## This is a collection of Architecture-related Concepts from the Planguage Glossary by Tom Gilb

URL: Concept Glo… http://www.gilb.com/tiki-download\_file.php?fileId=25

Book: Gilb, Competitive Engineering, 2005 (sample chapters at gilb.com)

## CE Cover.png

## Architecture (collective noun): Concept \*192. May 9 2005

The ‘architecture’ is the set of entities that

•  in fact exist and impact a set of system attributes directly,

• or indirectly, by

• constraining, or influencing, related engineering decisions.

Notes:

Interesting specializations:

• Perceivable Architecture: the architecture which is somehow directly or indirectly perceivable in a real system, as determining the range of performance and cost attributes possible. This applies regardless of who, if anyone, consciously specified the architecture design artifacts.

• Inherited Architecture: architecture which was not consciously selected at a particular level of architecture activity, but was either:

* incidentally inherited from older systems,
* accidentally inherited from specified design artifacts, specified by architects, managers or engineers.

• Specified Architecture: the formally defined architecture specifications at a given level and lifecycle point, including stakeholder requirements interpretation, architecture specification, engineering specification done by this architecture level, certification criteria, cost estimates, models, prototypes, and any other artifact produced as a necessary consequence of fulfilling the architecting responsibility.

Note: an extensive discussion of the architecture concept is given in Maier02, including a special appendix on the history of attempts to define a standard in DoD, IEEE, INCOSE. Appendix C pp283-9. In addition the book gives a great many other insights into the nature of the concept.

Note: XE "Architecture [Noun]." \*202.Architecture [noun] XE "\*202.Architecture [noun]"

The highest specified level of design ideas for a defined system is called the ‘architecture’. The architecture is the collection of controlling design ideas for a defined purpose. The architecture refers primarily to frameworks, interfaces and other technology and organizational ideas which more-detailed design ideas are expected to fit in to.

Notes:

1. The architecture specifications (\*617) would probably be classified as generic design constraints (or ‘architecture constraints’, if you wanted to emphasize the idea of ‘architecture’).

2. Architecture specifications would have priority over subsequent design decisions, made at more-specialized engineering levels.

3. ‘Architecture specification’ is the set of system-wide decisions, which are made in order to improve the systems survival ability, as it is threatened by changes to it, and by its environment.

Architecture: A high level design that provides decisions about:

purpose (What problem(s) that the product(s) will solve)

function description(s) (Why has it been decomposed into thesecomponents?)relationships between components (How do components relate inspace and time?)dynamic interplay description (How is control passed between andamong components?)flows (How does data or in-process product flow in space andtime?)resources (What resources are consumed where, in the process orsystem?).Source: Standard: FAA-iCMM Appraisal Method Version 1.0 A-19, INCOSE Conference CD, June 1999, Brighton UK [FAA98]

This definition differs from Planguage in that we are primarily concerned with design aspects, and this contains three requirement notions.

*architecture The organizational structure of a system or component.*

*Source: [IEEE 90] in [SEI-95-MM-003]*

Standard: An example of an IEEE definition of ‘architecture’.

*.*

*Domain: systems engineering.specification.design.architecture*

*Related Concepts [Architecture, \*192 collective noun]:*

*• Design (noun) \*047*

*• Design specification (\*586, or \*047 + \*137)*

*• Design Ideas \*047*

*• Architecture (the process) \*499*

*• Architecture Specification, \*617*

*• Artifact \*645*

*• Systems Architecture \*564*

*Keyed Icon* **\*192 Architecture: Δ (delta, symbol pyramid architecture)**

**Note keyed and drawn icon for design ( a subset of architecture is a rectangle (or [Design X] )which is analogous to the blocks used to make the pyramid)**

*Type: engineering specification type*

## Architecture Engineering Concept \*499 May 29, 2003

The architecture engineering process puts in place the systems architecture, which is a controlling mechanism for the design engineering of any project.

Architecture engineering defines the strategic framework (the systems architecture), which design engineering has to work within. It lays down the standards, which control such matters as the tradeoff processes amongst requirements. It helps synchronize design engineering disciplines across different systems.

The architecture engineering process is a *subset* of the Systems Engineering process.

Notes:

1. The architecture engineering process is distinct from the larger systems engineering process in that it is focused on *design issues*. (Systems engineering is broader. It includes consideration of the requirements, quality control, project management, and any other discipline, that is useful for satisfying requirements.)

2. The architecture engineering process is distinct from the other system level design engineering processes because it operates at a higher level, and is therefore concerned with wider issues. It has to consider the overall strategic framework, and provide guidance to all the lower-level systems.

