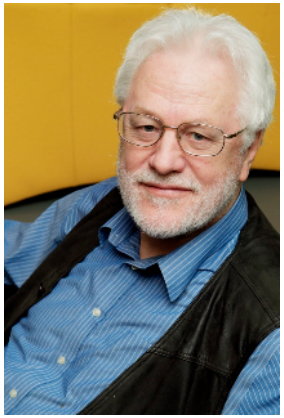
"***Real*** Architecture-Engineering: Engineering? or Pompous Bullshit?"



Tom Gilb

ACCU Conference, Bristol, UK
Saturday, April 13 2013. 11:30 to 13:00 (90 minutes)

2-3 Day Course Documentation
https://www.dropbox.com/sh/egy3scl9a52zuqt/R5HVAo0sCr
Free courses, Oslo May 7-8, London BCS August 28-30, NL April 25

The Architecture Manifesto

[Systems and Software]:
(advice for the revolution)

- 1. Architecture is the *servant* of the Priority Stakeholder Values.
- 2. The Architect is responsible for knowing, and specifying, all values and costs, long and short term of all architecture.
- 3. All architecture is suspected of having disappointments and surprises, until proven otherwise but real system measurement.
- 4. All architecture should be removable, if it fails, or if we get a much better idea.

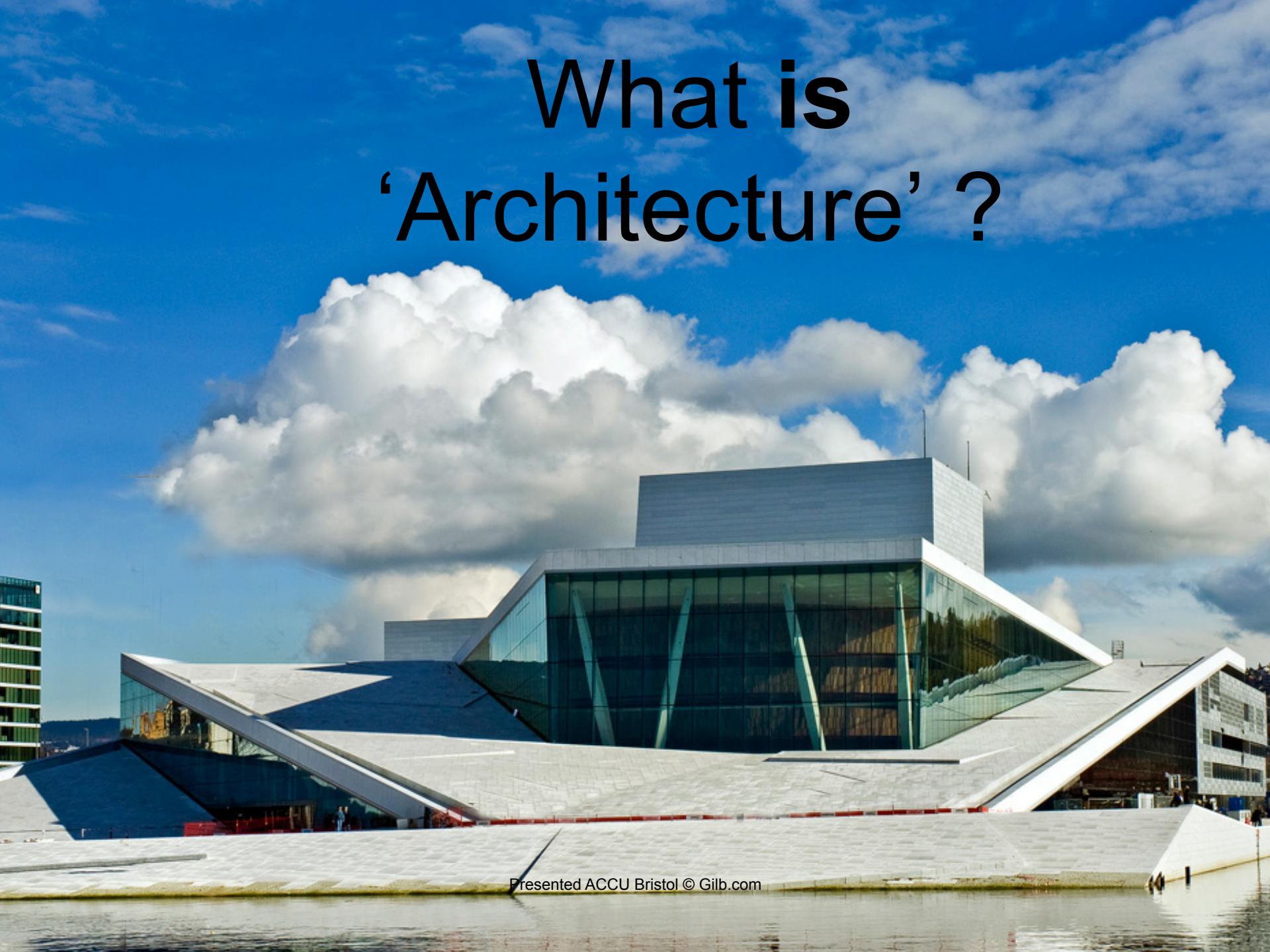
Conference Announcement

- What should software architecture be? How is it related to major critical software qualities and performance, to costs and constraints? How do we decide exactly what to propose, and how do we estimate and prove it is justified. How can an organization qualify their own architects, and know the difference between the frauds and the experts? Would real architects recognize what software architects know and do?
- We believe that most activity, going under the name architecture, is NOT real. Current Software architecture is no more real architecture than hackers are software engineers.
- If we are just informally throwing out nice ideas, let us call ourselves Software Brainstormers. But if we are dealing with large scale, serious, and critical systems, then we need to stop using cabin-building methods and start using skyscraper designing methods. We need a serious architecture and engineering approach.

- Summary:
- • defining architecture properly : even the standards are wrong
- • what is bad architecture
- • real architecture responsibilities who does what to whom
- • the technical disciplines we need; quantification, estimation measurement of multiple qualities and costs
- • architectural decomposition: a value basis
- • software design, the same process, a different level
- • the role of iterative feedback in verifying architecture
- • The Architecture Manifesto: (advice for the revolution

R U AN ARCHITECT ?

What **is** ‘Architecture’ ?



Architect = Master Builder

Architect is from ‘Archi-Tecton,’

which means ‘Master Builder’.

‘Archi’ is not from ‘Arch’,

but from ‘Arche’: primitive, original, primary.

Our *Personal Subjective* **Opinion** follows …

Kai Gilb and Tom Gilb

- I am happy to discuss with you here and via tom@gilb.com
- Or you can tweet your opinion at #ACCU2013!

The architecture is there to satisfy requirements

Oslo Opera house requirements

- Qualities
- Costs
- Constraints

Oslo Opera house requirements

- Qualities
  - Impressive
  - Acoustics
  - Flexibility
  - Extendibility
  - Integratedness
  - Performance Visibility
  - National Symbol
  - Access to Fjord View
  - Comfort
- Costs
  - Building
  - Maintenance
  - Operational manpower
- Constraints
  - Legal Building
  - National Architecture
  - Archeological Site
  - Local Materials
  - Local Labour

The architecture is there to satisfy requirements

Architecture
that never refers to
necessary qualities,
performance characteristics,
costs,
and constraints
Is not really architecture
Of any kind

The architecture is there to satisfy requirements

The Architecture *process* is driven by requirements

Real (IT/Sw) Architecture

Real Architecture

- Has multidimensional *clear* design performance objectives
- Has *clear* multiple constraints
- Produces architecture ideas which enable and permit objectives to be met reasonably within constraints
- Estimates expected effects

***Pseudo* Architecture**

- Lacks dedication to clear **objectives** and **constraints**
- Does not **estimate** or articulate the expected **effects, on** objectives & constraints, of suggestions

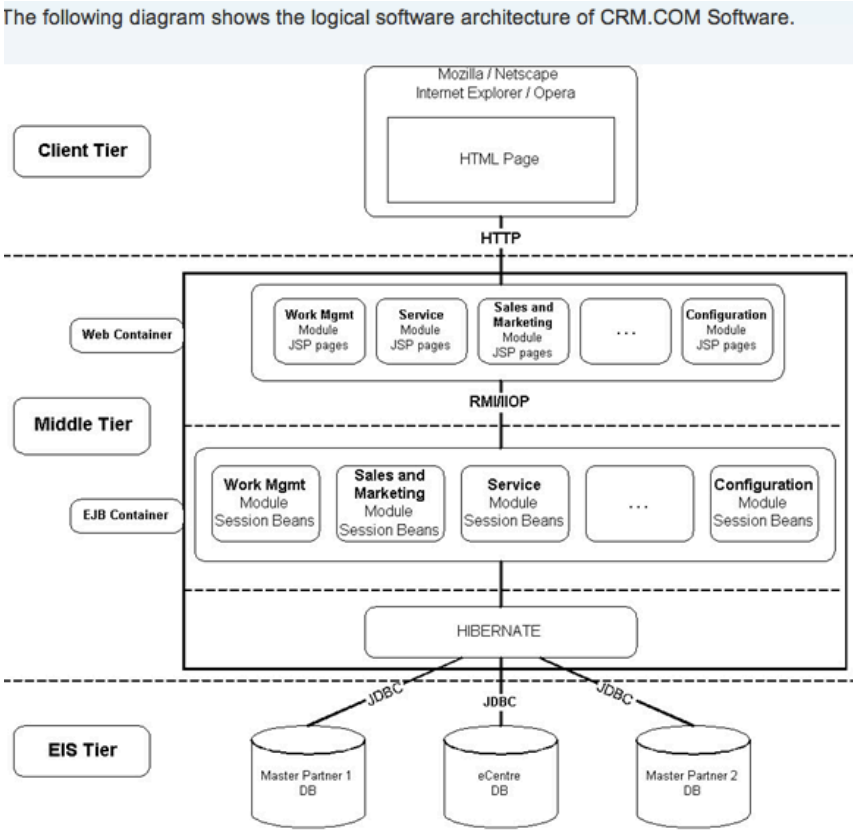
Pseudo Architecture
Does not mention goals and constraints

