctly.  
 Meter [Product Acceptance]: A formal test in field with at least 20

Resource Targets: Wish Stretch Budget  
 Performance Targets: Goal Stretch Wish

Resource Constraints: Survival Fail Survival  
 Performance Constraints: Survival Fail Survival

=====  
**Targets**  
 =====

**Goal [Next New Product Release, Urban Areas, Personal Users]: 5 minutes total,  
 [Next New Product Release, USA Market, Large Corporate Users]: 5 minutes <- Marketing Requirements 3 February Last Year.**

**Stretch [Next Year]: (Record - 10%).  
 Wish [Ultimately] <few seconds>  
 Ideal: 0 seconds.**

# Targets

Resource

Targets:

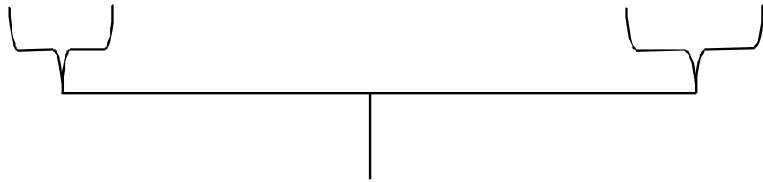
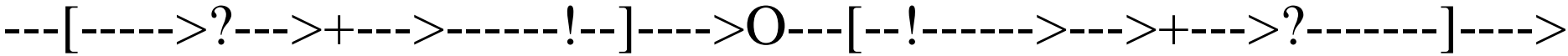
Wish Stretch Budget



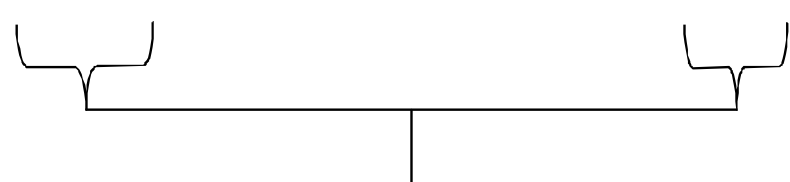
Performance

Targets:

Goal Stretch Wish



Resource  
Constraints:  
Survival                      Fail   Survival



Performance  
Constraints:  
Survival   Fail                      Survival

# Constraints: Fail, Survival

## Fail

Concept \*098 April 21, 2003

'Failure' signals an undesirable and unacceptable system state.

A Fail parameter is used to specify a Fail level constraint; it sets up a failure *condition*.

A Fail level specifies a point at which a system or attribute failure state can occur.

A single specified number (like Fail: 90%) is assumed to be the leading edge of a Failure Range.

## Survival Concept \*440 March 3, 2003

Survival is a state where the system can exist.

Outside the survival range is a 'dead' system caused by a specific attribute level being outside the survival range.

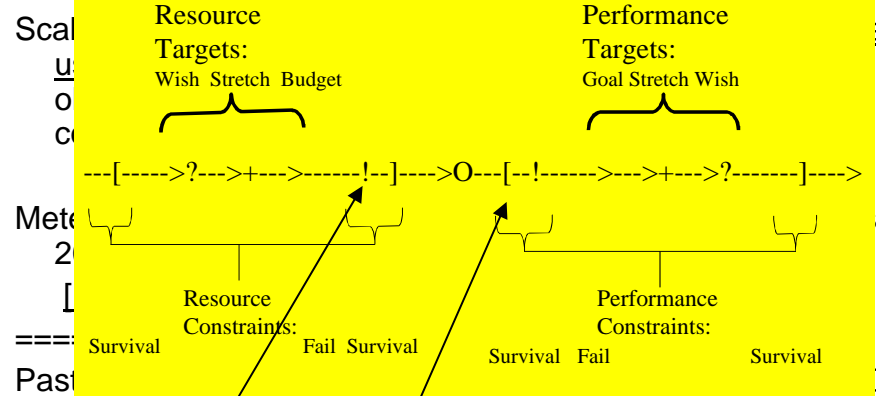
For example, 'frozen to death' or 'suffocated'.

A Survival *parameter* specifies the upper or lower acceptable limits under specified conditions [time, place, event], for a scalar attribute.

It is a *constraint* notion used to express the attribute levels, which define the survival of the entire system.

Usability [New Product Line, Major Markets]:

Ambition: To achieve a low average time-to-learn to use our telephone answerer under various conditions



years old, in homes in Urban Areas, <For one explanation & demo>: 10 minutes.

Record [Competitor Product XX, Field Trials]: < 5 minutes?>  
<- one single case reported,

[USA Market, S Corporation]: 10 seconds <-  
Public Market Intelligence Report.

=====  
**Constraints**=====

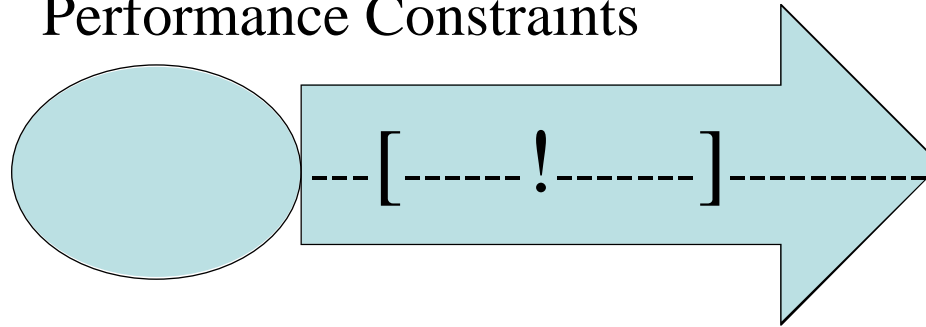
**Fail [Next New Product Release, Children over 10]: 5 minutes**

**<- Marketing Requirements 3 February Last Year.**

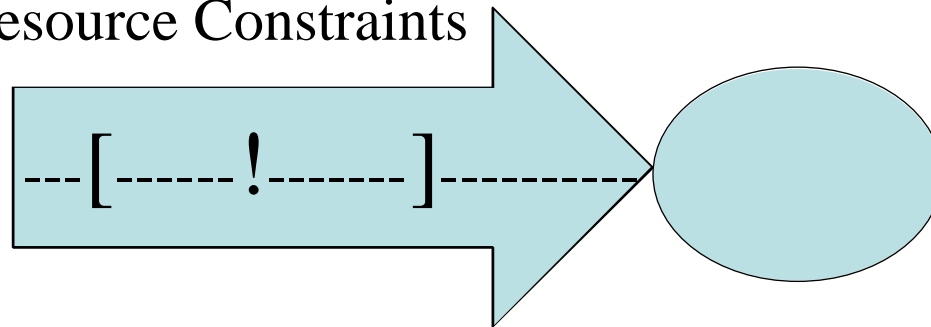
**Survival [Next New Product Release, Children over 10]: 10 minutes**

# Constraint Graphics

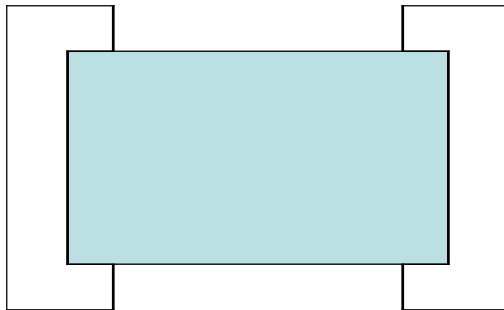
Performance Constraints



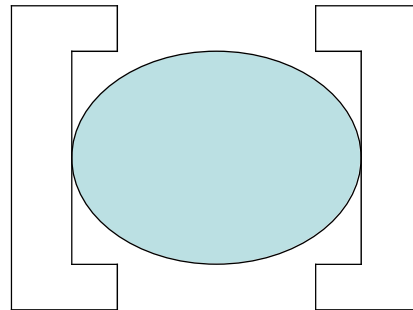
Resource Constraints



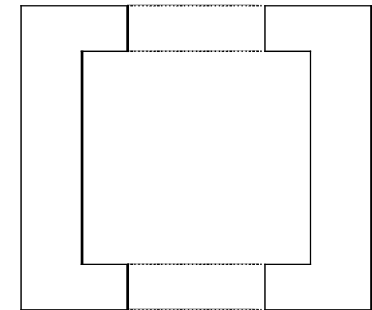
Design Constraint



Function Constraint



Condition Constraint





# Enthoven on Numbers

“Numbers are a part of our language.