The architecture engineering process considers especially the long-term objectives, and the totality of the requirements for all systems.

3. The architecture engineering process is, ideally, technologically neutral. It should provide guidance on design, using any relevant technology, policy, motivation, organizational idea, contractual agreement, sales practice and other devices. One of the main criteria is that the architecture is cost-effective.

Technological neutrality is not always achieved! For example, promotion of the use of standard platforms could be included within a systems architecture; and while that is an architectural decision, it is not technologically neutral.

Synonyms [Architecture Engineering \*499]:

• Architecting \*499

Related Concepts [Architecture Engineering \*499]:

• Systems Architecture \*564

• Requirement Engineering \*614

• Design Engineering \*501

• Architecture Specification \*617

Type [Architecture Engineering \*499]: Process.

## Architectural Description [IEEE] Concept \*618 July 18, 2003

Architectural description is “a collection of products to *document* an architecture.”

(This definition is identical with IEEE Draft Standard 1471, December 1999)

This concept is generic and can apply to any specific architecture type.

Notes:

1. The intentionally broad term ‘products’ is used to include *anything,* which might be useful in describing an architecture. *Anything* can include physical models, computerized models, prototypes, blueprints, parts lists, planned test results, actual input and outputs from tests, Planguage architecture specifications, sales and training materials, and real systems – as long as their *purpose* is to document an architecture.

2. The term ‘Architecture Description’ is an IEEE term, it is NOT used in the Planguage sense of a ‘Description’ parameter: it should really be equated to the Planguage term, ‘Definition.’)

Related Concepts [Architectural Description \*618]:

• Architecture Specification \*617: This concept does not include models and real systems, but only abstract specifications

• Systems Architecture \*564: An architecture description can be for any specialized subset of a systems architecture, such as software or hydraulics.

• Architecture (collective noun) \*192: this is the real set of artifacts that the architectural description describes.

Type [Architectural Description \*618]: Specification.

## Architecture Specification Concept \*617 June 17, 2003

An architecture specification is the written definition of an architectural component.

Notes:

1. An architecture specification either specifies a component of a systems architecture, or it specifies an architectural component of a specific system.

2. An architecture specification is a specialized form of design specification.

3. Architecture (the collective noun) is the *real* set of artifacts that the architecture specification describes. In other words, this is the *observable* architecture in a defined system. The specification may be describing *desired* *future* states of that system. Some parts of that specification might *never be implemented* in practice, since it serves as a vehicle to discuss architectural possibilities and options.

4. Architecture Description: This can be broader than a specification, and include models, prototypes, and real systems to aid in describing a real or projected architecture.

(Note: the term ‘Architecture Description’ is an IEEE term, it is NOT used in the Planguage sense of a ‘Description’ parameter: it should really be equated to the Planguage term, ‘Definition.’)

Synonyms [Architecture Specification \*617]:

• Architectural Specification \*617

Related Concepts [Architecture Specification \*617]:

• Architecture (collective noun) \*192

• Architecture Engineering \*499

• Systems Architecture \*564

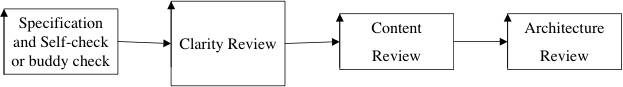
• Architecture Description \*618

Type [Architectural Specification \*617]: Specification.

## Architecture Review: Concept \*596 August 15, 2002

An architecture review is a review process which looks at the *entire* set of engineering artifacts for a given system at a given level of process or development.

It is not limited to the design architecture, but may choose to look at requirements, the architecture of a sub-system, the software architecture, the test planning. But the point is that it looks at a complete set of specified ideas which apply to a defined system.

Fig. \*596 The Architecture review is one type of content review. It can choose many review focus ideas, for example ‘End Life Disposal’, for a variety of purposes.

Related Concepts [Architecture Review, \*596]

• Specification Quality Control, \*051 this just checks rules for clear specification. Is the requirement or design clear, complete, consistent and has necessary background information?

• Specification Concept Review \*542 this just checks that a single artifact is specified according to applicable Content Rules \*543. Is it a good design or good requirement?

Spec \*137 -> Clarity Review \*607 -> Content Review \*542 -> AR \*596 -> next process

## Design Constraint Concept \*181 May 5, 2003

A design constraint is an explicit and direct restriction regarding the choice of design ideas. It either declares a design idea to be compulsory (Mandatory Design) or to be excluded (Prohibited Design).

Design constraints are dictated from an earlier system development stage (a higher level or a more specialized level). For example, the system architects pass on a number of design constraints, within the architecture specifications, to the system engineers.

A design constraint is a binary requirement. It can be a generic constraint or involve specific design(s).