‘Bad’ ‘Arch.’ definitions

- Software architecture is a collection of software **components** unified via interfaces into decomposable system based on one or more technology platforms.
- Software Architecture shows the **structural** and **behaviour** of a system which is comprised of software **elements** and *exposing the properties* of those elements and relationships among them.

http://www.sei.cmu.edu/architecture/start/community.cfm

Uninformative diagrams



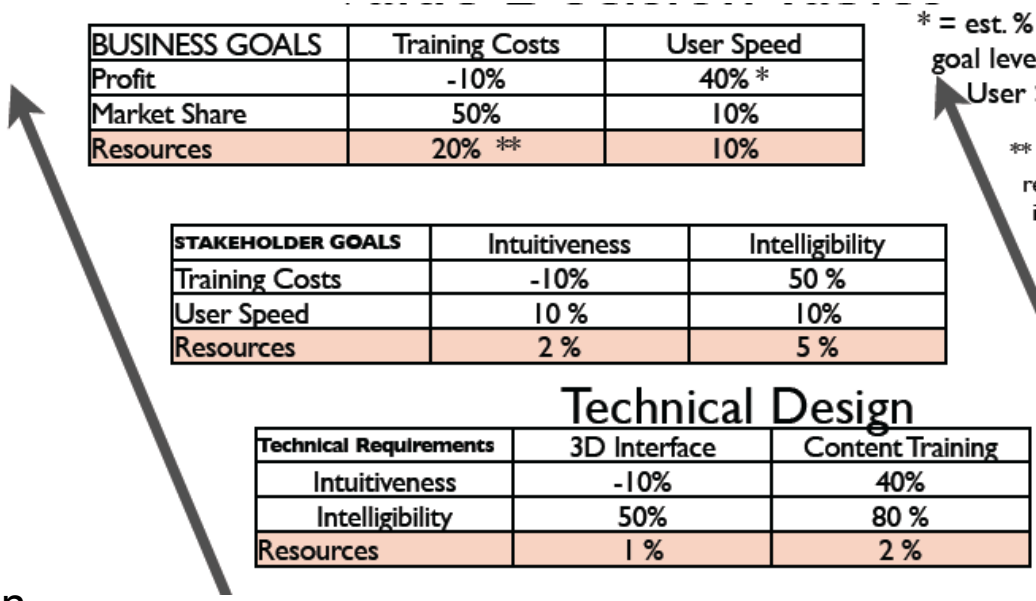
Better Architecture

Better definitions

- Software …needs to address the needs of business **stakeholders** within the organizational, technical and any other **constraints** to achieve the business, technical or any other **goals**.
  - It also needs to address software trustworthy characteristics like reliability, availability, maintainability, robustness, safety, security and survivability.

- System Architecture should contain **goals/requirements** artifacts, and structure and behavior artifacts **based on** those goals.

Real Architecture diagrams



http://www.sei.cmu.edu/architecture/start/community.cfm

A Distinction

Architecture *Process*

- A continuous, and lifecycle long, **activity of finding means for ends**

Architecture *Specification*

- A specification of
  - a set of means
  - for a set of ends

We argue that the following are **absolute essentials** for 'real' architecture

Architecture *Process* has

- Clear multiple objectives
- Clear constraints
- A process of identifying and analyzing (estimating effects of) potential means
  - For reaching objectives, within constraints

Architecture *Specification* has

- Well defined components
  - Able to deliver predictable attributes
- Credible estimates of the multiple effects of each component, and the whole

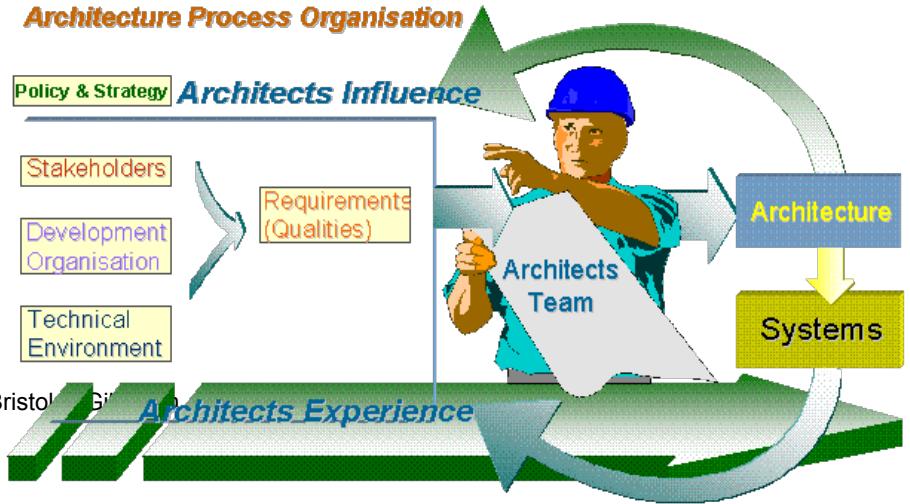
Why are these Architecture essentials, essential?

Why?

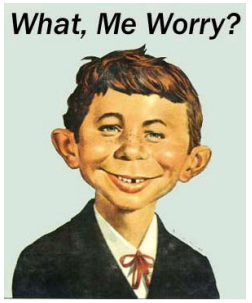
- Failure to reach even one ‘critical’ objective can mean total system failure
  - Example: reliability
- Failure to respect even a single constraint can mean total system failure
  - Example: cost

And if they are missing…

- You cannot expect the specified architecture will reach objectives, within constraints
- You have lost architectural control



What a Difference



A Real Architect

- Can and does estimate resources needed for any suggested architecture
  - Capital Cost
  - Maintenance Cost
  - Skilled People hours to install and maintain
- Can and Does estimate the impact of each architecture component on the top level critical objectives
  - All '-ilities' (security etc)
  - All Performance (Capacity

A False Architect

- Does not even try to **estimate any costs**
- of any architectures
  - Does **not know how** to do so if asked
  - If they try to estimate they are at least 10x wrong
- Does not **even try to estimate the numeric impact** on even the most critical architectural objectives
- Does not even **realize** they need **quantified performance and quality objectives** to drive and justify architecture
- They have no specific verifiable idea of the impact their ideas have on numeric quality and performance levels.
- It is all 'smoke and mirrors'
- They take **no responsibilit**y for the performance and quality attributes or costs of their suggested architecture: no skin in the game.

“Architecture Engineering”

A high level design process

- 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 (*499) is a *subset* of the Systems Engineering process (*233).

Engineering Hierarchy

Engineering *224

Systems Engineering *223

Other Engineering

Systecture (Systems Architecture) *564

- Data Structures Strategy
- Application Portfolio Strategy
- Platform Strategy
- Methods Strategy
- Standards Development

Program Management

- Project

Engineering Concepts

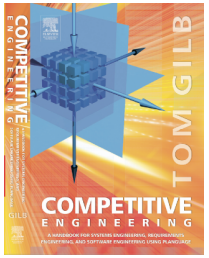
Processes

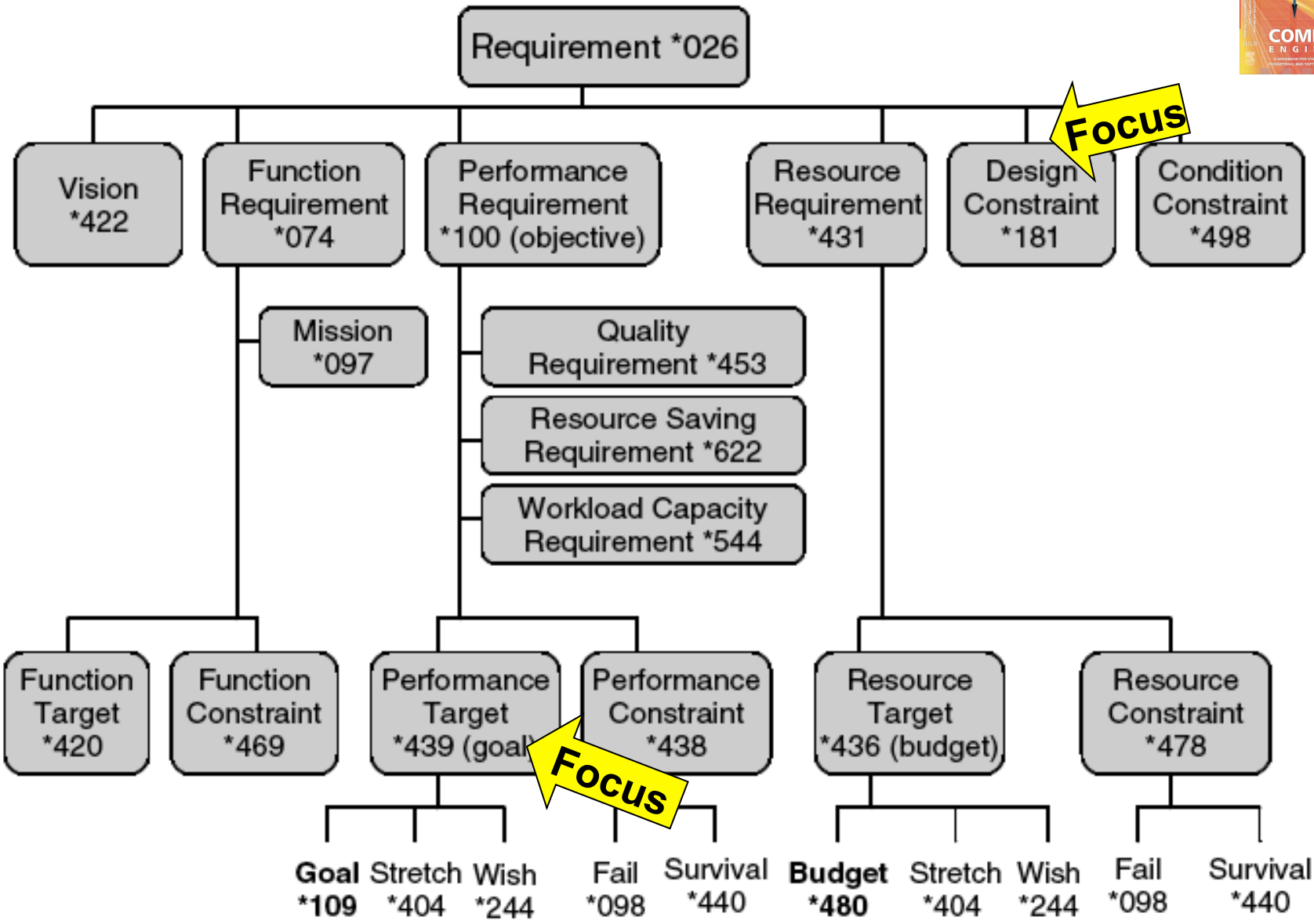
- Architecture Process *499
- Requirements Process *612
- Design Engineering *501
- Evolutionary Project Management (Evo) *355
  - Design Process *046
  - Impact Estimation *283