Where a quantitative matter is being discussed,

the greatest clarity of thought is achieved by using numbers instead of avoiding them, *even when uncertainties are present.*

*This is not to rule out judgment and insight.*

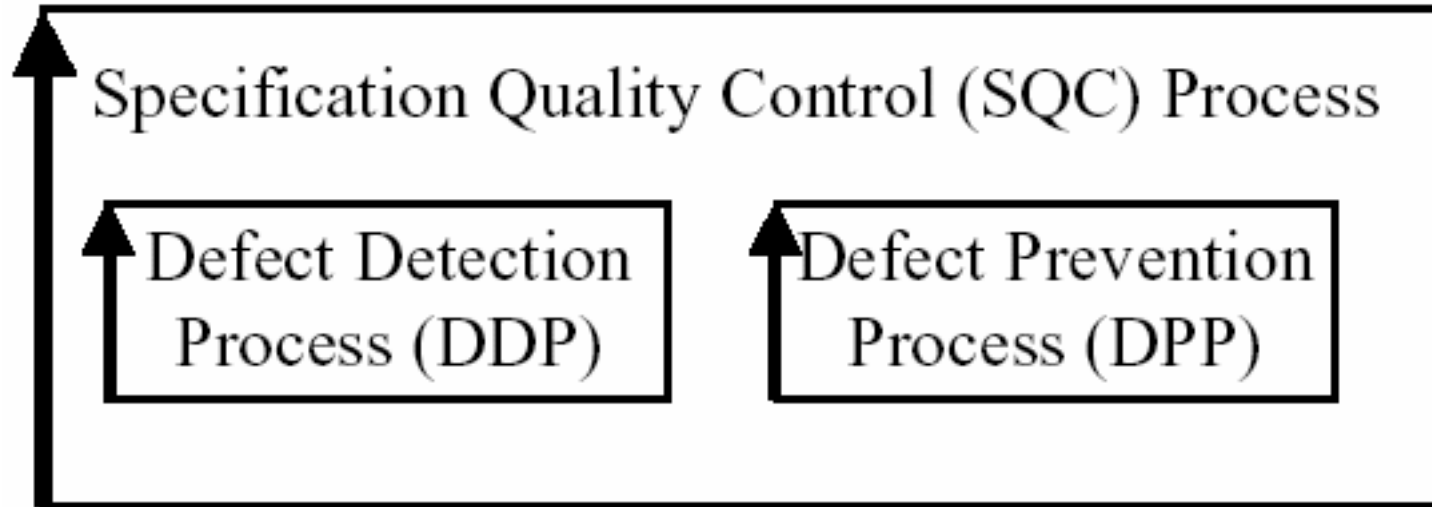
*Rather, it is to say, that judgments and insights need, like everything else, to be expressed with clarity if they are to be useful.”*



**Alain Enthoven**, June 1963, Naval War College, Newport Rhode Island (see note for more detail), Hughes98, Rescuing Prometheus p164

See the note for more detail on Enthoven

# Part 3: Quantified Quality Control of specifications



Quality Control of Specification (SQC)

The quantified Exit and Entry controls

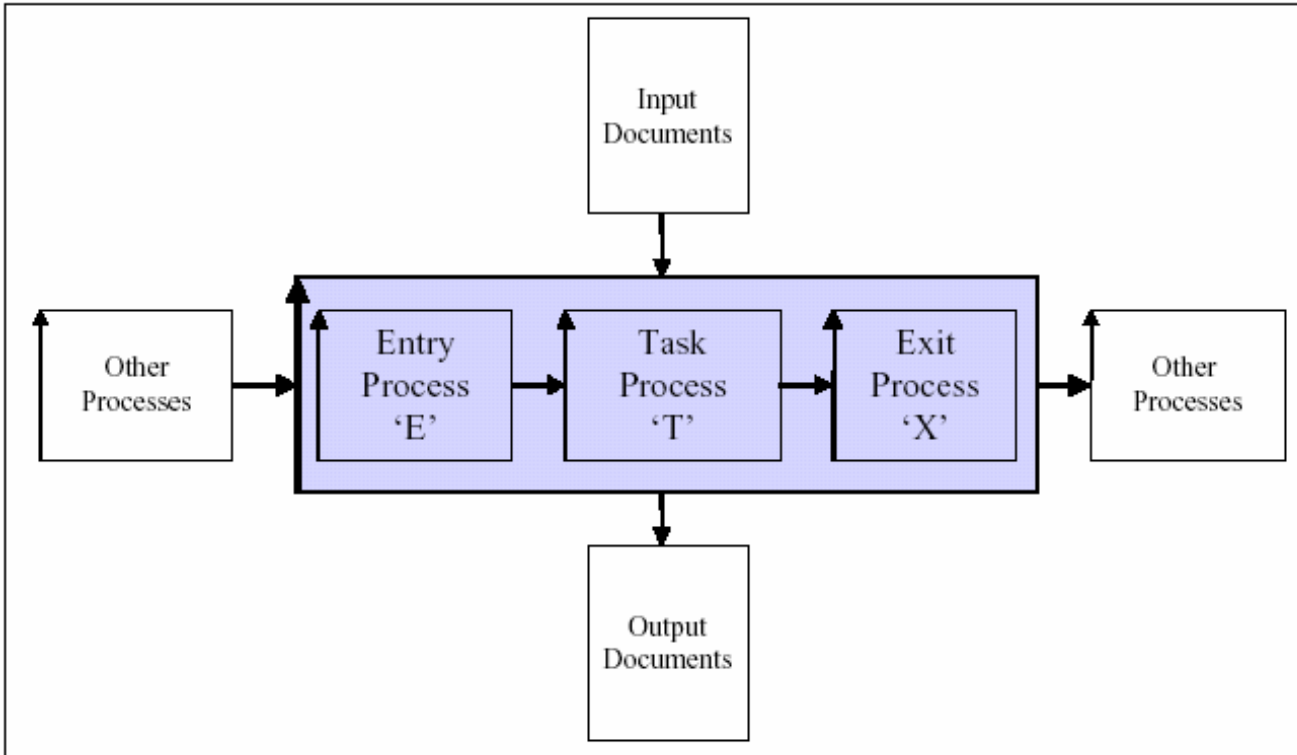
Reviewing the Quality of a specification's 'Competitiveness'

How does Planguage help QC?

How does Planguage help Reviews?

How does QC impact competitiveness?

# Quality Control of Specification (SQC)



Spec QC is done

when the input (other) work process meets entry conditions (E)

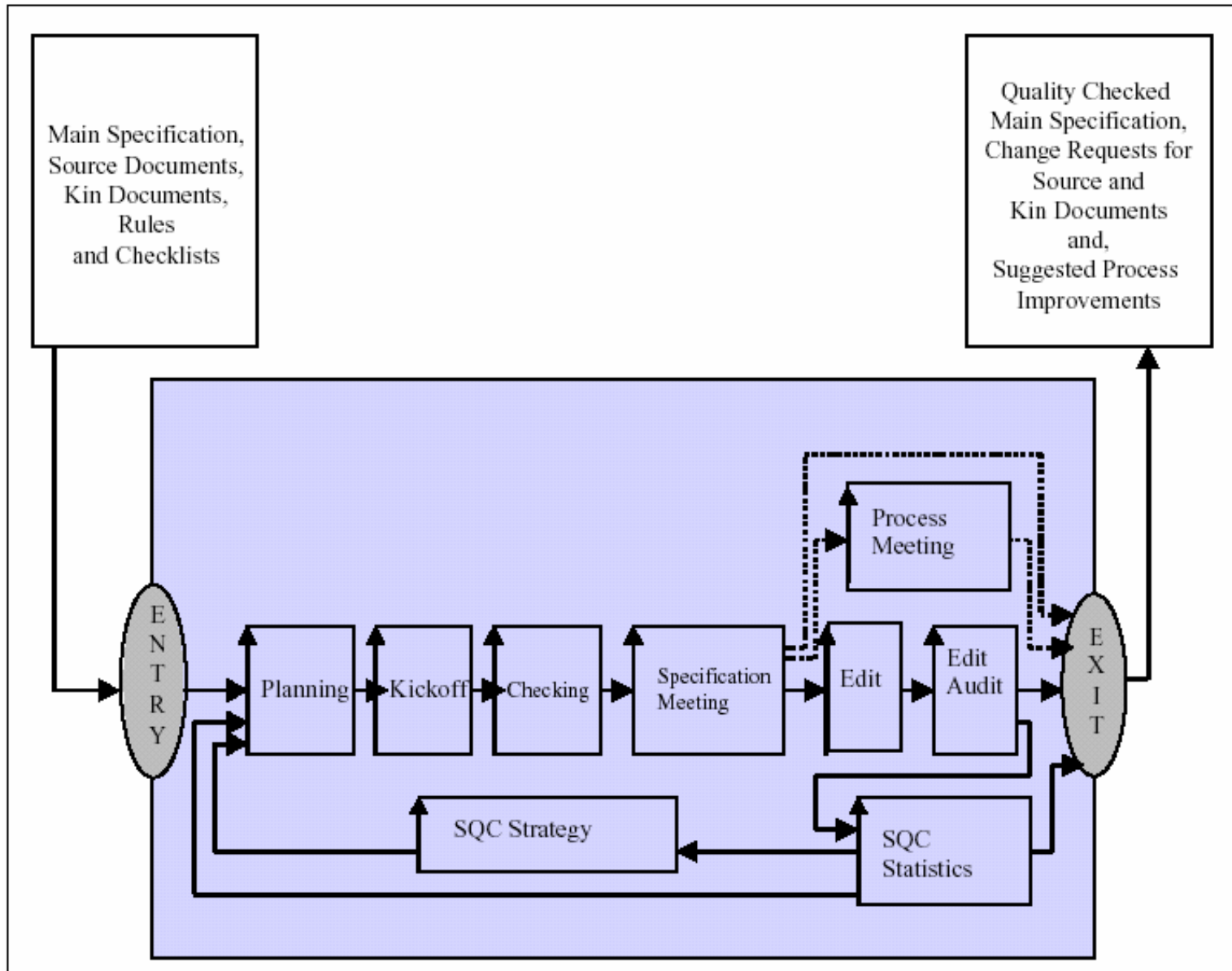
According to a defined QC process (T)

And is released to other process when exit conditions are met (X)