Examples:

================ Prohibited Designs ==========

P1: Constraint: Products and services of direct competitors shall be avoided.

P2: Constraint: No software product version shall be released for sale until at least 3 month field trial has completed reporting no major faults outstanding. <- Technical Director’s Policy 6.9.

P3: Constraint [Europe]: No goods will be shipped without advance payment or bank guarantee.

============= Mandatory Designs ==============

M1: Constraint: Resident workers in country of export shall be used wherever possible.

M2: Constraint [IT Projects, In House]: Commercial Off The Shelf Software shall be used exclusively.

M3: Constraint: Products and Services from Our Corporation, Our Customers and Partners are preferred. <- Corporate Policy 5.4.

M4: Constraint [Programming]: Use Java as Programming Language.

Notes:

1. Some people use the term ‘Design Constraint’ to mean anything that constrains the choice of design. However, within Planguage, the term is more restricted. It is a direct constraint on design ideas themselves; directly referring to design ideas, generically or specifically. All other types of requirements ‘constrain’ our choice of design, but not as directly as a design constraint.

*Indirect Constraints:*

*• A Resource Constraint determines resource, and so impacts optional design.*

*• A Performance Constraint determines performance, and so impacts optional design.*

*• A Function Constraint determines function, and so impacts optional design.*

*• A Condition Constraint determines conditions, and so impacts optional design.*

*Direct Constraints:*

*• Design Constraints determine design directly, by specifying a mandatory design or a prohibited design.*

All requirement types: targets and constraints - have some potential effect on our design choices. But, design constraints are ‘direct’ in the sense that they make specific design decisions.

Example:

Spruce Goose:

Type: Generic Design Constraint.

Definition [If Wartime]: A troop transport plane may not use scarce <metal alloys>.

Howard Hughes’ airplane, ‘The Spruce Goose’, had this design constraint before the end of the Second World War. He made the plane largely of ‘spruce’ wood.

2. Only designs that are ‘design constraints’ should be allowed within *requirement* specifications. All *optional* design ideas, designs you can swap out if you find a better one, should be specified in *design* specifications. This is so that each level of design responsibility knows what it is free to do, and not free to do.

Synonyms [Design Constraint \*181]:

• Architectural Constraint \*181

• Design Restriction \*181

• Constrained Design: Informal Use

• Required Design: Informal Use

• Solution Constraint: Informal Use

Related Concepts [Design Constraint \*181]:

• Constraint \*218

• Requirement \*026

• Design Idea \*047

• Condition Constraint \*498

Keyed Icon [Design Constraint \*181]:

[<Design Idea Tag>] “The same icon as for a design. Alludes to a rectangle. [\_\_\_\_].”

[Editor Note to Publisher, the underline MUST be included in the above term with Design Idea tag. “Design Idea Tag” IS ALSO A USER DEFINED TERM, LIKE OTHER UNDERLINED TERMS]

Drawn Icon [Design Constraint \*181]: A rectangle within square brackets.



Type [Design Constraint \*181]: Constraint.

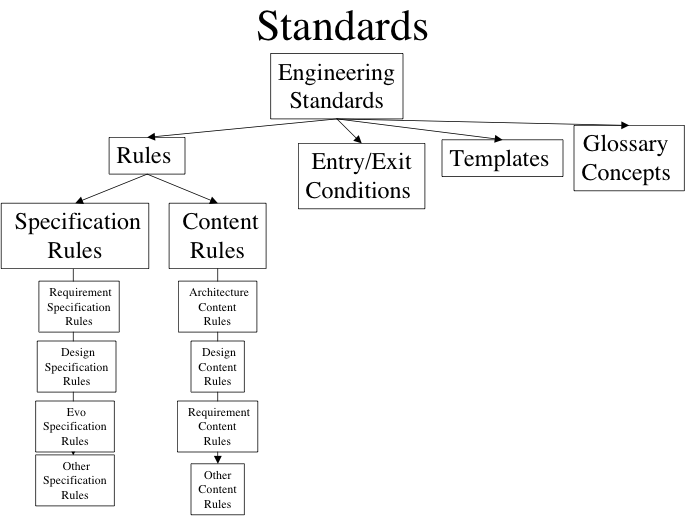
## Design Content Rules \*593 July 20, 2002

Design rules state expected design practices.

Design (content) rules give advice on best practices, and recommended practices, when designing (or architecting or design engineering, or finding means or strategies). Deviation from design rules should be the basis for a design review.

Unjustified violation of design rules are classified as design *defects*. These should not be confused with design specification defects, which are about how clearly a design is written up – not about its content or essence.