Specification Types

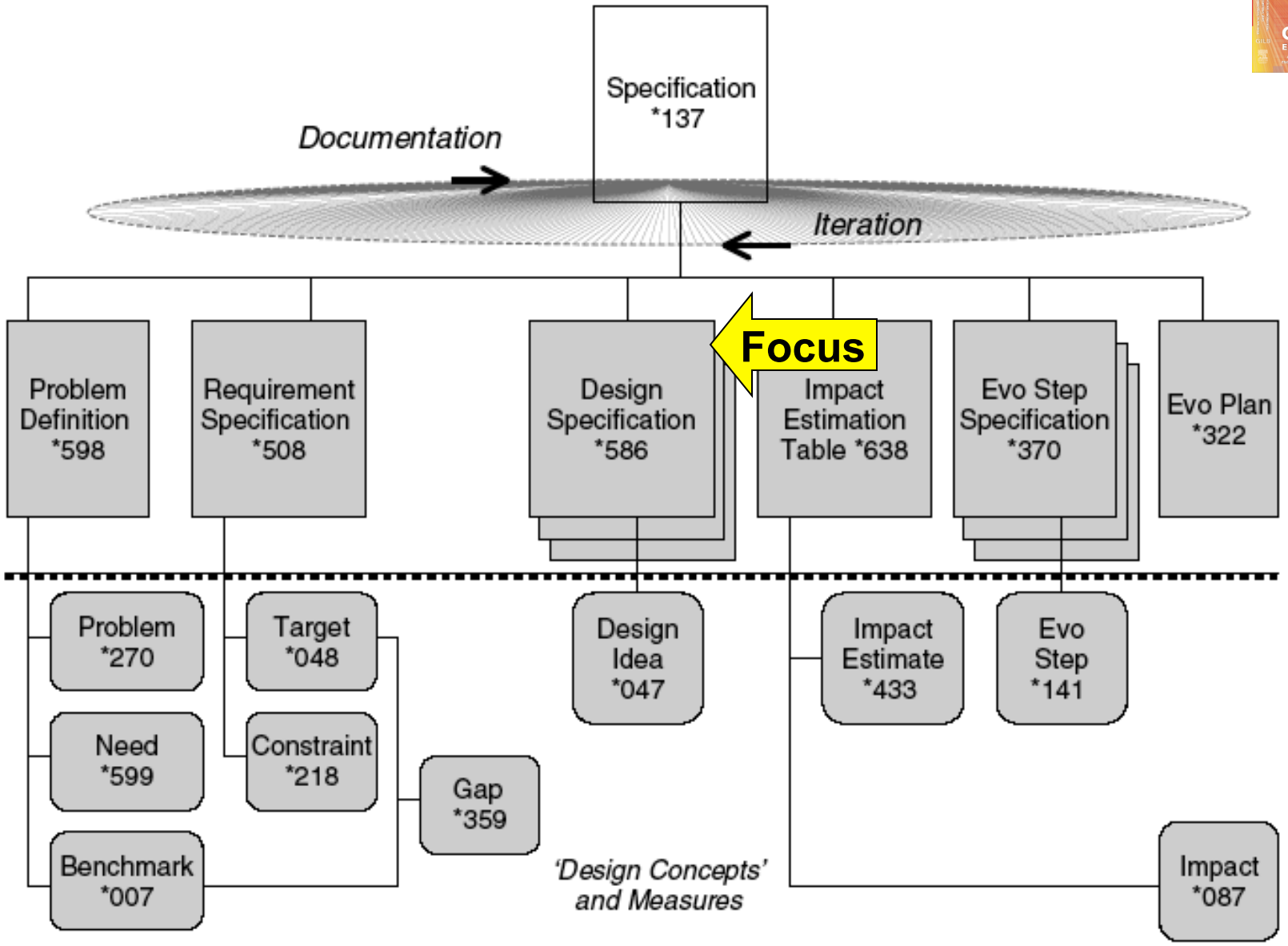
- (The) Architecture *192 (Artifacts)
- Architecture Specification *617
- Standards *138
  - Security Standards
  - Interface Standards
  - Requirement Specification Standards
  - Other
- Requirement Specification *508
- Design Specification *586
- Impact Estimation Table
- Evo Step Specification *370
- Evo Plan *322