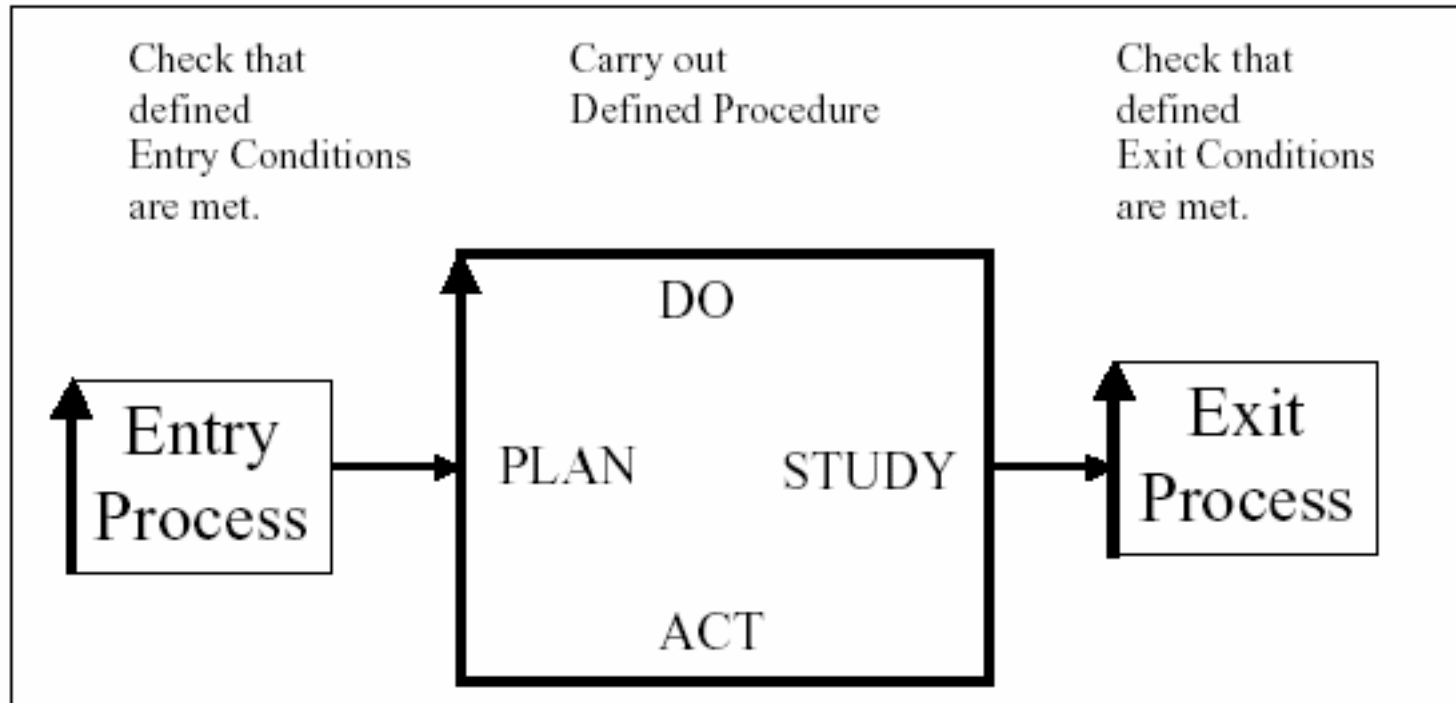
And is done by comparison with other related documents and spec rules  
(Input)

Producing reports and process control statistics (Output)

# Quality Control of Specification:



# The quantified Exit and Entry controls



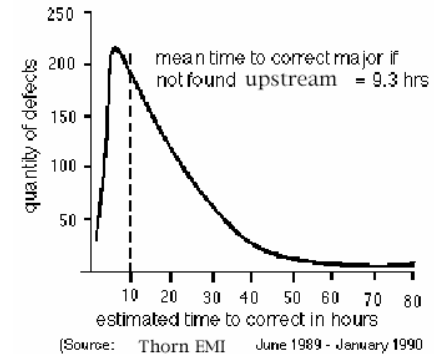
Entry and Exit Condition example:

Maximum estimated 1.0 Major defects per logical page remaining.

# The quantified Exit and Entry controls (2)

Assumptions:

- 1) 30 major defects/page have been found during SQC.
- 2) Your SQC effectiveness is 60% and your SQC is a statistically stable process).
- 3) One sixth of your attempts to fix defects fail (One sixth is average failure to fix.)
- 4) New defects are injected during your attempts to fix defects at 5%.
- 5) The uncertainty factor in the estimation of remaining defects is  $\pm 30\%$ .



**Probably remaining major defects in each (logical) page =**

‘probably unidentified majors’ + ‘bad fix majors’ + ‘majors Injected’

Let E = Effectiveness expressed as a percentage (%) = 60%

Probably unidentified majors = major defects acknowledged-by-editor for each page at Edit \* (100 – E) / E  
= 30 major defects/page found \* (100 - 60) / 60 = **20 major defects/page.**

Bad Fix Majors = One sixth of fixed majors = So, of 30 attempted fixes,

**5 major defects in each page** are not fixed.

Majors Injected = 5% of majors attempted to be fixed = **1.5 major defects/page.**

**Probably remaining major defects/page = 20 + 5 + 1.5 = 26.5 remaining major defects/page**

Taking into account the uncertainty factor of  $\pm 30\%$  and rounding down to the nearest whole

number gives **26  $\pm$  7 Remaining Major Defects/Page**

(Minimum = 19, Maximum = 33 remaining major defects/page).

# Reviewing the Quality of a specification's 'Competitiveness'

## Entry Condition:

Low-defect exit from  
*Specification Rules QC*

So it is complete, clear,  
consistent, correct

## Different people (Senior)

*Different Rules, ask*

About idea value  
About other investments  
About competition  
About economics  
About risks

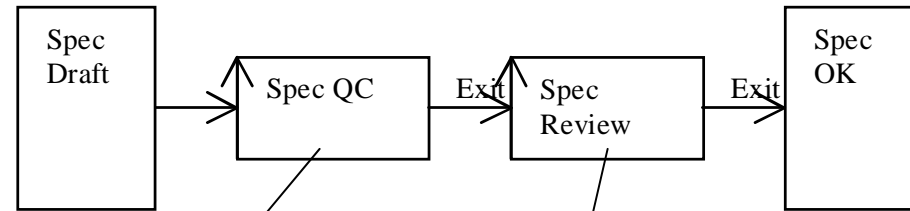
## Different Evaluation

*Not 'defects'*

*Go or no go to next stage of  
development*

*Responsible recommendations*

*Status set (Approved, ...)*



## QC & Spec Rules

1. Performance requirements must be quantified
2. Sources must be specified for all details
3. Unambiguous to readership
4. Clear enough to test
5. Consistent with sources and siblings

## Competitiveness Rules.

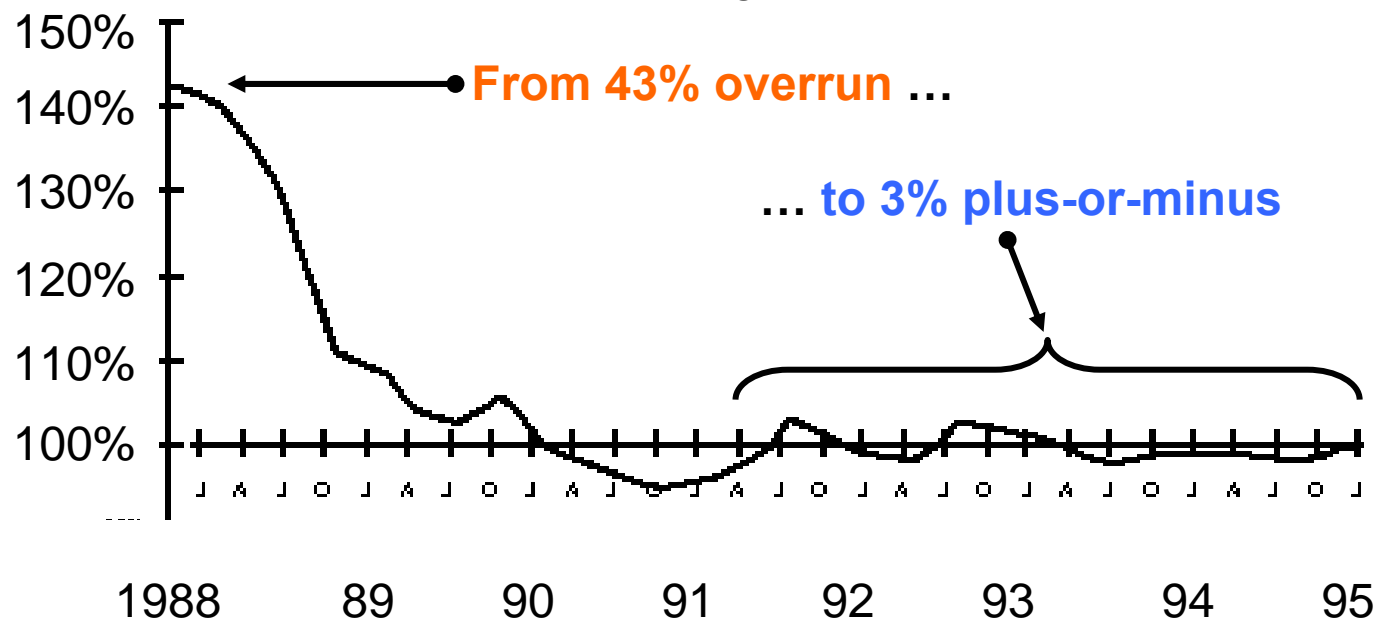
1. Number one in market performance levels
2. Number one in cost levels
3. Number one in service levels
4. Number one in distribution capability