The density of design defects should be the basis for quantification of design quality. It should also be the basis for Entry to other processes of a design spec, and Exit from design processes of a design specification.



Related Concepts [Design Rules, \*593]

• clarity rules [Design Idea] \*129 rules about how articulate a design idea

• design specification \*586 – the spec which needs to be both written and judged according to the design rules.

• review criteria \*541 – Design Rules are one class of Review criteria

## Design Engineering Concept \*501 July 18, 2003

Design Engineering is an iterative process of determining a set of designs, with rigorous attention to quantified and measurable control of their impact on requirements.

The design engineering process implies the matching of potential and specified design ideas with quantified performance requirements, quantified resource requirements, and defined design and condition constraints.

Notes:

1. Planguage involves design engineering. By contrast, conventional ‘design’ activity (the kind of ‘design’ often found in the literature and in practice), usually has a less systematic, less quantified process, using perhaps intuition, tradition, and more trial and error, to determine satisfactory technology, and to determine stakeholder satisfaction. It is characterized by naming objectives (for example, ‘better usability’), and naming designs (for example, ‘single standard interface’), but not following up with *quantified* versions (that is, providing the information captured in Planguage, using such parameters as Scale, Goal, Impact Estimate and Meter).

2. The design *engineering* process can be subdivided into the following sub-processes:

a. *design requirements entry*: all relevant requirements to the design task are complete, clear, quantified, testable, validated by stakeholders, quality controlled, reviewed, and have thus obtained formal entry into the design process.

* 1. *search and acquire candidate design ideas* and their expected performance and cost attributes, along with any constraint threatening information. The search is based on matching design attributes with requirements, roughly
  2. *pick a reasonable set of design ideas*: this is done using Impact Estimation type of logic. It is done with regard to design attribute credibility, with regard to design attribute uncertainty, and with regard to necessary safety factor. When necessary, promising but uncertain, designs might need to be analyzed in depth using experiments, research, evolutionary implementation, and other devices to get more certain data about their attributes.
  3. *Deliver designs evolutionarily*: the candidate set of design ideas can then be delivered in an evolutionary sequence, one or more at a step, in priority sequence (high value to cost first) measuring actual results of performance and costs, until targets are reached, or budgets exceeded. Superfluous design ideas, in terms of reaching targets within constraints, can be kept in reserve for meeting improved target levels. But they do not necessarily have to be implemented when agreed targets are in fact reached.
  4. *Post delivery design improvement*: after a design is successfully implemented and delivered to some stakeholders, there is still the possibility that it might be replaced later by a later emerging superior (value to cost) design. This can happen before major project completion, or after initial system or project launch.

Related Concepts [Design Engineering \*501]:

• Design Process \*046

• Architecture Engineering \*499

• Systems Engineering \*223

• Engineering \*224

Type [Design Engineering \*501]: Discipline [Systems Engineering].

## Design Idea Concept \*047 July 23 2008

A design idea is a concept that is intended to satisfy some requirements.

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

Notes:

1. A *design specification* is a *written definition* of a specific design idea. A template for design specification is given in Figure 7.6.

2. A design idea is not usually a requirement. A design idea can, in principle, be changed at any time for a ‘better’ design idea (without having to ask the permission of any stakeholders because the system designers are responsible for the proposed design ideas). A ‘better’ design meets the requirements by giving more performance and/or less cost.

Requirements are inputs into a design process; design ideas are the outputs.

Note: A design can be a requirement if it is a *design constraint*. That is, a specific design is stated as mandatory or prohibited in the requirements.

3. A satisfactory design idea can have some negative performance scalar impacts, and still be acceptable overall. As long as the negative impacts (negative side effects) of a design idea do not prevent us from reaching all the required Goals levels, the design idea can be used.

4. <A design may be described in terms of its attributes. Design attributes can be function (and sub-functions). Performance attributes, costs, technical specification.

Synonyms [Design Idea \*047]:

• Design \*047

• Strategy \*047

• Proposed Solution \*047

• Means \*047

Related Concepts [Design Idea \*047]:

• Architecture \*192

• Policy \*111

• Design Constraint \*181

• Design Specification \*586

• Design Target \*338

• Design Problem \*654

• Entity \*645

• Function Design \*521

Keyed Icon [Design Idea \*047]: |\_\_\_\_\_\_\_\_|

S1 S2

|\_\_\_\_\_||\_\_\_\_\_\_\_|

O-------[---🡨------🡪---]----🡪-----🡪 P

M Vb B T1 Vb T2

Illustration for paper ‘Vision Engineering’ (July 2008). M = Mission, Vb = Core Value barrier – constraint. B = Benchmark level. T1 and T2 are Target levels of Performance attribute P (a Core Purpose). S1 and S2 are Strategies. Based on ideas in Collins and Porras, “Built To Last”.