Requirement Concepts for Architects

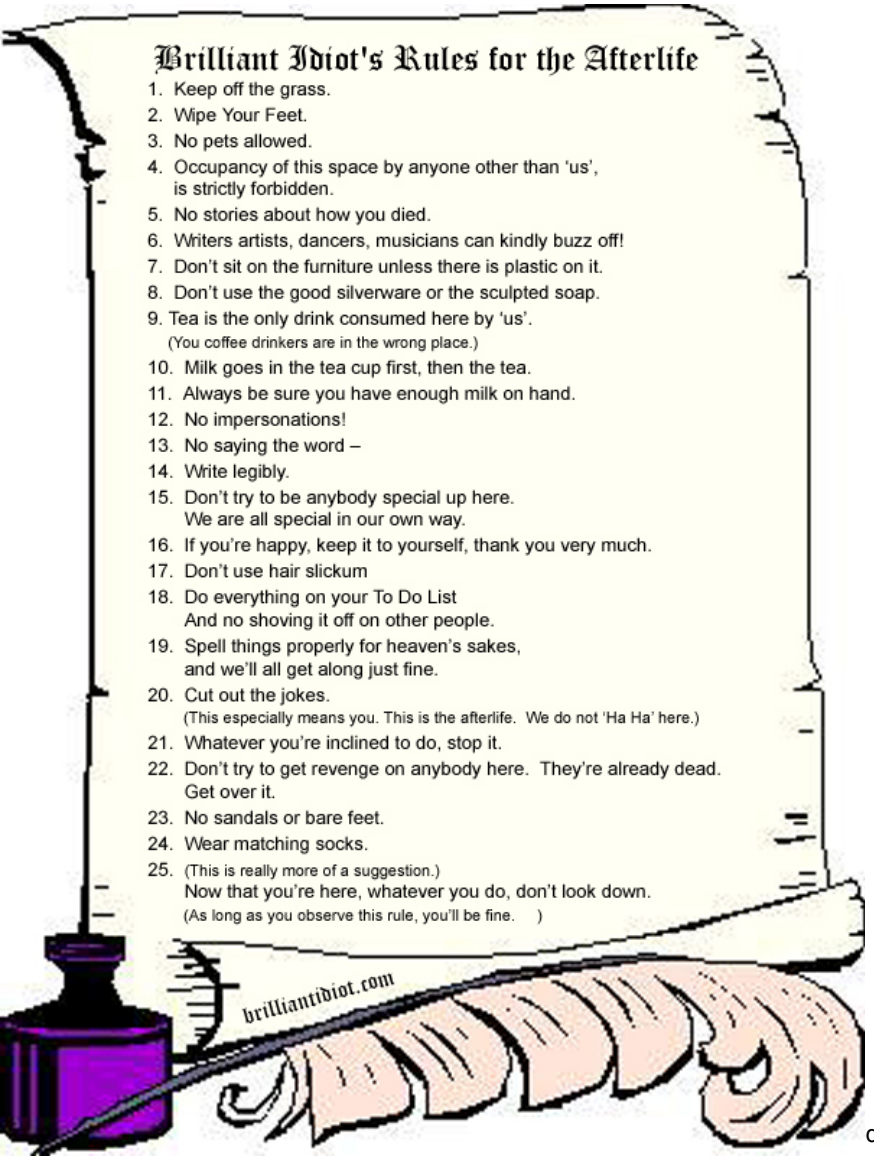


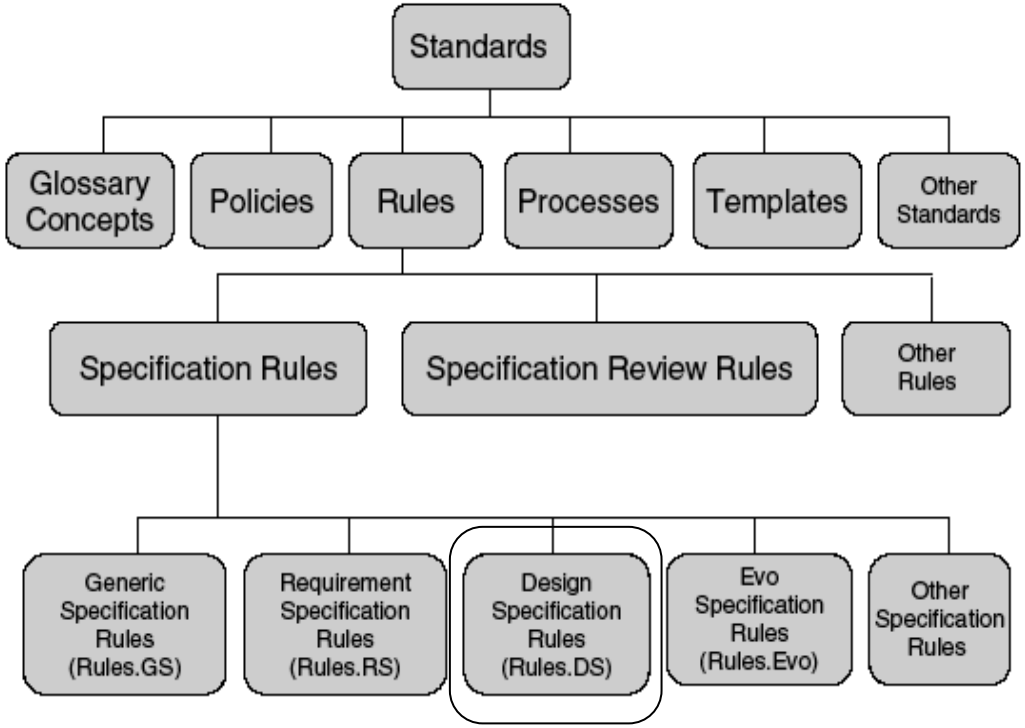


Specification Types for Architects

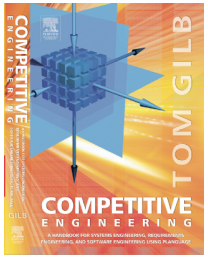


Specification Rule Types: useful for Architecture Processes and Specification





See next slide
For detailed example



Architecture Specification Rules from CE Book Ch. 7

7.4 Rules: Design Specification

(edited down for simplicity)

R1: Design Separation: Only design ideas that are intentionally 'constraints' (Type: Design Constraint) are specified in the requirements. Any other design ideas are specified separately (Type: Design Idea).

R2: Detail: A design specification should be specified in enough detail so that we know precisely what is expected, and do not, and cannot,

inadvertently assume or include design elements, which are not actually intended.

R3: Explode: Any design idea (Type: Complex Design Idea), whose impact on attributes can be better controlled by detailing it, should be broken down into a list of the tag names of its elementary and/or complex sub-design ideas.

R4: Dependencies: Any known dependencies for successful implementation of a design idea need to be specified explicitly.

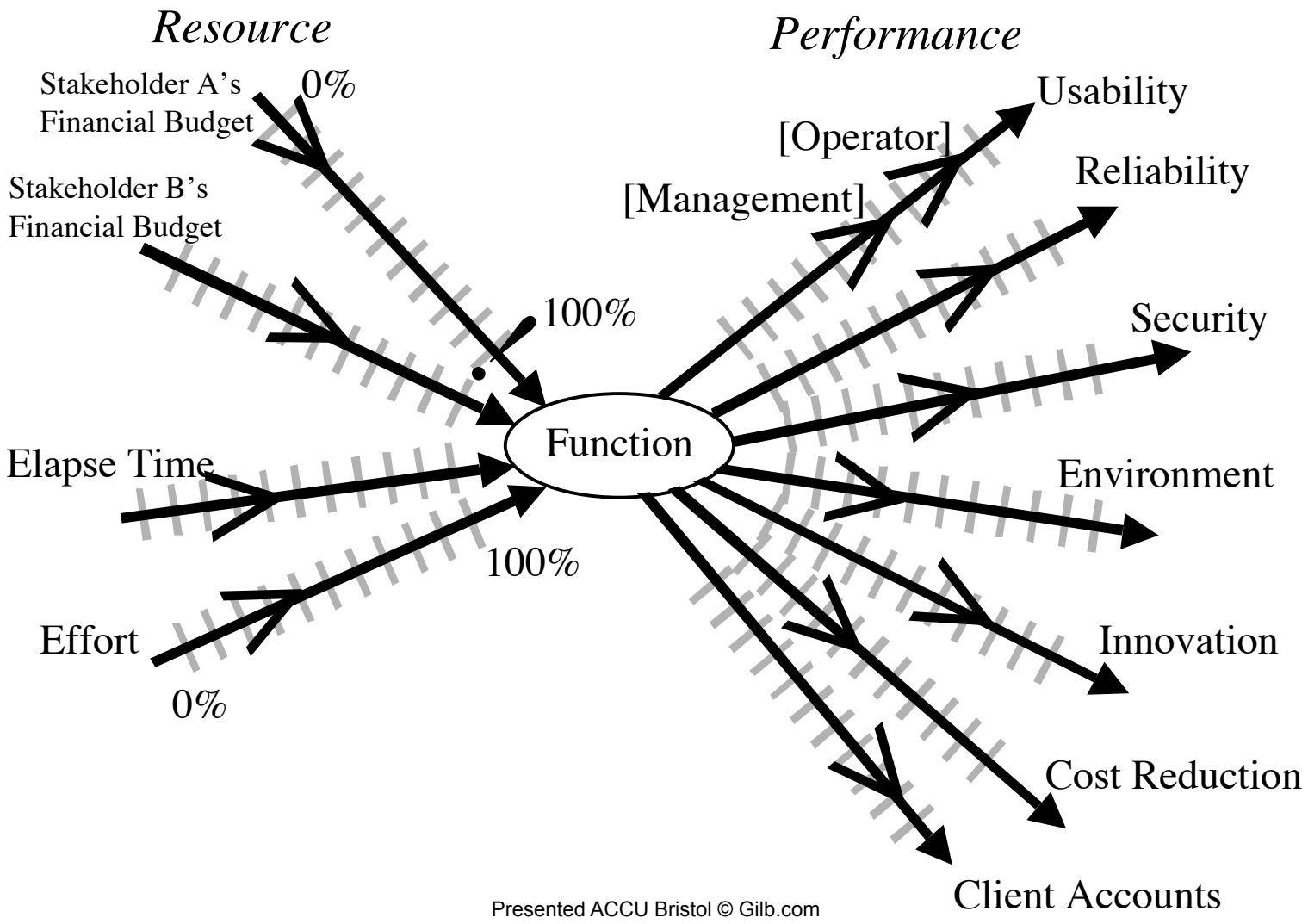
R5: Impacts: For each design idea, specify at least one main performance attribute impacted by it. Use an impact arrow '->' or the Impacts parameter.

R6: Side Effects: Document in the design specification any side effects of the design idea (on defined requirements or other specified potential design ideas) that you expect or fear. Do this using explicit parameters, such as Risks, Impacts [Side Effect] and Assumptions.

R7: Background Information: Capture the background information for any estimated or actual impact of a design idea on a performance/ cost attribute. The evidence supporting the impact, the level of, the level of credibility of any information and the source(s) for all this information should be given as far as possible.

R8: IE table: The set of design ideas specified to meet a set of requirements should be validated at an early stage by using an Impact Estimation (IE) table.

Multiple Required Performance and Cost Attributes are the basis for architecture selection and evaluation



Planguage Glossary

(full glossary 650+ concepts download at www.gilb.com)
http://www.gilb.com/tiki-download_file.php?fileId=387

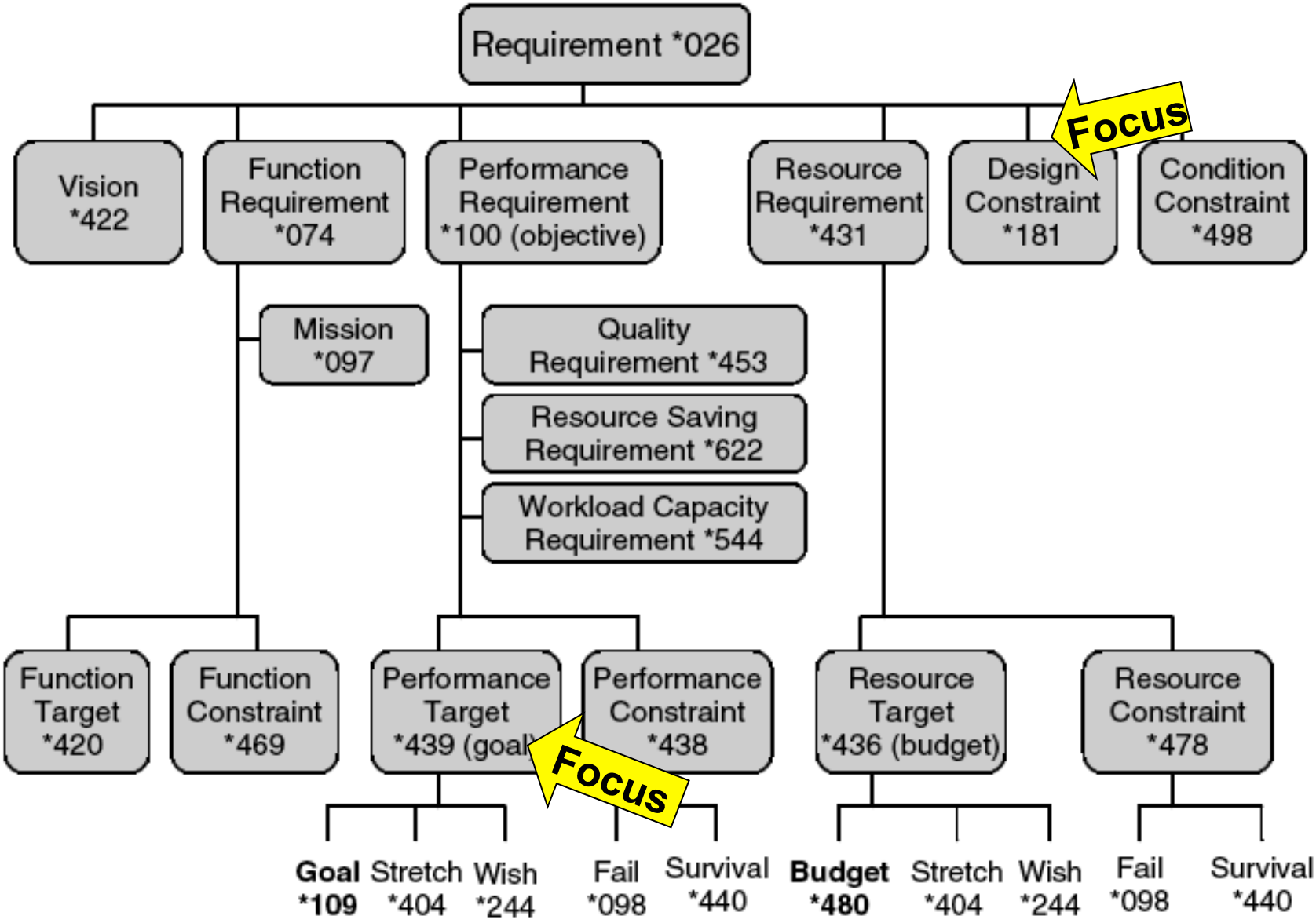
- **Architecture (collective noun):**
  - Concept *192. May 9 2005