# How does Planguage help Spec Quality Control?

## Planguage:

- Provides specific standards to check for defects (rules, exit conditions, entry conditions)
- Provides well defined and integrated processes for QC and all related processes of specification and project management
- Contains structures which enable efficient cross checking of information by people and computers.
- Contains a consistent set of standards and concepts for all types of specification - 'once learned applies to all'

Cost at Completion as a % of Budget

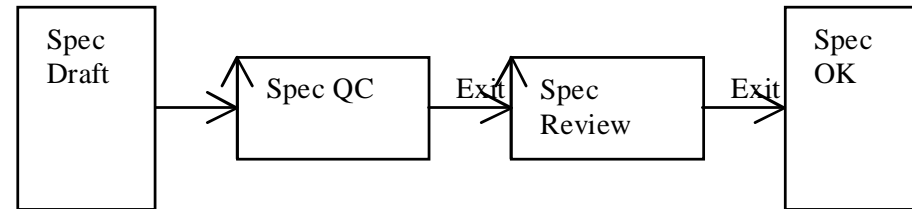


***Achieving Project Predictability at Raytheon***



# How does Planguage help Reviews?

It ensures



*intelligible and consistent specifications*

Numeric exit from SQC *before* review

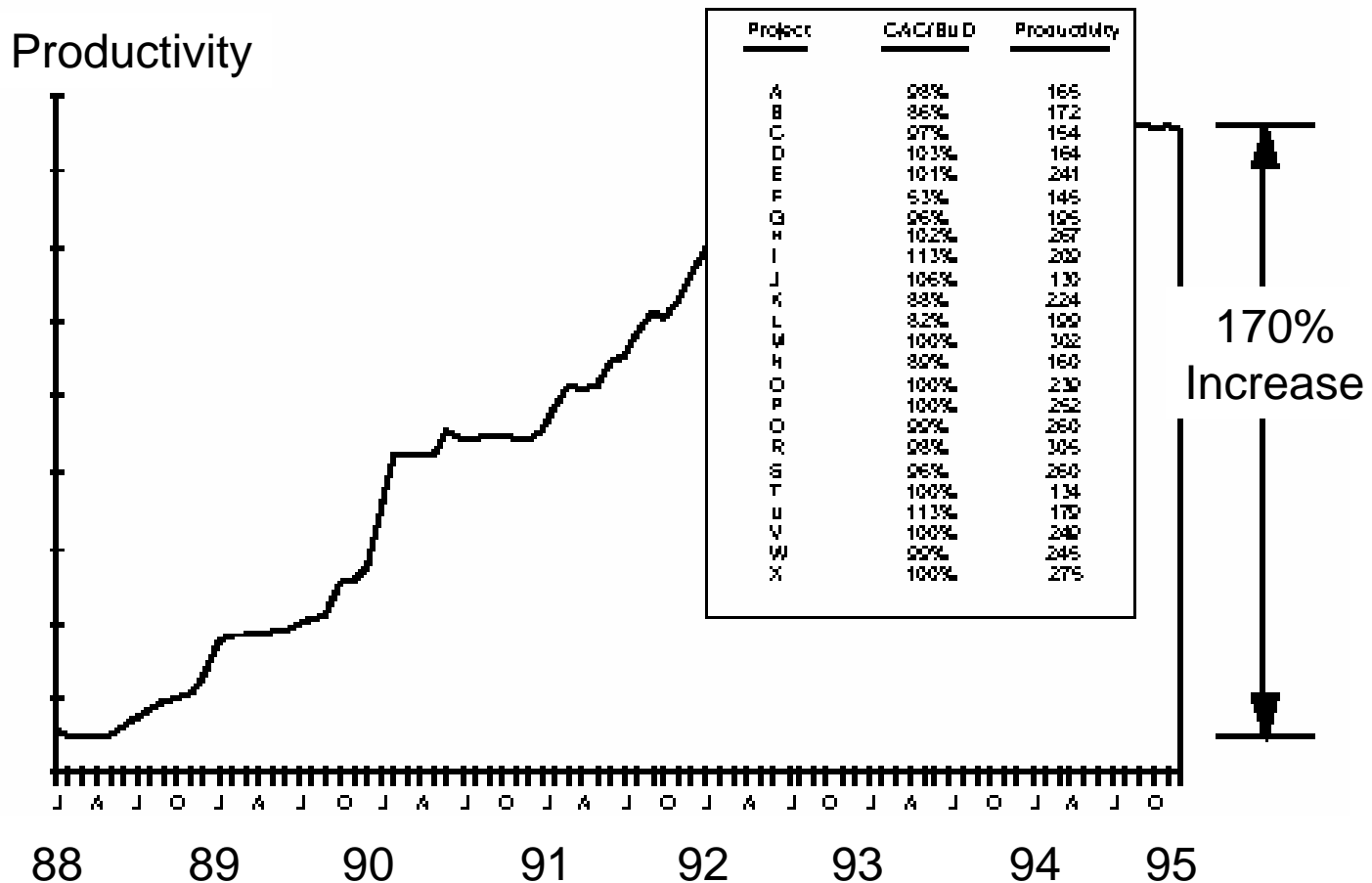
so that reviews are based on a solid foundation - and do not waste senior people's time, with sloppy work

# How does Spec QC impact competitiveness?

Indirectly

By avoiding rework (40%+ of total project cost if you are not careful!)

Speeds up projects by factor 2 to 3 (ex. Raytheon 95 SEI, below)



# POSSIBLE PURPOSES FOR USING SQC

- Reducing Time-to-Delivery
- Measuring the Quality of a Document
- Measuring the Quality of the Process producing the Document
- Enabling Estimation of the Number of Remaining Defects
- Identifying Defects
- Removing Defects
- Preventing additional 'Downstream' Defects being generated by removing existing Defects
- Improving the Engineering Specification Process
- Improving the SQC Process
- On-the-Job Training for the Checkers
- Training the SQC Team Leader
- Certifying the SQC Team Leader
- Peer Motivation
- Motivating the Managers
- Helping the Specs Writer
- Reinforcing Conformance to Standards
- Capturing and Re-using Expert Knowledge (by use of Rules and Checklists)
- Reducing Costs
- Team Building
- Fun – a Social Occasion

# Part 4: Impact Estimation Tables for quantified evaluation of design

- What is a 'design'?  
(architecture, solution)
- What are the principles of evaluating a design?
- How do we evaluate a single dimension of impact?
- How can we evaluate all dimensions of impact?
- What uses can we put impact estimation to?
- How does Impact Estimation relate to Planguage?
- How do we specify a design with impacts?

Potential Design Solutions		Design Idea 1	Design Idea 2
		'Standard' Pen	Laptop
<b>Requirements</b>			
<i>Binary - Function Target</i>			
<u>Function 1</u>	Recording Information	Yes	Yes
<i>Binary - Design Constraint</i>			
<u>Design Constraint 1</u>	Titanium Casing	No, Fail	Yes
<i>Binary - Condition Constraint</i>			
<u>Legal Constraint 1</u>	Legal in the UK	Yes	Yes
<i>Scalar - Performance Target</i>			
<u>Performance 1</u>	Portability	20g	1Kg
<i>Scalar - Budget Target</i>			
<u>Resource 1</u>	Financial Cost	GBP 1	GBP 2.5K

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

Design Idea Concept \*047 March  
15, 2003

**A design idea is anything that will satisfy some requirements.**

**A set of design ideas is usually needed to solve a 'design problem'.**

A design is a specific idea about how to solve a defined design problem.

A design (or design idea: synonym) may be in our minds, spoken aloud, found to exist in existing systems, and it may be formally or informally specified (design specification).

A Design is a 'consciously selected means' to reach defined 'ends'.

A design idea must be consistent with a set of requirements, all at once.

It must positively serve the improvement of at least one item towards specified requirements.

But it must also not violate any other constraint (function, condition constraint, scalar constraint ) which it can impact.

A design is different from a requirement in that it can in principle be changed at any time for a better design, which better meets the requirements.

Design is not holy and fixed.

## SCALAR REQUIREMENT SPECIFICATION

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

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

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

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

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

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

Past: 90%, Goal: 95%.

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

Past: 45%. Plan: 60%.

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

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

## DESIGN SPECIFICATION (simple version)

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

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

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

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

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

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

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

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

**Tag:** OPP Integration.

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

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

Version:

Status:

Quality Level:

Owner:

Expert:

Authority:

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

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

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

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

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

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

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

Reuse of Other Design:

Reuse of this Design:

Design Constraints:

Sub-Designs:

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

Impacts [Intended]: Interoperability.

Impacts [Side Effects]:

Impacts [Costs]:

Impacts [Other Designs]:

Value:

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

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

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

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

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

Dependencies:

**Risks:** <none identified>.

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

Priority:

Issues:

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

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

# Design Specification Template <with Hints>

Tag: <Tag name for the design idea>

Type: {Design Idea, Design Constraint}.

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

Version: <Date or version number>.

Status: <{Draft, SQC Exited, Approved}>.

Quality Level: <Maximum remaining major defects/page, sample size, date>.

Owner: < Role/e-mail/name of person responsible for changes and updates>.

Expert: < Name and contact information for a technical expert, in our organization or otherwise available to us, on this design idea>.