Graphic Icon [Design Idea \*047]: A lying-down rectangle. (The standing rectangle is a document icon.)



Type [Design Idea \*047]: Parameter, Design.

## Design Problem: (Concept \*654) November 13 2005

A design problem is expressed as a requirement specification that includes a ‘complete set’ of valid targets and constraints, together with updated information about the degree of satisfaction in the design or the implementation. A design problem is the ‘set of gaps’ (to meet required levels) that remain to be designed for.

Note 1. We assume that the designer does not yet contain or have access to sufficient design solutions to reach all their targets, within all specified constraints, otherwise there would be no ‘problem’ remaining.

Note 2. Under Evolutionary Project Management, the design problem will evolve as the project progresses. Each new design idea, when implemented at an Evo step, will have a set of cumulative impacts on different requirements. The then remaining set of unsatisfied requirements (‘gaps’) become the new ‘design problem’.

Note 3.

On the notion of a ‘complete set’ of requirements in problem definition.

A simplified design problem could be formulated using one or a few requirements or constraints, just to see if that were technically or economically feasible at all. But the real design problem must be solved using all valid requirements. If this is impossible, then perhaps some of the requirements need to be realistically ‘relaxed’ to allow some of the others to be satisfied at all.

Short form: Problem.

Keyed icon [Design Problem, \*048]: -----<===>---> (The gap between < a past and > a target is an iconic symbol of a design problem)

O------------<-----------|=====The Gap===>------->

The progress from the Benchmark (<) to the current level (|) leaves a Gap (==) to the Target (>). This illustration brings in the idea that there has been some progress (‘<-------|’) since initial requirements were specified (‘-<-‘) , in delivering some degree of requirements target satisfaction using some design ideas. The gap (‘|====>” represents the remaining problem with this particular dimension of a requirement.

Synonyms: Design Target, residual system requirements, design gap, gap \*359 (a single instance of a gap amongst many which constitute the total design problem).

Related Concepts [Design Problem \*654 ]:

• Control Point \*346

• Problem \*270

Type [Design Problem \*654 ]:Engineering Concept.

## Design Process Concept \*046 July 18, 2003Btg

The design process is the act of searching for, specifying, evaluating and selecting design ideas, in an attempt to satisfy specified stakeholder requirements.

Design is finding a set of solutions (design ideas) for a set of defined requirements.

Overview of the Design Process

• Analyze the Requirements

• Find & Specify Design Ideas

• Evaluate the Design Ideas

• Select Design Ideas & Produce Evo Plan

Design can be carried out in several ways. It can be based on tradition, on intuition, on dogma, on principles or heuristics. It can also be based on multidimensional quantified logic – this latter we would call ‘engineering’ or ‘systems engineering’.

“Design comes about entirely from the playing out of the evolutionary algorithm.”

Blackmore in ‘The Meme Machine’ (See page 205)

Notes:

1. The design process can be subdivided into the following sub-processes:

a. Acquire stakeholder requirements.

b. Identify candidate designs

c. Tailor candidate designs to current purpose

d. Decide on a finite set of designs.

e. Do design review.

f. Implement designs in development system and test their validity.

g. Revise Design if necessary.

h. Implement designs in final system.

“Heuristic: At some point in the project, freeze the design. …..

This rule of thumb recognizes that a point is often reached in design where the character of a project, and hence the appropriate allocation of resources, changes from seeking alternative solutions to perfecting a chosen solution. …. After this point is reached, a major design change runs the unacceptable risk of introducing a fatal flaw because insufficient resources remain to evaluate all of its ramifications.” Source: Koen03, p.35

This quotation refers to b. (Identify) and c. (Tailor) above. It reminds us that the design process is not merely one of finding a design, and analyzing it. The selected design is likely to be generic in nature, or was applied in a different set of circumstances than the current project. So, we must expect to ‘perfect the chosen solution’ so that it better fits the current requirements and environment.

2. The reader might like to compare this simple design process with the corresponding design *engineering* process (\*501 note 2). The design engineering process is characterized by rigorous use of quantified information. It takes a clearer position on ‘how’ to design. The design process decomposition above specifies ‘what’ is to be done, but does not specify ‘how’.

Related Concepts [Design Process \*046]:

• Design Engineering \*501

• Systems Engineering \*223

• Engineering \*224

Type [Design Process \*046]: Process.

## Design Specification Concept \*586 May 29, 2003B

A design specification is the written specification of a design idea.

A set of design specifications is the main output of a design engineering process.