- 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.

Architecture *Requirements*

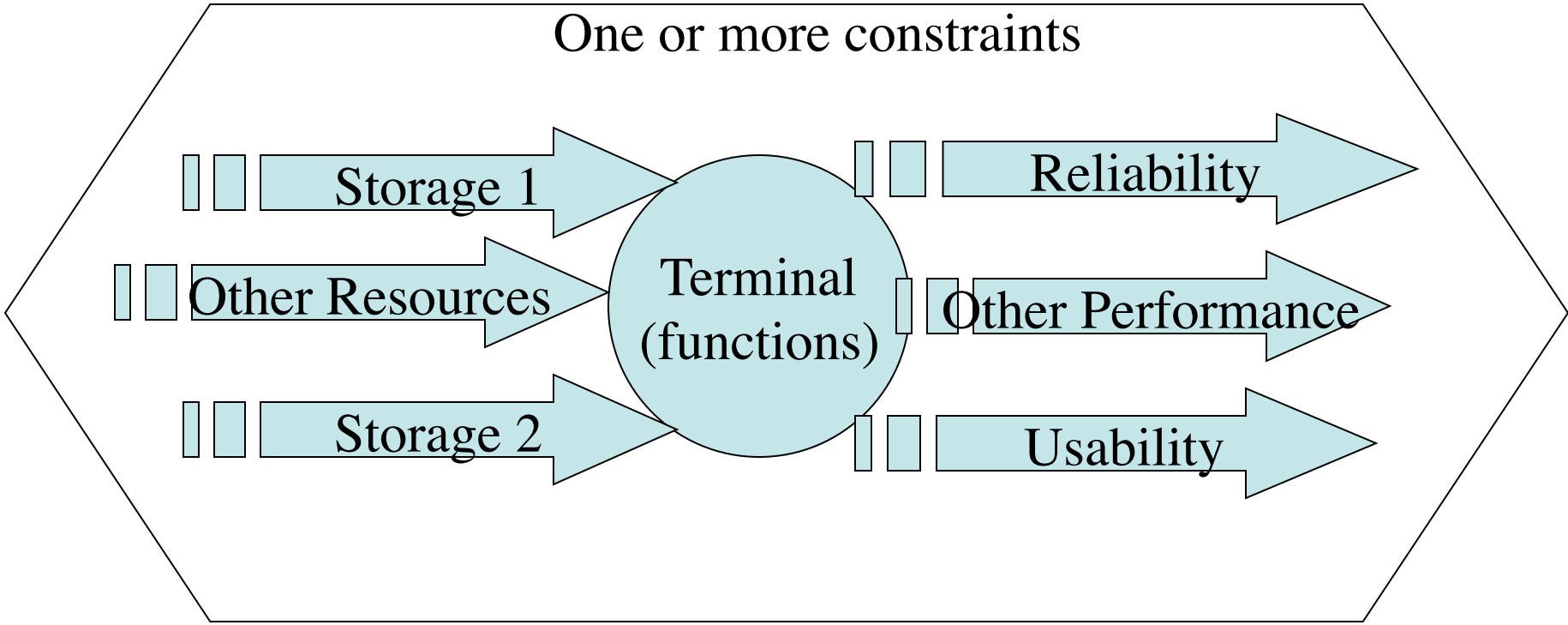
- *Requirements are*
  - *a set of architecture process inputs which include:*
    - *function (what the system must do)*
    - *performance goals (how well it must perform its functions)*
    - constraints
      - (resource constraints, performance constraints, design constraints, other restrictions).

Requirement Concepts <- CE, page 401, Figure G20, *026



Evo and Requirements, Conceptually

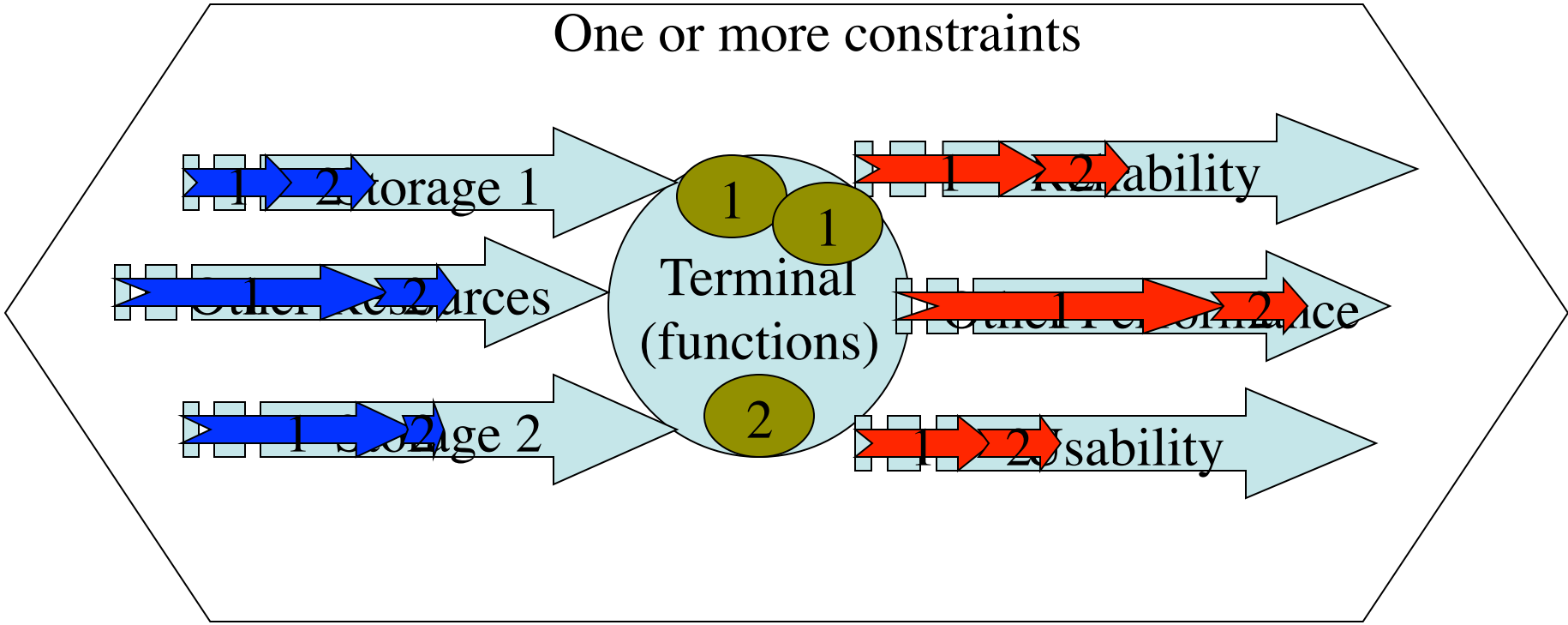
Requirements are the framework for Evo development



Basic requirements model:
We need to meet performance and function requirements,
Within available/planned resources and within constraints.