Authority: <Name and contact information for the leading authorities, in our organization or elsewhere, on this technology or strategy. This can include references to papers, books and websites>.

Source: <Source references for the information in this specification. Could include people>.

Gist: <Brief description>.

Description: <Describe the design idea in sufficient detail to support the estimated impacts and costs given below>.

Stakeholders: <Prime stakeholders concerned with this design>.

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

Reuse of Other Design: <If a currently available component or design is specified, then give its tag or reference code here to indicate that a known component is being reused>.

Reuse of This Design: <If this design is used elsewhere in another system or used several times in this system, then capture the information here>.

Design Constraints: <If this design is a reflection of attempting to adhere to any known design constraints, then that should be noted here with reference one or more of the constraint tags or identities>.

Sub-Designs: <Name tags of any designs, which are subsets of this one, if any>.

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

Impacts [Functions]: <list of functions and subsystems which this design impacts attributes of>.

Impacts [Intended]: <Give a list of the performance requirements that this design idea will impact in a major way, good or bad. The positive impacts are the main justification for the existence of the design idea!>.

Impacts [Side Effects]: <Give a list of the performance requirements that this design idea will impact in a more minor way, good or bad>.

Impacts [Cost]: <Give a list of the budgets that this design idea will impact in a major way>.

Impacts [Other Designs]: <Does this design have any consequences with respect to other designs? Name them at least>.

Value: <Name or quantify value produced, and stakeholders affected by this design. Use Qualifiers>

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

For each Scalar Requirement in Impacts [Intended] (see above):

Tag: <Tag of a scalar requirement listed in Impacts [Intended]>.

Scale: <Scale for the scalar requirement>.

Scale Impact: <Give estimated or real impact, when implemented, using the defined Scale. That is, given current baseline numeric value, what numeric value will implementing this design idea achieve or what numeric value has been achieved?>.

Scale Uncertainty: <Give estimated optimistic/pessimistic or real  $\pm$  error margins>.

Percentage Impact: <Convert Scale Impact to Percentage Impact. That is, what percentage of the way to the planned target, relative to the baseline and the planned target will implementing this design idea achieve or, has been achieved? 100% means meeting the defined Plan level on time>.

Percentage Uncertainty: <Convert Scale Uncertainty to Percentage Uncertainty  $\pm$  deviations>.

Evidence: <Give the observed numeric values, dates, places and other relevant information where you have data about previous experience of using this design idea>.

Source: <Give the person or written source of your evidence>.

Credibility: <Credibility 0.0 low to 1.0 high. Rate the credibility of your estimates, based on the evidence and its source>.

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

Assumptions: <Any assumptions that have been made>.

Dependencies: <State any dependencies for this design idea>.

Risks: <Name or refer to tags of any factors, which could threaten your estimated impacts>.

Priority: <List the tag names of any design ideas that must be implemented before or after this design idea>.

Issues: <Unresolved concerns or problems in the specification or the system>.

===== Implementation Control =====

Supplier: < Name actual supplier or list supplier requirements>

Responsible: <Who in or organization is responsible for managing the supplier relation?>

Contract: <Refer to the contract if any, or the contract template>

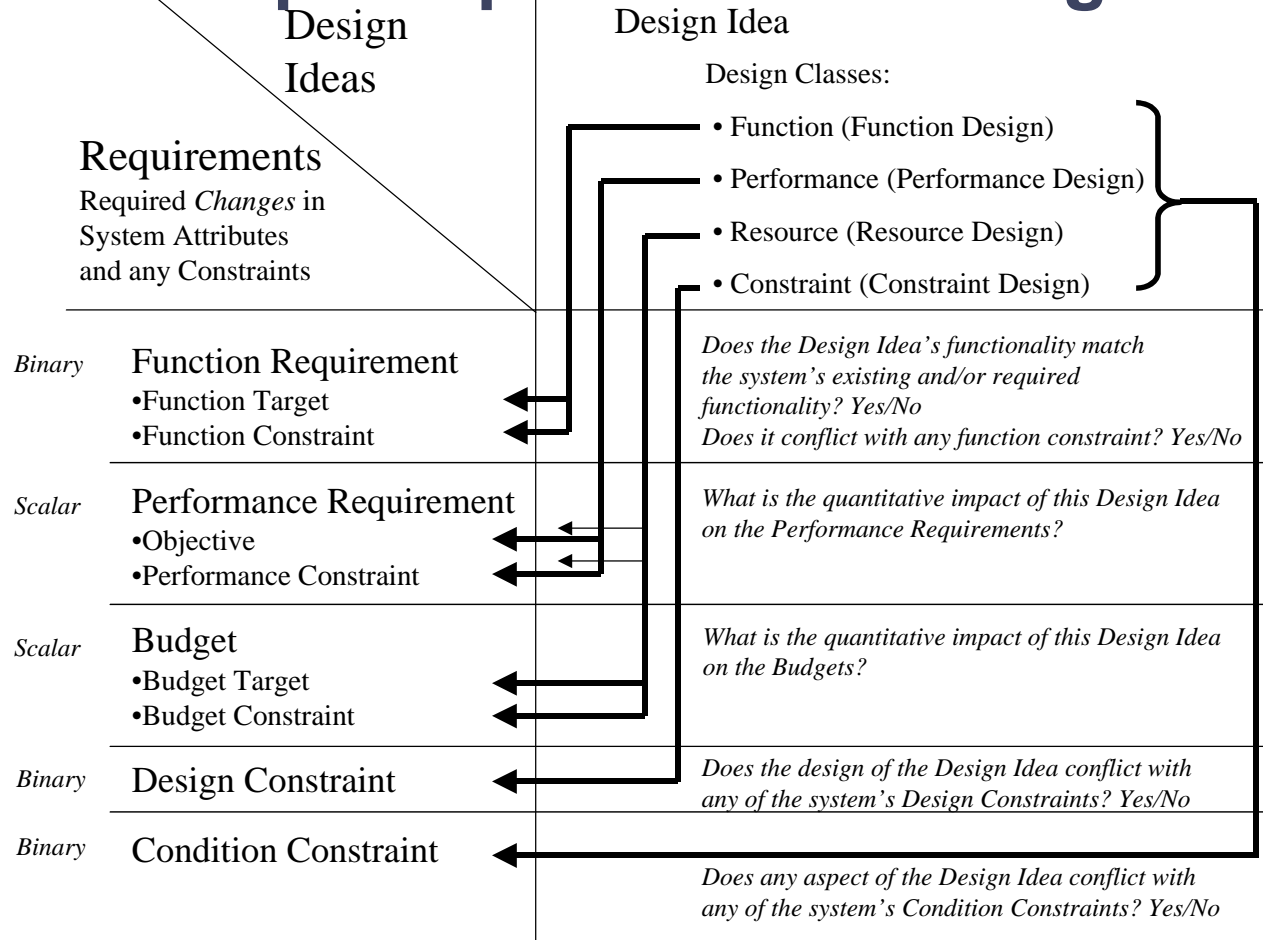
Test Plan: <Refer to specific test pan for this design>

Implementation Process: <Name any special needs during implementation>

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

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

# What are the principles of evaluating a design?

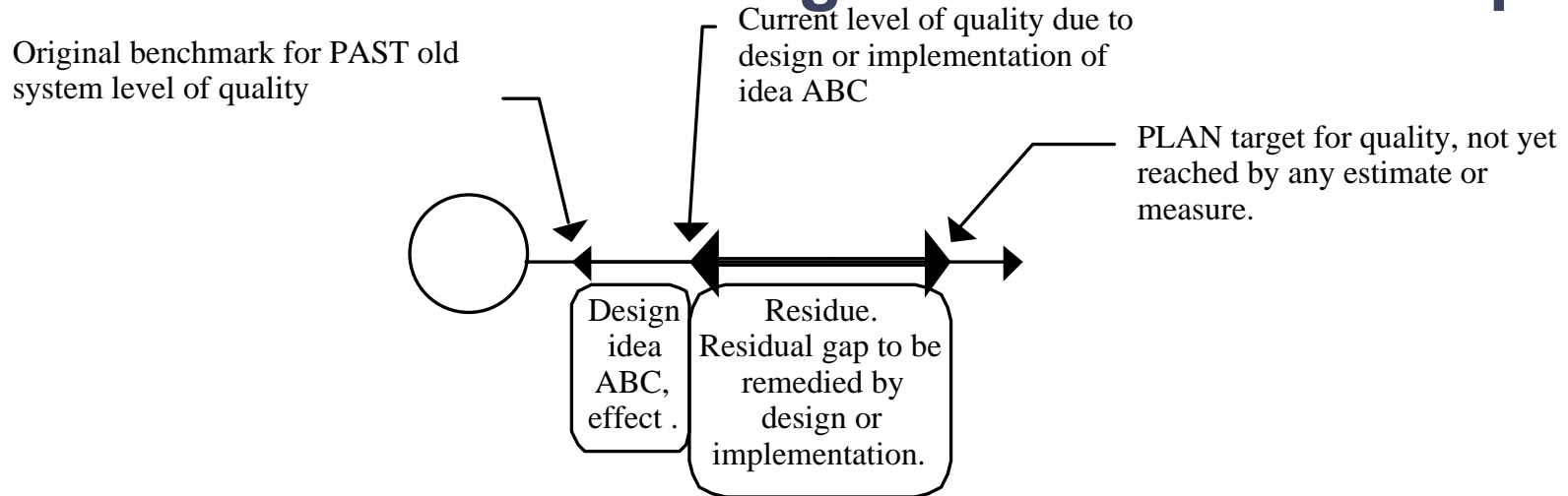


Avoid violating constraints

Meet Target and Function requirements



# How do we evaluate a single dimension of impact?



We must estimate or measure the numeric cumulative impact of the design on a defined Scale, using a defined Meter, with respect to target and constraint levels.