A specific set of design specifications, when implemented, will, to some degree, meet the stated requirements.

Notes:

1. A set of design specifications attempts to solve a design problem. Identification and documentation of the individual design ideas, and their potential contribution towards meeting the requirements, helps selection of the ‘best’ design ideas for implementation.

2. The design engineering process uses the requirement specification as input. The design engineering process output is a set of design (solution) specifications (of design ideas). These design specifications might also contain information about their expected attributes for meeting requirements. This ‘expected attributes’ information of a design specification might be in the form of an Impact Estimation table or, it can be as simple as an assertion of impacts on requirements, referenced by their tags (see example below).

Example:

Engineer Motivation:

Gist: Motivate, using free time off.

Type: Design Idea.

Impacts [Objectives]: {Engineering Productivity, Engineering Costs}.

Impacts [Costs]: {Staff Costs, Available Engineering Hours}.

Definition: Offer all Engineers up to 20% of their Normal Working Hours per year as discretionary time off to invest in Health, Family and Knowledge (Studies, Write Papers, Go to Conferences).

Source: Productivity Committee Report 1.4.3.

Implementor: Human Resources Director.

Template [Design Specification \*586]: Design Specification Template.

Abbreviations [Design Specification \*586]:

• Design Spec \*586

Synonyms [Design Specification \*586]:

• Technical Design: Informal use

• ‘The Design’: Informal use

Related Concepts [Design Specification, \*586]:

• Design Engineering Process \*501

• Design Idea \*047

• Systems Architecture \*564

• Architecture \*192

• Architecture Specification \*617

• Specification \*137

Keyed Icon [Design Specification \*586]: []->@

This icon can be interpreted as a specification, ‘[]’ which impacts, ‘->’ target, ‘@’.

Type [Design Specification \*586]: Parameter, Specification.

## Design Target: Concept \*338 March 15, 2003

A Design Target is the specific target *level,* including *qualifier,* of an attribute requirement that is being worked towards, in the design or implementation process.

Usage 1. The designer uses a set of such design goals, including budget targets, as well as constraint information, to decide if they have found good enough design ideas to satisfy the performance requirements, within the constraints specified.

Keyed icon [Design Target, \*338 ] : ----> <level spec> [<qualifiers>].

The target level plan (or other target symbols) symbol, coupled with the qualifier

Example:

O-------------------->6,000 Hours MTBF [Release 6.0, US Market]----->. Reliability

Example:

Fail [End Next Year] 30. “a design constraint”

Goal [Within 6 Months After Launch] 60. “a design target”

The target is not merely the ‘60’ but includes the [qualifier] information.

In addition, the Fail and Goal specification both give critical information about the priority of the target (design survival point (Fail) and success point (Goal)).

*Usage 2. It is normal engineering practice for a designer to apply a performance target level with a built in safety factor, as the level they try to design for; for example two times the nominal specified level might be applied as the safety factor target. The Impact Estimation process makes use of this safety factor concept.*

*Usage 3. A design target may be specified using one of the target specification parameters {Goal, Stretch, Wish}. This applies to performance and budget targets.*

*Example:*

*Stretch [Next Release] 66.5 hours.*

*Related Concepts [Design Target \*338]:*

*• Benchmark \*007*

*• Performance Target \*439*

*• Qualifier \*124*

*• Budget Target \*481*

*• Condition Constraints \*498*

• Design problem, \*048

• Scalar target \*470

Type [Design Target \*338]: Requirement Level, Target

## Design To Cost Concept \*472 July 17, 2003

‘Design To Cost’ is an engineering strategy of consciously selecting design ideas, which will allow you to stay within any and all resource budgets.

Related Concepts [Design To Cost \*472]:

• Design Process \*046

Type [Design To Cost \*472]: Design Strategy, Process.

## Designer Concept \*190 May 6, 2003

A designer is a person or group who specifies design ideas, in an effort to satisfy specified requirements.

Notes:

1. This is a generic term. More specific terms include systems engineer, planner, or systems architect.

Synonyms [Designer \*190]:

• Systems Designer \*190

Related Concepts [Designer \*190]:

• Systems Architect \*193

Type [Designer \*190]: Role.

## Systecture © Concept \*564 May 27, 2003

See Systems Architecture \*564. Systecture is a conjunction of the term ‘system architecture’.

Historical Background [Systecture \*564]: In July 2002, in connection with a book manuscript on systems architecture, I needed a catchy term for the book title. In my 1988 book, ‘Principles of Software Engineering Management,’ I had coined the terms ‘softecture’ and ‘softect’. So, it seemed natural to extend this to the system engineering area. A web search turned up [www.systect.com](http://www.systect.com) (Systect, Inc. ‘The system architects’) a systems architecture company, but no use of Systecture at all. © [Tom@Gilb.com](mailto:Tom@Gilb.com) 2002. Permission is granted to use the term as a generic word. I felt there was a need to get away from the ‘architecture’ term .