Evo and Requirements, Conceptually

Evo steps deliver partial requirements



Evo development gradually delivers function and performance, while eating up resources

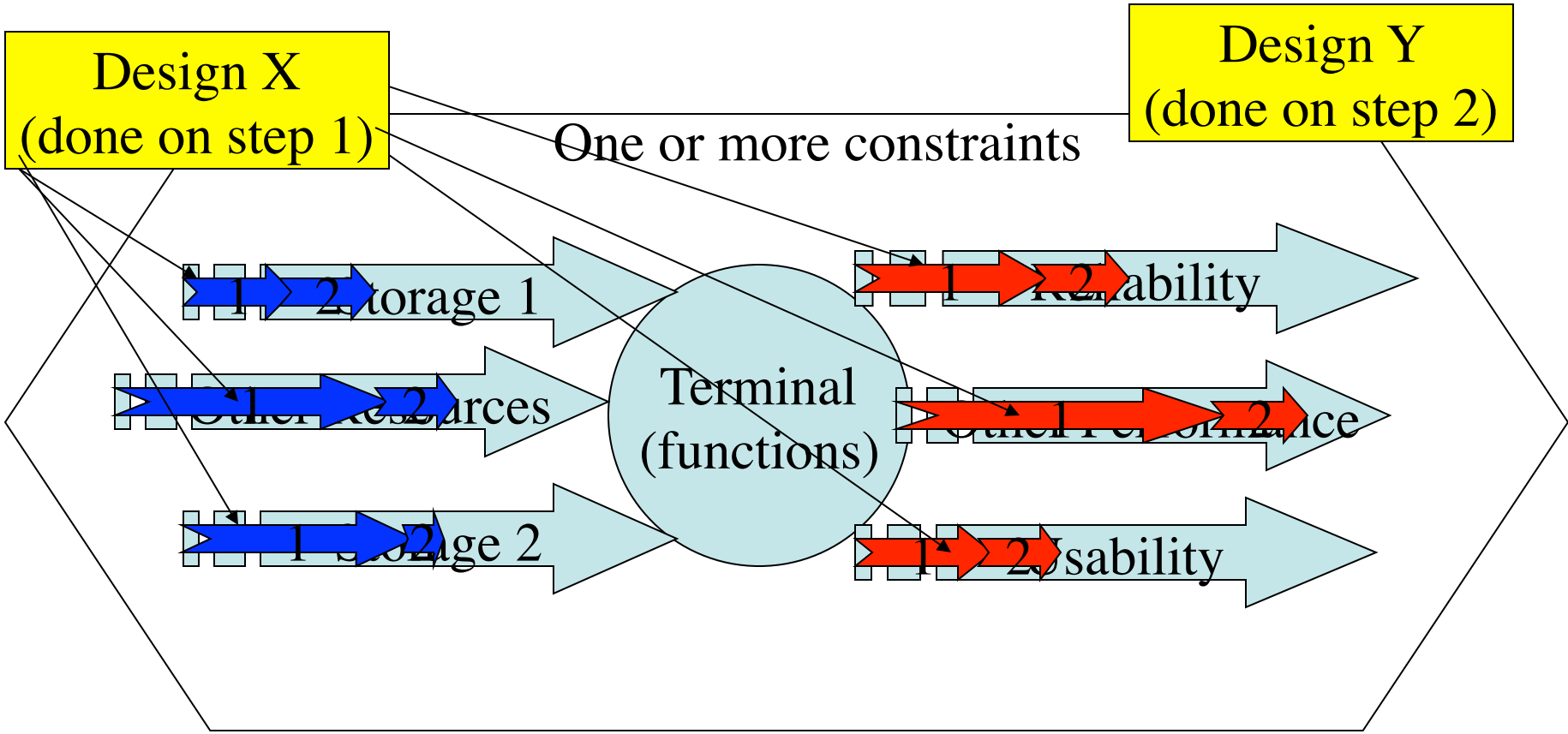
n

n

n

Evo and Requirements, Conceptually

'Design' is what delivers performance, and costs resource



Evo development gradually delivers performance, while eating up resources by Implementing 'design'

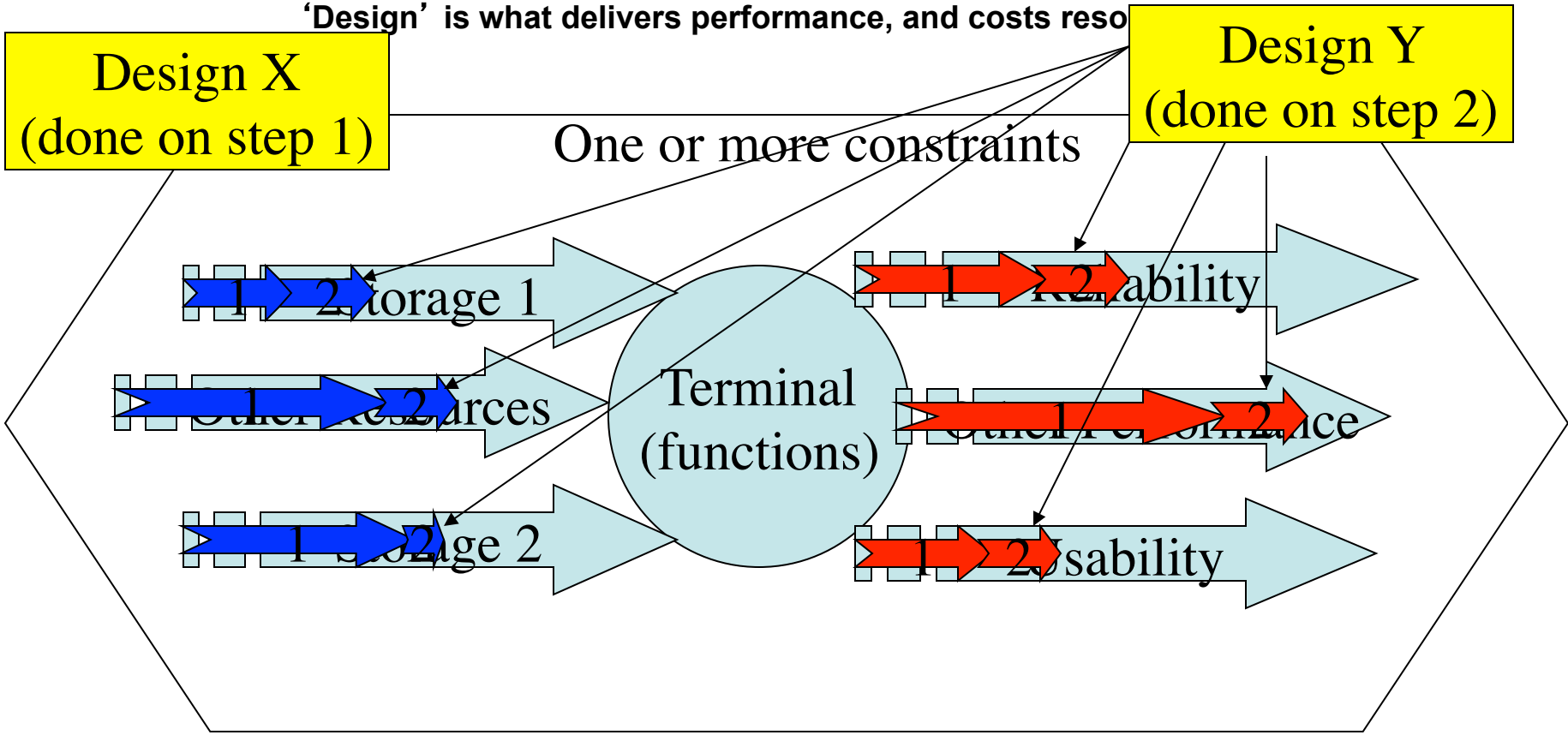
n

n

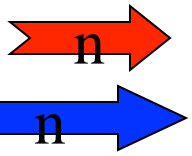
Design _
(done on step n)

Evo and Requirements, Conceptually

'Design' is what delivers performance, and costs reso



Evo development gradually delivers performance, while eating up resources by Implementing 'design'



Design _
(done on step n)

Evo and Requirements, Conceptually

'Design' is what 'delivers performance', and 'costs resource'

Function is selected or built to deliver more function

Evo steps are packages of either function and/or design

Evo steps

Design _
(done on step n)

Fx n

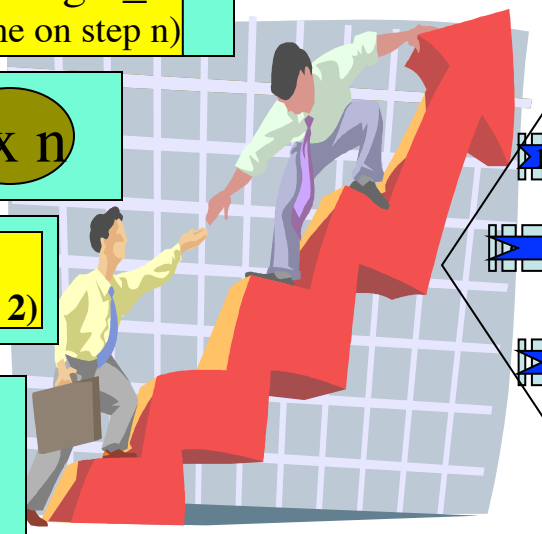
2

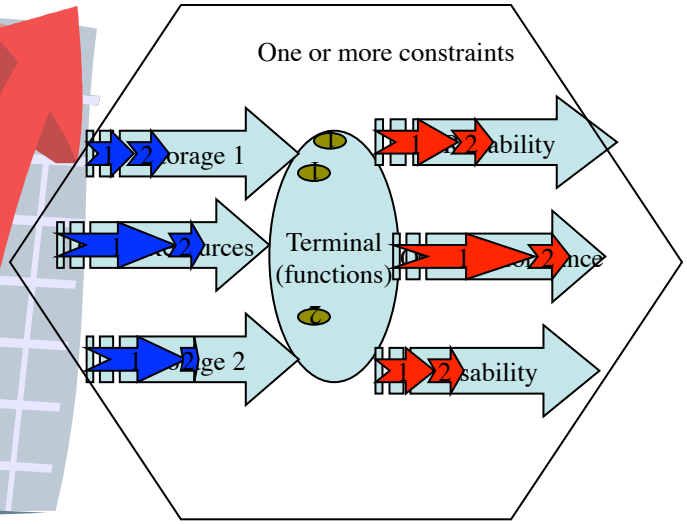
Design Y
(done on step 2)

Fa 1

Fb 1

Design X
(done on step 1)





Evo development
Plans and executes Evo steps
which
Deliver requirements
And cost resources

2

Design Y
(done on step 2)

n

Fx n

n

The **Architecture** is

(collective noun)

- 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.