# How can we evaluate all dimensions of impact?

<b>Design Ideas</b>	<u>Central</u>	<u>Youth</u>	<u>Facts</u>	<u>London</u>	<u>Diploma</u>	<u>Events</u>	<u>Discounts</u>	<i>Total</i>
<b>Objectives</b>								
<u>Participation</u>	80%±50%	60%±70%	0%±50%	0%±50%	30%±50%	20%±50%	30%±50%	220%±370%
<u>Representation</u>	80%±50%	80%±50%	10%±50%	0%±50%	10%±50%	20%±50%	50%±40%	250%±340%
<u>Information</u>	0%±50%	20%±40%	80%±50%	0%±20%	20%±50%	0%±50%	0%±30%	120%±290%
<u>Conviction</u>	0%±10%	20%±50%	60%±30%	80%±50%	10%±50%	80%±50%	0%±50%	250%±290%
<u>Influence</u>	0%±50%	40%±40%	60%±50%	0%±50%	80%±50%	80%±50%	0%±50%	260%±340%
<u>Fun</u>	50%±50%	40%±50%	10%±50%	0%±0%	0%±0%	80%±50%	0%±0%	180%±200%
<b>Total</b>	210% ±260%	260% ±300%	220% ±280%	80% ±220%	150% ±250%	270% ±300%	80% ±220%	
<b>Budgets</b>								
<u>Cost</u>	10%	10%	10%	10%	1%±5%	50%±50%	80%±50%	171%±105%
<b>Benefit<sup>Sto</sup>- Cost Ratio</b>	210%/10%	260%/10%	220%/10%	80%/10%	150/1	270/50	80/80	

We can use an Impact (Estimation) Table

# What uses can we put impact estimation to?

IE can be used for a wide variety of purposes including:

1. Evaluating a single design idea. How good is the idea for us?
2. Comparing two or more design ideas to find a winner, or set of winners. Use IE, if you want to set up an argument against a prevailing popular, but weak design idea!
3. Gaining an architectural overview of the impact of all the design ideas on all the objectives and budgets. Are there any negative side effects? What is the cumulative effect?
4. Obtaining systems engineering views of specific components, or specific performance aspects.  
Are we going to achieve the reliability levels?
5. Analyzing risk: evaluating a design with regard to 'worst case' uncertainty and minimum credibility.
6. Planning evolutionary project delivery steps with regard to value and cost.
7. Monitoring, for project management accounting purposes, the progress of individual evolutionary project delivery steps and, the progress to date compared against the requirement specification or management objectives.
8. Predicting future costs, project timescales and performance levels.
9. Understanding organizational responsibility in terms of performance and budgets by organizational function.  
[In 1992, Steve Poppe pioneered this use at executive level while at British Telecom, North America.](#)
10. Achieving rigorous quality control of a design specification prior to management reviews and approval.
11. Presenting ideas to committees, management boards, senior managers, review boards and customers for approval.
12. Identifying which parts of the design are the weakest (risk analysis). If there are no obvious alternative design ideas, any 'weak links' should be tried out earliest, in case they do not work well (risk management). This impacts scheduling.
13. Enabling configuration management of design, design changes, and change consequences.
14. Permitting delegation of decision-making to teams. Teams can achieve better internal progress control using IE, than they can from repeatedly making progress reports to others, and acting on others' feedback.
15. Presenting overviews of very large, complex projects and systems by using hierarchical IE tables. Aim for a one page top-level IE view for senior management.
16. Enabling cross-organizational co-operation by presenting overviews of how the design ideas of different projects contribute towards corporate objectives. Any common and conflicting design ideas can be identified. This is important from a customer viewpoint; different projects might well be delivering to the same customer interface.
17. Controlling the design process. You can see what you need, and see if your idea has it by using an IE table. For example, which design idea contributes best to achieving usability? Which one costs too much?
18. Strengthening design. You can see where your design ideas are failing to impact sufficiently on the objectives; and this can provoke thought to discover new design ideas or modify existing ones.
19. Helping informal reasoning and discussion of ideas by providing a framework model in our minds of how the design is connected to the requirements.
20. Strengthening the specified requirements. Sometimes, you can identify a design idea, that has a great deal of popular support, but doesn't appear to impact your requirements. You should investigate the likely impacts of the design idea with a view to identifying additional stakeholder requirements. This may provide the underlying reason for the popular support. You might also identify additional types of stakeholders.

# How does Impact Estimation relate to Planguage?

## Learning:

Ambition: Make it substantially easier for our users to learn tasks <- Marketing.

Scale: Average time for a defined [User Type: default UK telesales trainee] to learn a defined [User Task: default Response] using <our product's instructional aids>.

Response: Task: Give correct answer to simple request.

Past [last year]: 60 minutes.

GN: Goal [By start of next year]: 20 minutes.

GA: Goal [By start of year after next]: 10 minutes.

	<u>On-line Support</u>	<u>On-line Help</u>	<u>Picture Handbook</u>	<u>On-line Help + Access Index</u>
<b><u>Learning</u></b> Past: 60min. <<-> Plan: 10min.				
Scale Impact	5 min.	10 min.	30 min.	8 min.
Scale Uncertainty	±3min.	±5 min.	±10min.	±5 min.
Percentage Impact	110%	100%	67% (2/3)	104%
Percentage Uncertainty	±6% (3 of 50 minutes)	±10%	±20%?	±10%
Evidence	Project Ajax, 1996, 7 min.	Other Systems	Guess	Other Systems + Guess
Source	Ajax report, p.6	World Report p.17	John B.	World Report p.17 + John B.
Credibility	0.7	0.8	0.2	0.6
Development Cost	120K	25K	10K	26K
Benefit-To-Cost Ratio	110/120 = 0.92	100/25 = 4.0	67/10 = 6.7	104/26 = 4.0
Credibility-adjusted B/C Ratio (to 1 decimal place)	0.92*0.7 = 0.6	4.0*0.8 = 3.2	6.7*0.2 = 1.3	4.0*0.6 = 2.4
Notes: Time Period is two years.	Longer timescale to develop			

**Picture Handbook:** Gist: Produce a radically changed handbook that uses pictures and concrete examples to *instruct*, without the need for *any* other text.

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

## Design Specification Template with Annotation

Tag: <Unique Name Capitalized>

Type: Design Idea.

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

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

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

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

**Constraint:** <if this design is a reflection of attempting to adhere to any known design constraints, then that should be noted here with reference one or more of the constraint tags or identities>.

### ===== **Real Expected Impact Section** =====

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

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

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

### ===== **More Formal Impact Estimation** =====

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

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

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

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

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

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

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

### ===== **Other Useful Parameters for Design Specification** =====

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

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

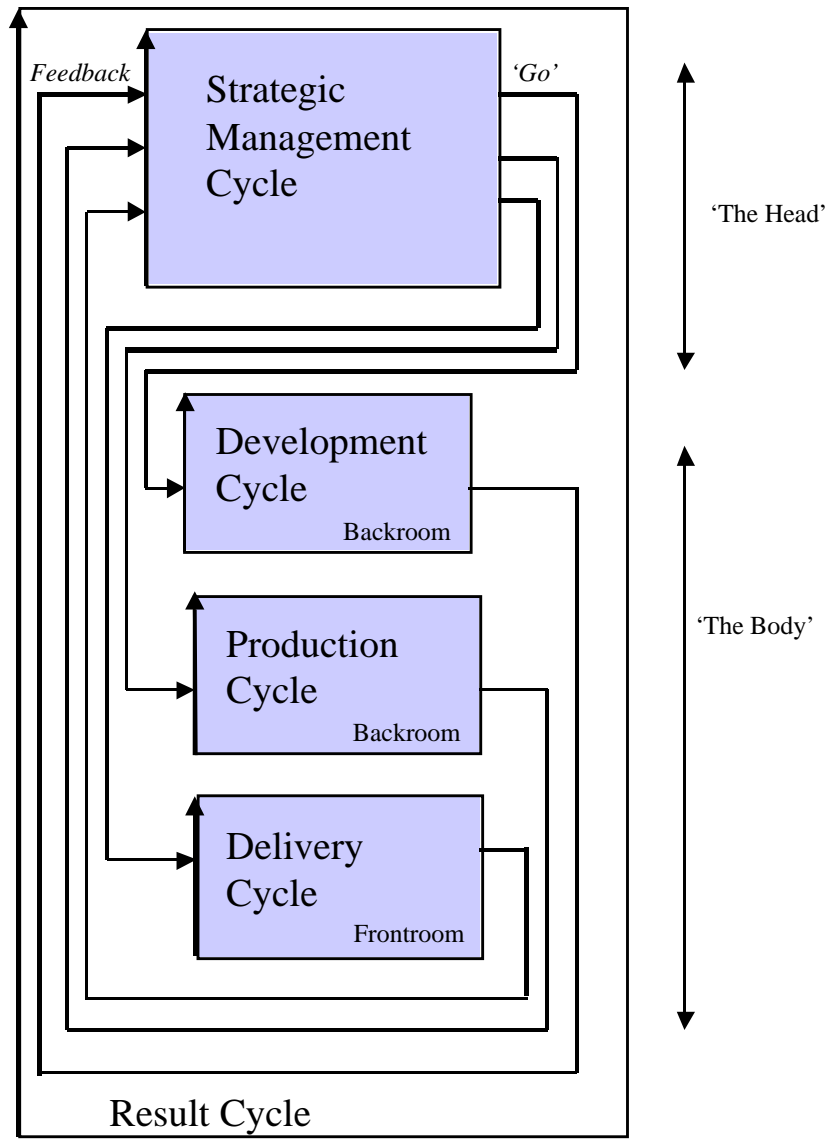
**Expert:** < name and give contact (email?) a useful technical expert in our company or otherwise available to us on this design idea>.