Architect is from ‘Archi-Tecton,’ which means ‘Master Builder’. ‘Archi’ is not from ‘Arch’, but from ‘Arche’: primitive, original, primary[[1]](#footnote-0).

## Systect: Concept \*565. July 19, 2002

A systect is a person who does Systecture (systems architecture) – a systems architect. It is a conjunction (systems architect).

Related Concepts [Systect, \*565]

• Systecture \*564 systects do systecture

• architect \*193 systects are systems architects

• designer \*190 all systects are designers, but not all designers are systects

Background: See background for Systecture above.

A web search turned up [www.systect.com](http://www.systect.com) (Systect, Inc. ‘The system architects’) a systems architecture company.

## Systems Architect Concept \*193 July 23, 2003

A systems architect is a person or group, who carries out the work tasks of systems architecture (a process).

Notes:

1. The systems architect, the agent for the systems architecture process, is the technical interface between the customer and all the technological means for helping the *stakeholders* reach their objectives.

2. The systems architect manages all other necessary technologies, and is empowered to choose the best alternative technology areas to solve ‘the problem.’

3. The systems architect must also ensure suitable interfaces between the technologies that must be integrated.

4. The specifications of the systems architect will have higher priority than design ideas originating at lower levels of systems engineering specification.

5. The role of systems architect might be carried out by an individual, or it could be a series of people through time, or it might be a collective responsibility. It can be a part-time responsibility.

6. Architect is from ‘Archi-Tecton,’ which means ‘Master Builder’. ‘Archi’ is not from ‘Arch’, but from ‘Arche’: primitive, original, primary[[2]](#footnote-1).

I think architecture is the science of structure and the structure of whatever is, whether it is music, whether it is painting or building or city planning or statesmanship’

Frank Lloyd Wright, An Autobiography, Horizon Press, NY, 1977 (See page 131)[[3]](#footnote-2)

Related Concepts [Systems Architect \*193]:

• Systems Architecture \*564

• Designer \*190

Type [Systems Architect \*193]: Role.

## Systems Architecture Concept \*564 July 23, 2003

Systems Architecture is the set of artifacts produced by Architecture Engineering. A systems architecture is a strategic framework and consists of models, standards and design constraints specifying mandatory and recommended best practice for implementing and maintaining systems.

Notes:

1. A systems architecture usually applies across a division or an entire organization.

2. A systems architecture varies in its level of detail depending on its maturity and what is required of it. Different organizational cultures will require different things. The main point is that a systems architecture should be cost-effective.

3. The aims of a systems architecture could include:

• imparting technical strategy

• sharing best practices

• ensuring specific standards are adhered to (for example, security)

• avoiding duplication of effort

• reducing risk by promoting tried and trusted information

• encouraging recognition and use of standard interfaces

• promoting reuse

• ensuring compatibility of data structures amongst systems

• achieving economies of scale through standard platforms (especially for training, support and maintenance)

4. Individual systems will have their own architecture (Architecture \*192), which will adhere to any relevant mandatory systems architecture.

Synonyms [Systems Architecture \*564]:

• Systecture \*564

• Architecture \*192

Related Concepts [Systems Architecture \*564]:

• Architecture Engineering \*499

Type [Systems Architecture \*564]: Architecture.

<http://www.sei.cmu.edu/architecture/community_definitions.html>

is a link with a great many opinions about the meaning of software architecture. Nov 22 2007 tg

## Systems Engineer: Concept •574 July 12, 2002

One who practices systems engineering (\*233) or is qualified to do so.

## Systems Engineering Concept \*223 July 17, 2003

Systems Engineering (SE) is an engineering process, encompassing and managing all relevant system stakeholders requirements; as well as all design solutions; and necessary technology, economic, and political areas.

The fundamental purposes of systems engineering are to:

• optimize the system solution at the highest level of stakeholder concerns,

• synchronize all contributing disciplines to contribute efficiently to the final system characteristics,

• consider the entire system life cycle needs,

• manage risks for the entire system and the entire system life.

An INCOSE Definition

“Systems Engineering integrates all the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation. Systems Engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs.”