Some slides at the end, after the end go into some detail about this definition

Engineering Hierarchy

Engineering *224

Systems Engineering *223

Other Engineering

Systecture (Systems Architecture) *564

- Data Structures Strategy
- Application Portfolio Strategy
- Platform Strategy
- Methods Strategy
- Standards Development

Program Management

- Project

Engineering Concepts

- Architecture Process *499

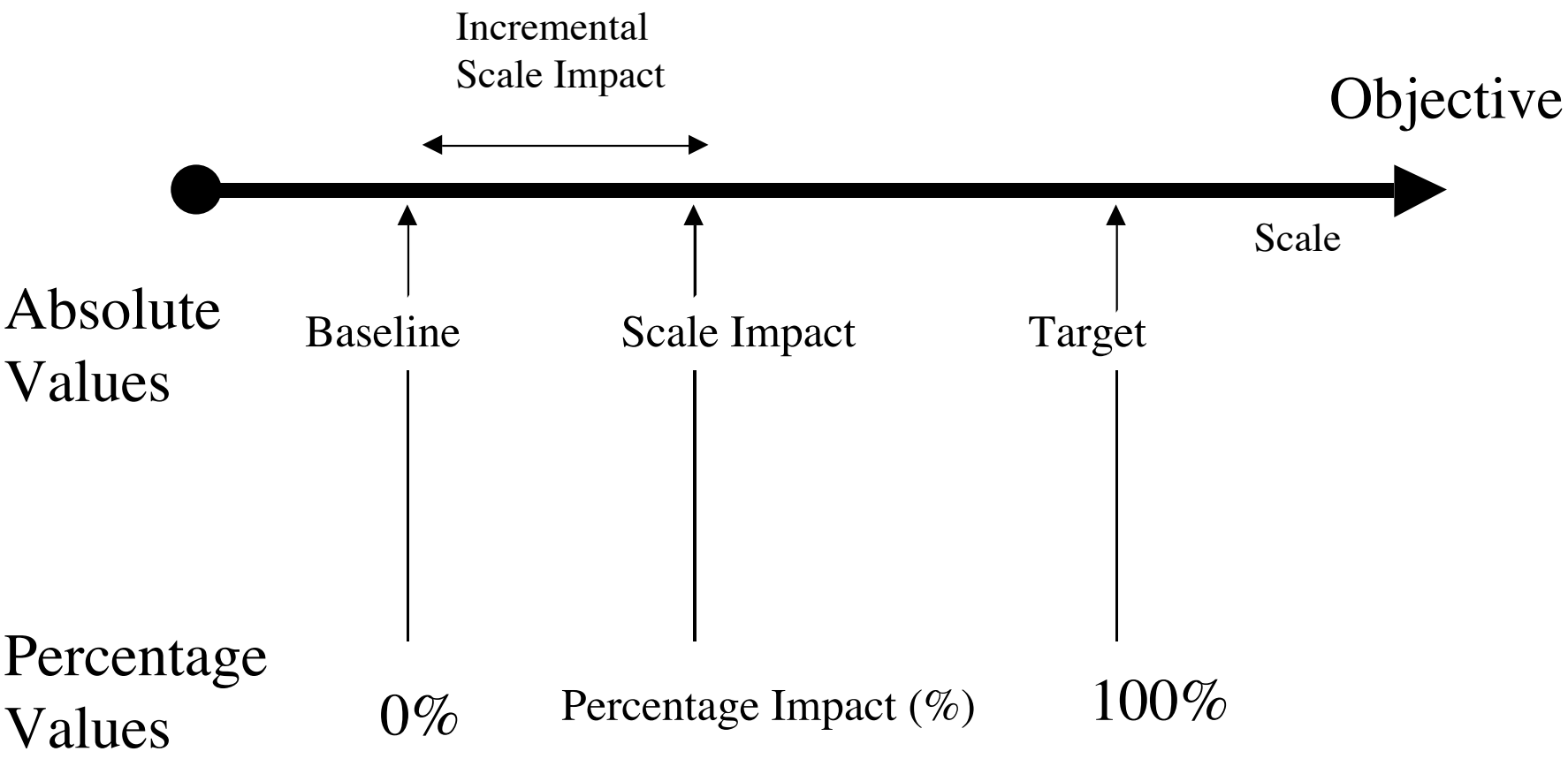
Processes

- Requirements Process *612
- Design Engineering *501
- Evolutionary Project Management (Evo) *355
- Design Process *046
- Impact Estimation *283

Specification Types

- (The) Architecture *192 (Artifacts)
- Architecture Specification *617
- Standards *138
  - Security Standards
  - Interface Standards
  - Requirement Specification Standards
  - Other
- Requirement Specification *508
- Design Specification *586
- Impact Estimation Table
- Evo Step Specification *370
- Evo Plan *322

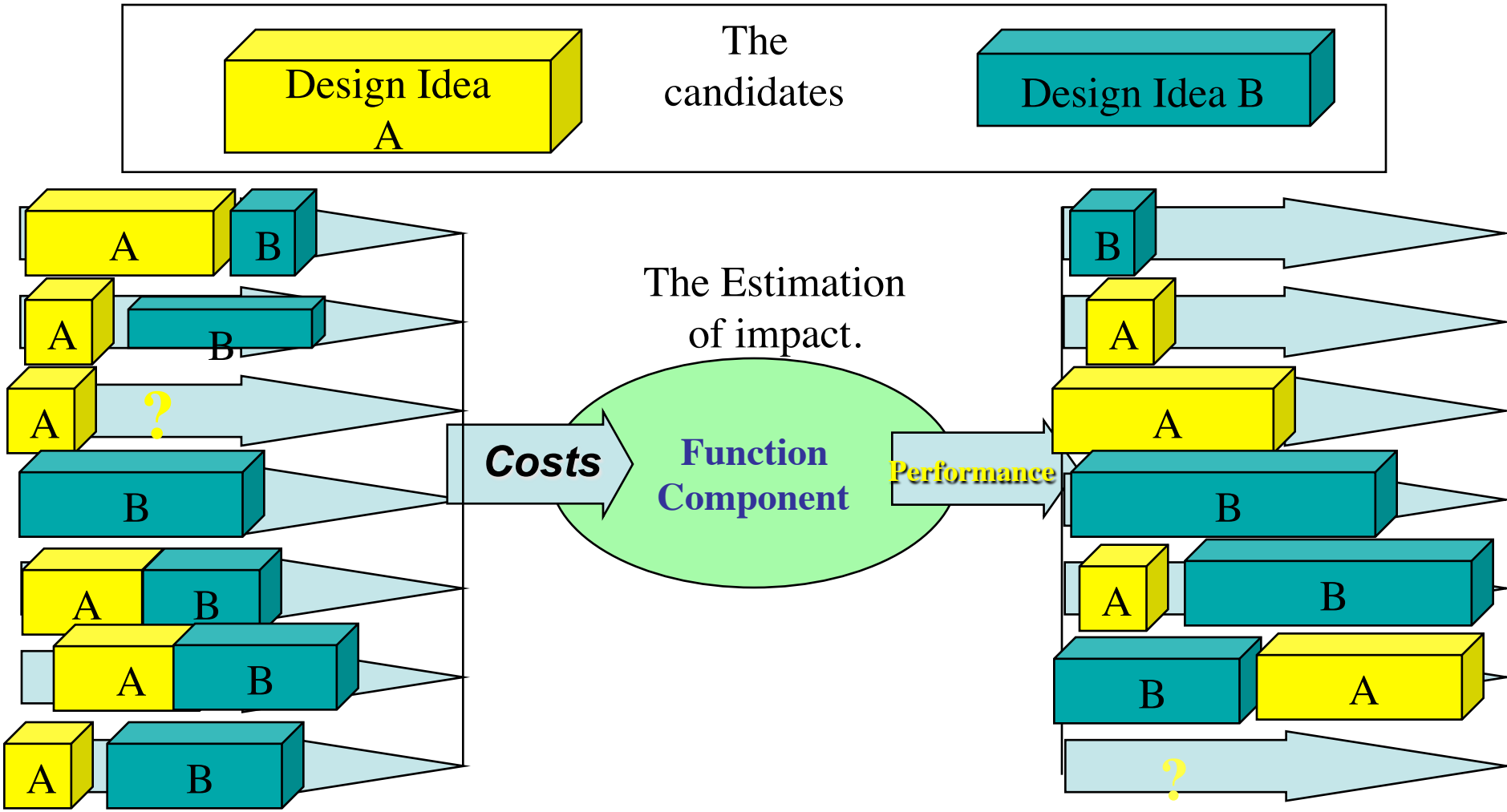
Impact Estimation Basic Concepts



Source: Lindsey Brodie, Editor of Competitive Engineering May 2000

Impact Estimation:

How much do designs impact all critical cost and quality attributes?



•Figure 1: Real (NON-CONFIDENTIAL version) example of an initial draft of setting the objectives that engineering processes must meet.