**Authority:** <name and give contact information to the leading authorities in our co. or elsewhere on this technology. Reference papers or books for example and websites>

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

# Part 5: Evolutionary Project Management

- The fundamentals of an Evo step
- How does Planguage support Evo project management?
- How do you plan an Evo step in Planguage?
- How does Evo relate to requirements?
- How does Evo relate to Design?
- How does Evo relate to Risk?
- How does Evo relate to process improvement?
- How does Evo relate to competitiveness?



# The fundamentals of an Evo step

An Evo step must

Deliver some planned function and/or performance values to some stakeholders

Maximize the efficiency (value to cost ratio) of the delivery

Give useful feedback before scaling up (risk management)

Give project teams practical experience in technology, engineering processes, and stakeholder feedback

## 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!

I have never seen an exception in 33 years of doing this within many varied cultures.

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 Plan 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, and do other useful cycles while you wait. This is called 'backroom concurrent engineering'.

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.

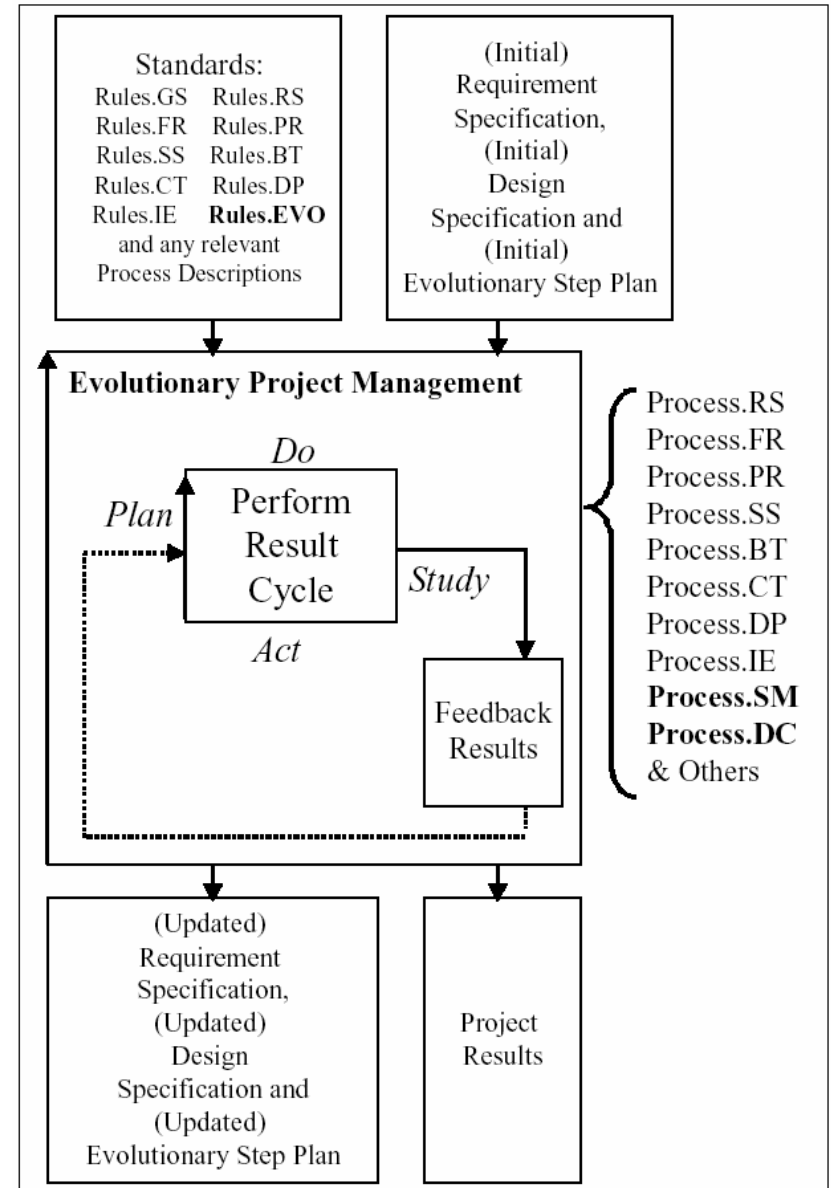
19 Talk with end users and other stakeholders in any case, they have insights you need.

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.

# How does Planguage support Evo project management?

Well-defined requirements are the *project management* result delivery targets and constraints

Well-defined *designs*, and quantified *impact estimates* help control the delivery and implementation process





# How do you plan an Evo step in Planguage?

**Step Name:** Tutorial [7777, Basic].

**Stakeholder:** Marketing, XX (<agreed, Next Friday>).

**Step Implementor:** <XX>.

**Step Content:** HCTD :<Hard Copy Text document> <- Can do 1 week MMM.

- . Basic minimal functions
- . Step by Step Instructions, in English
- . Focus on sales aspects, not how to do it (not yet, in this step)
- . Go to specific web sites
- . Pinpoint some characteristics of what we see on the terminal
- . Compared with what we see on a PC or other terminal
- . What instructions should be on the terminal to begin
- . Questionnaire for Stakeholder
- . Intended audience: Marketing
- . Process for Testing with Stakeholder (example observation, times)
- . No illustrations, just text.

**Step Value:** Stakeholder: TTT: Saleability: <some possibility of value>.

Stakeholder: Developers: <value of feedback on a tutorial>.

**Step Cost:** 10 hours per page, < 10 hours <-MMM.

**Step Constraints:** Must be deliverable within 1 calendar week.

At Least 3 hours of TTT's time for input and trial feedback.

**Step Dependencies:** <Feature list of WWW and 7777 WWW Browser> <-MMM.

# How does Evo relate to requirements?

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

Evo relates directly, measurably, testably, early and frequently to unfulfilled requirements.

Evo is always seeking the most efficient way to close the requirements gap and complete a project

The primary measure of Evo project progress is the degree of stakeholder satisfaction (in terms of agreed requirements) as a result of delivered Evo steps.

# How does Evo relate to Design?

Evo implements designs *selectively* depending on priority.

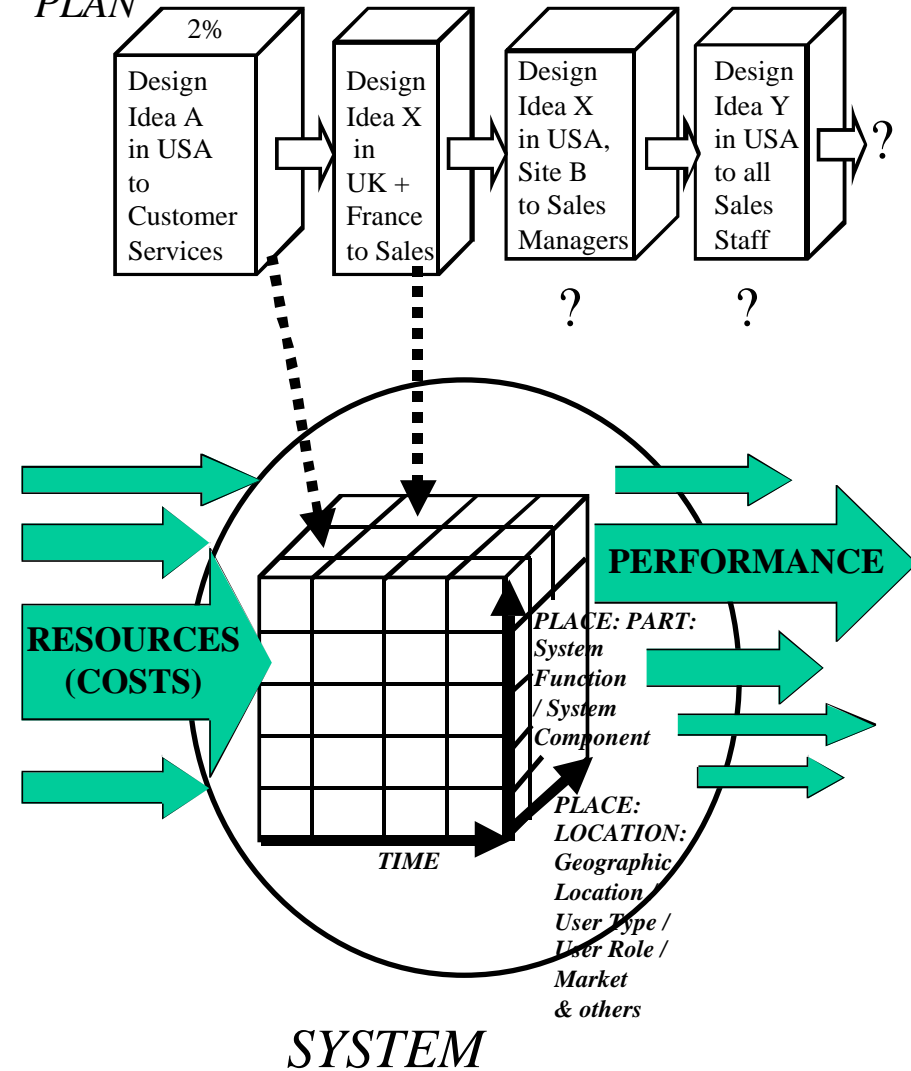
Designs can be implemented *partially* (example in one geographic market or system component) in a *single* step.