(http://www.incose.org/ whatis.html)

Blanchard’s Department of Defence (DoD) Definition

Systems Engineering is the “process that shall:

1. Transform operational needs and requirements into an integrated system design solution through concurrent consideration of all life-cycle needs (i.e., development, manufacturing, test and evaluation, verification, deployment, operations, support, training and disposal);

2. Ensure the compatibility, interoperability, and integration of all functional and physical interfaces and ensure that system definition and design reflect the requirements for all system elements (i.e., hardware, software, facilities, people, data); and

3. Characterize and manage technical risks.” [BLANCHARD98]

An FAA (the USA Federal Aviation Authority) Definition

Systems Engineering is:

“A hybrid methodology that combines policy, analysis, design, and management. It ensures that a complex man-made system or product, selected from the range of options available, is the one most likely to satisfy the customer’s objectives in the context of long-term future operation or market environments.

Systems engineering is applied throughout the system or product life cycle as a comprehensive, possibly iterative, interleaved, or recursive, technical process to:

a. Translate an operational need into a configured system or product meeting the operational need

b. Integrate the technical contributions of all available development resources, including all technical disciplines into a coordinated effort that meets established program cost, schedule and performance objectives. This involves a “holistic view” (the design of the whole as distinguished from the design of the parts). Such a view is multi-disciplinary in nature, rather than disciplinary or interdisciplinary;

c. Ensure the compatibility of all function and physical interfaces (internal and external)

d. Ensure that system or product definition and design reflect the requirements in system or product elements (outcome, hardware, software, facilities, people, and data).

e. Characterize [identify, define, and classify] technical risks, develop risk abatement approaches, and reduce technical risks by prevention and mitigation of impacts when risks are realized.”

Source: FAA-iCMM Appraisal Method Version 1.0 A-19, INCOSE Conference CD, June 1999,

Brighton UK [FAA98].

Notes:

1. The Systems Engineering process is a conscious attempt to avoid sub-optimal engineering. Without Systems Engineering, the success of the resulting system is more accidental than predictable. Systems engineering is necessary because there are so many possible places for product development to go wrong. For example, sub-optimal results might be caused by setting requirements for too narrow a list of stakeholders, or by using too narrow a set of design ideas to solve the problem of satisfying all project requirements. Another frequent problem, especially in well-established large companies, is for groups to produce optimal components yet produce a very sub-optimal complete system.

*2.* Systems engineering includes a broad application of disciplines such as requirements engineering, quality control, project management, test engineering, and any of the many other disciplines that might be found useful for satisfying stakeholders. Architecture Engineering, a subset of systems engineering, is by contrast, directed only towards the design aspects.

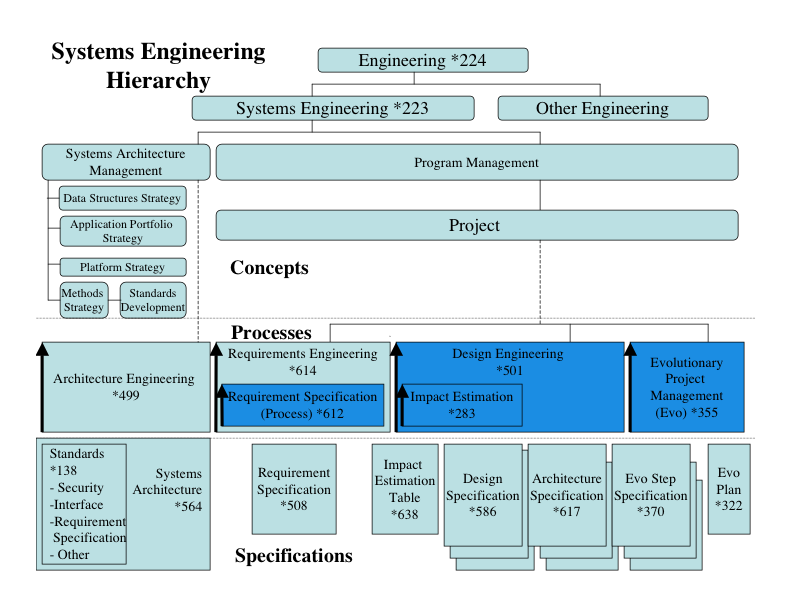


Figure \*223: [Editor Note to Publisher: This diagram had to be imported via PDF???]

Related Concepts [Systems Engineering \*223]:

• Engineering \*224

• Architecture Engineering \*499

• Requirement Engineering \*614

• Design Engineering \*501

• Evolutionary Project Management \*355

Type [Systems Engineering \*223]: Discipline [Engineering].

1. Contributed by Niels Malotaux. [↑](#footnote-ref-0)
2. Contributed by Niels Malotaux. [↑](#footnote-ref-1)
3. Cited page 306 Principles of Software Engineering Management ( so permission obtained for that) [↑](#footnote-ref-2)