Evo allows us to be sure that the designs give *maximum value/cost*

Evo allows us to verify *by measurement* that designs deliver value/cost estimated *before* we commit on a large scale

EVO

PLAN



# How does Evo relate to Risk?

Evo reduces risk of deviation from plans

By doing projects in early and small increments

By 'learning' from practical experience

And correcting bad specifications

By grasping and integrating new opportunities outside the project (technology, customer, economics)

## BASIC EVO PLANNING POLICY

1:**Financial Budget:** No project cycle shall exceed 2% of total financial budget before delivering some measurable, required results to the user.

2:**Deadline:** No project cycle will exceed 2% of total project time (one week for a one year project) before delivering some measurable, required results to the user.

3:**Priority:** Project cycles which provide the best ratio of required results to utilized resources (highest benefit-to-cost ratios), must be delivered first to the stakeholders.

# How does Evo relate to process improvement?

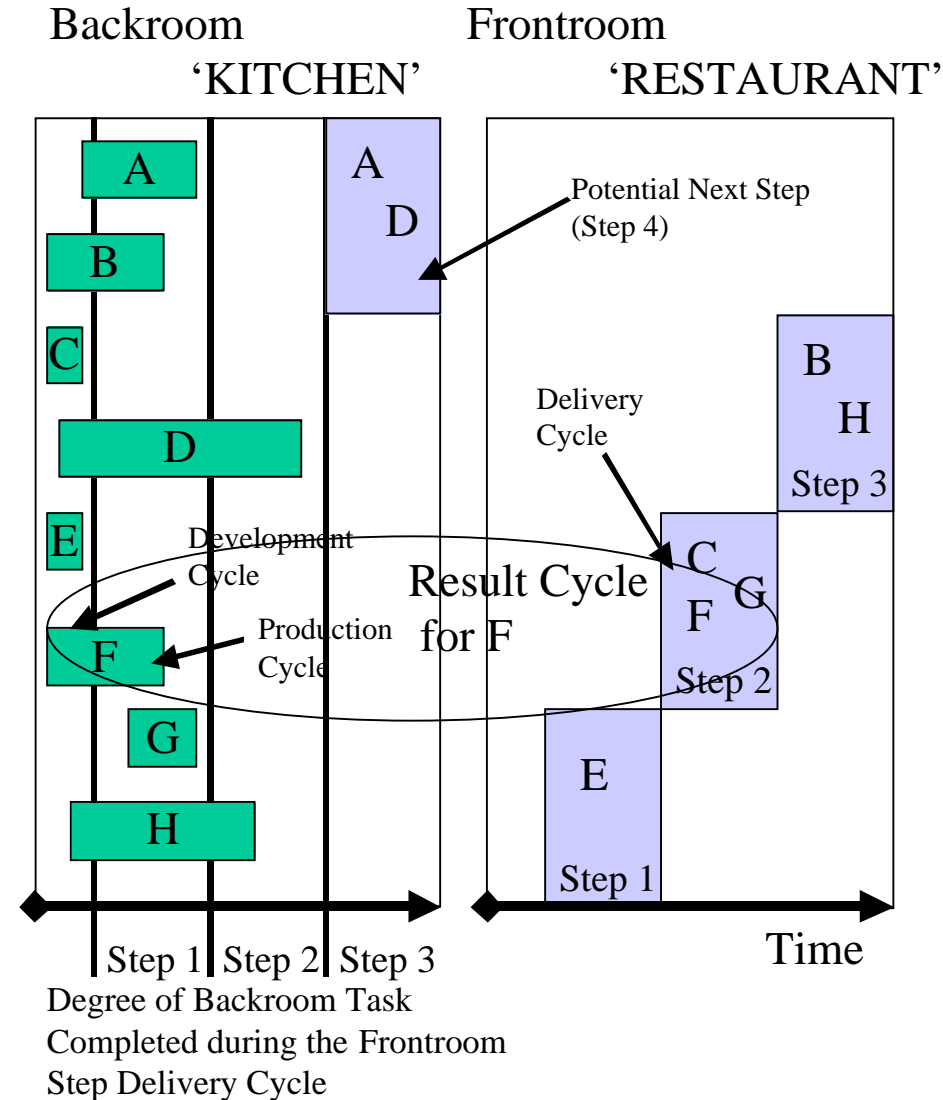
Evo can measure

the success of current processes against expectations,  
or new experimental ones against expectations

Evo can signal the need for process improvement and verify that such improvement has taken place

Evo can help you

*early* in the project,  
continuously,  
and helps to *train* new people in the adopted processes by frequent cycles of practice and feedback



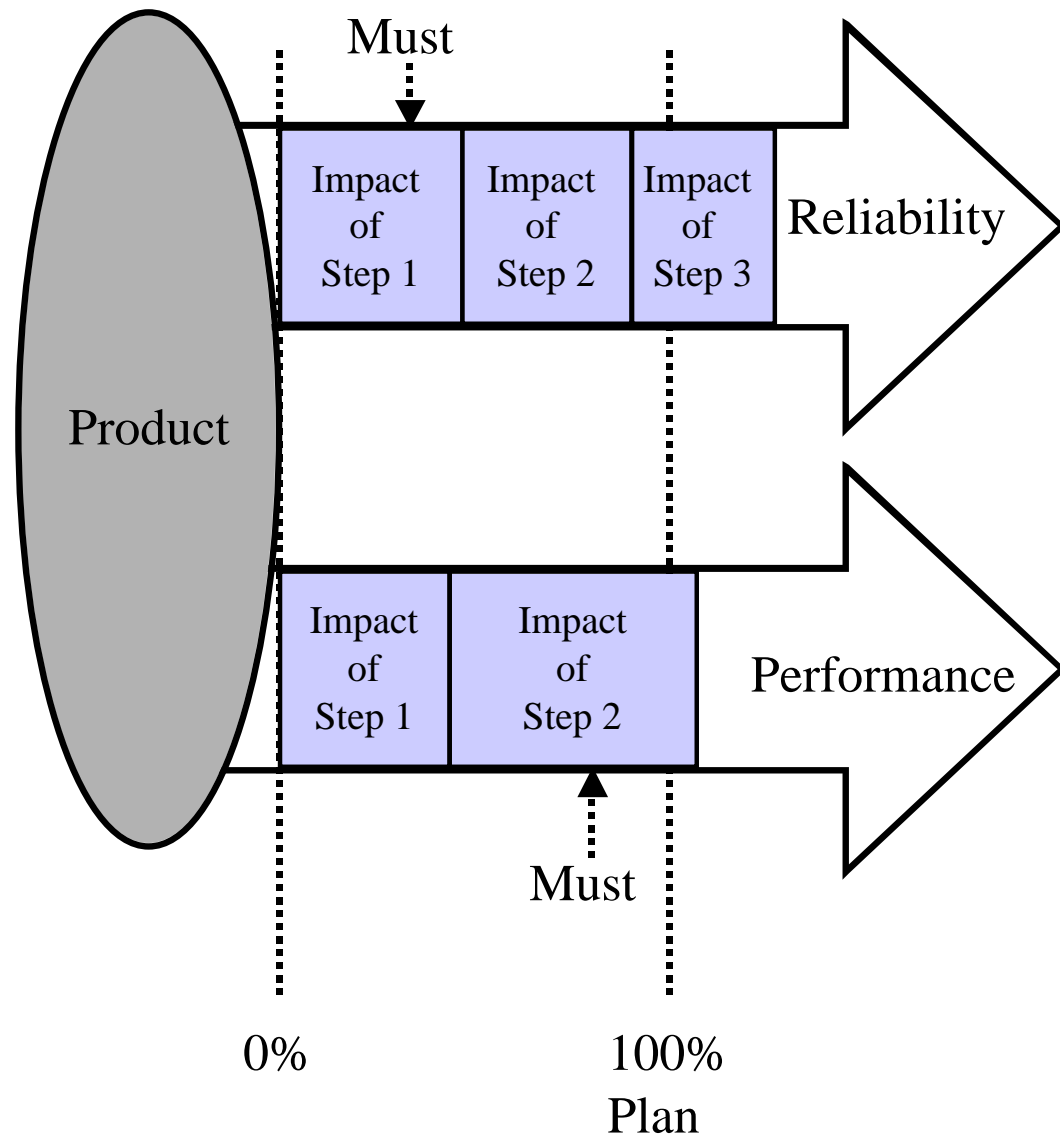
# How does Evo relate to competitiveness?

Evo is focused on delivery of quantified specified stakeholder value

Evo is 'agile'

and can change plans, designs, processes, and requirements - in order to deliver the most competitive solutions

early, gradually, and with smart priorities.



# Summary



Tom@Gilb.com

Planguage gives you tools to be more competitive  
The entire set of Planguage tools also applies to  
software engineering and top management  
planning (see 'Priority Management' book at  
[www.gilb.com](http://www.gilb.com))

Last Slide