Business objective	Measure	Goal (200X)	Stretch goal ('0X)	Volume	Value	Profit	Cash
Time to market	Normal project time from GT to GT5	<9 mo.	[illegible]	X		X	X
Mid-range	Min BoM for The Corp phone	<$90	[illegible]	X		X	X
Platformisation Technology	# of Technology 66 Lic. shipping > 3M/yr	4	6	X		X	X
Interface	Interface units	>11M	>13M	X		X	X
Operator preference	Top-3 operators issue RFQ spec The Corp	[illegible]	[illegible]	X		X	X
Productivity						X	X
Get Torden	Lyn goes for Technology 66 in Sep-04	Yes		X		X	X
Fragmentation	Share of components modified	<10%	<5%		X	X	X
Commoditisation	Switching cost for a UI to another System	>1y	[illegible]s			X	X
Duplication	The Corp share of 'in scope' code in best-selling device	>90%	>95%		X	X	X
Competitiveness	Major feature comparison with MX	Same	Better	X		X	X
User experience	Key use cases superior vs. competition	5	10	X	X	X	X
Downstream cost saving	Project ROI for Licensees	>33%	>66%	X	X	X	X
Platformisation IFace	Number of shipping Lic.	33	55	X		X	X
Japan	Share of of XXXX sales	>50%	>60%	X		X	X
	Numbers are intentionally changed from real ones						

Business Objectives Quantified

Strategy Impact Estimation

Technical Strategies

Objectives

Strategy Impacts on Objectives

Viking Deliverables

Business Objective	hardware adaptation	Telephony	Reference designs	IFace	Modularity	Defend vs Technology 66	Tools	User Experience	GUI & Graphics	Security	Defend vs OCD	Enterprise
Time to market	20%	10%	30%	5%	10%	5%	15%	0%	0%	0%	5%	5%
Mid-range	15%	0%	[illegible]	[illegible]	[illegible]	[illegible]	5%	10%	5%	5%	0%	0%
Platformisation Technology	25%	10%	30%	0%	0%	[illegible]	0%	5%	0%	10%	0%	5%
Interface	5%	15%	15%	0%	5%	0%	5%	0%	0%	10%	0%	10%
Operator preference	0%	10%	[illegible]	[illegible]	[illegible]	[illegible]	5%	10%	10%	20%	5%	10%
Get Torden	25%	10%	10%	10%	0%	20%	0%	10%	-20%	10%	10%	5%
Commoditisation	20%	10%	20%	10%	-20%	25%	15%	0%	0%	5%	10%	5%
Duplication	15%	10%	10%	[illegible]	[illegible]	40%	0%	0%	0%	5%	20%	5%
Competitiveness	10%	15%	20%	0%	10%	20%	10%	10%	20%	10%	10%	10%
User experience	5%	[illegible]	[illegible]	0%	[illegible]	[illegible]	0%	30%	10%	0%	0%	0%
Downstream cost saving	15%	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	10%	0%	0%	10%	5%
Platformisation IFace	10%	10%	20%	40%	0%	20%	5%	0%	0%	0%	0%	5%
Japan	10%	5%	20%	0%	10%	0%	0%	10%	5%	0%	0%	0%
Contribution to overall result	15%	9%	17%	4%	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	6%	6%	5%
Cost (£M)	£ 2.85	£ 0.49	£ 3.21	£ 2.54	£ 1.92	£ [illegible]	[illegible]	[illegible]	1.21	£ 2.68	£ 0.79	£ 0.62	£ 0.60
ROI Index (100=average)	106	[illegible]	109	33	78	[illegible]	[illegible]	[illegible]	407	10	152	202	174

Cost

Benefit/Cost ratio

The **Evo Startup** Process
a practical example of high level Architecture Engineering

- The ‘standards for Startup are at
  - Evo Startup Standard, Jan 12 2013
  - http://www.gilb.com/dl562

- Evo Project Management Standard, Jan 12 2013
  - http://www.gilb.com/dl563

Startup Process Day 1 and 2

- Day 1: **Project Objectives**: The top few critical objectives quantified.
  - Objective: Determine, clarify, agree critical few project objectives – results – end states
  - **Process**:
    - Analyze current documentation and slides, for expressed or implied objectives (often implied by designs or lower level objectives)
    - Develop list of **Stakeholders** and their needs and values
    - Brainstorm 'top ten' critical objectives names list. Agree they are top critical few.
    - Detail definition in Planguage – meaning quantify and define clearly, unambiguously and in detail (a page)
    - Quality Control Objectives for Clarity: Major defect measurement. Exit if less than 1.0 majors per page
    - Quality Control Objectives for Relevance: Review against higher level objectives than project for alignment.
    - Define Constraints: resources, traditions, policies, corporate IT architecture, hidden assumptions.
    - Define Issues – yet unresolved
    - Note we might well choose to several things in *parallel*.
  - **Output**: A solid set of the top few critical *objectives* in quantified and measurable language. *Stakeholder* data specified.
  - **Participants**: anybody who is concerned with the business results, the higher the management level the better.
  - **End of Day Process**: meet 30 minutes with any responsible interested managers to present the outputs, and to get preliminary corrections and go-ahead.
  - **Note**: this process is so critical and can be time consuming, so if necessary it can spill over to next day. Perhaps in parallel with startup of the strategy identification. *Nothing is more critical or fundamental than doing this well.*
- Day 2: **Project Strategies and Architecture**: the top few critical strategies for reaching the critical objectives
  - **Objective**: to identify the top 'ten' most critical strategic decisions or architectures; the ones that will contribute or enable us most, to reach our primary objective goal levels on time.
  - **Process**:
    - Analysis of current documentation and slides to identify candidate strategies, implied or expressed.
    - Brainstorming of the 'names' of the specific strategy list, the top ten and a set of less powerful ideas (say 11-30)
    - Detail each top ten strategy sufficiently to understand impacts (on objectives, time and costs)
    - Specify, for each strategy all critical related information (like stakeholders, risks, assumptions, constraints, etc.)
    - Quality Control for clarity – correct unclear items. Exit based on defect level, or not.
    - Likely that work will need to be done in parallel in order to do ten strategies to a rich level of specification.
  - **Output**: A formal strategy specification, ready for evaluation, and decomposition and delivery of partial value results.
  - **Participants**: system architects, project architects, strategy planners. And members of the project team who will be in on the entire weeks process. The major input here is technical and organizational strategy (the means to reach the objectives)
  - **End of Day Process: :** meet 30 minutes with any responsible interested managers to present the outputs, and to get preliminary corrections and go-ahead.

Startup Process
Day 3 and 4

Day 3: Evaluation of Strategies using Impact Estimation: our best estimates with experience and risk. How sure are of the major strategy decisions.

- **Objective**: to estimate to primary effects and all side effects of all top critical strategies on all top critical objectives, and on some resources (time, cost, effort). The estimates will be backed up by evidence, or their credibility will be rated low.
- **Process**:
  - Using the objectives and strategies developed on first 2 days as inputs
  - Populate an Impact Estimation table (aka Value Decision Table) with estimates of the expected result of deploying defined strategies. Estimate main intended impacts
  - And all side effects (on other core objectives)
  - And on all resources (time, money. Effort)
  - Estimate ± ranges
  - Specify evidence and sources for estimates
  - Determine Credibility level
  - Quality Control the IE table against standards (Rules for IE in CE book), for possible 'exit' (meets standards)
  - Lots of parallel work needed and expected to do a good job.
- **Output**:
  - A fairly decent Impact Estimation table, possibly a several level set of them.
    - This will tell us if it is safe to proceed (we have good enough strategies)
    - And it will help us prioritize high value deliveries soon.
- **Participants**: architects, planners, anybody with strong views on any of the strategies. The team for the week.
- **Note**: *it might be necessary and desirable, now or later, to do this impact estimation process at 2 or 3 related levels (Business, Stakeholder, IT System) in order to see the Business-IT relationship clearly. This might exceed time limits and be done parallel or later.*
- **End of Day Process**: meet 30 minutes with any responsible interested managers to present the outputs, and to get preliminary corrections and go-ahead.

Day 4: **Evolutionary Step Decomposition**: what are the high value short term value delivery steps we can execute.

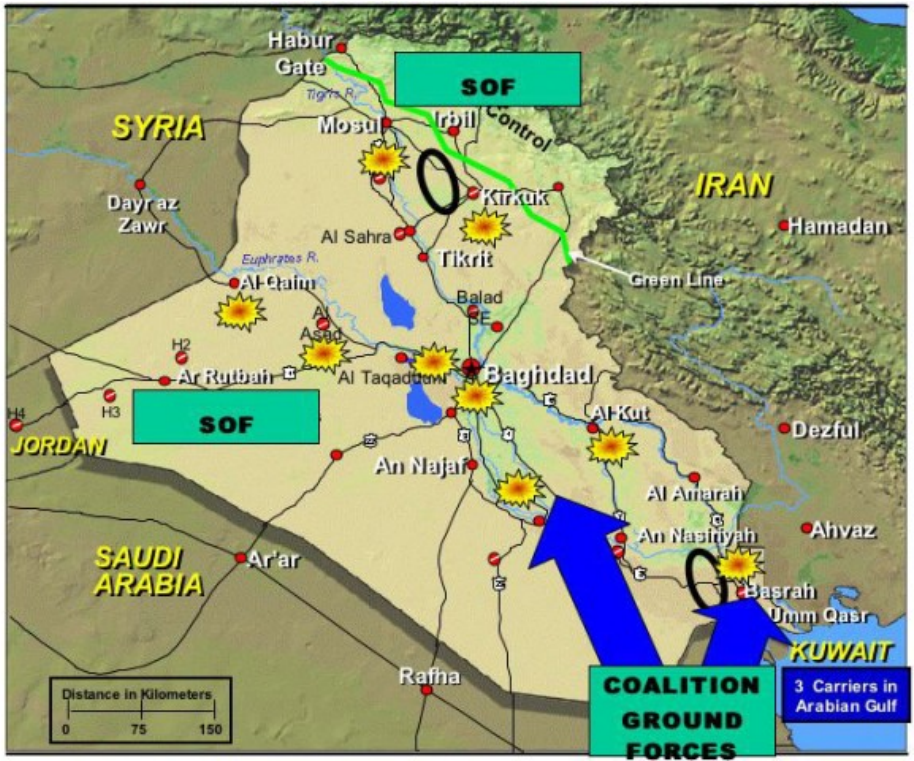
- **Objective**: to identify near team candidates for real value delivery to real stakeholders. What can we do for real next week!
- **Process**:
  - Identify highest value (to costs) strategies and sub-sets of strategies
  - Decompose into doable subsets in weekly to monthly cycles of result delivery
  - Plan the near steps (1 or more) in detail so that we are ready to execute the step in practice.
    - Who does it, main responsible, team.
    - Expected measurable results and costs
    - Stakeholder involved in receiving
    - Test process (for value)
- **Output**: 1 or more potential steps for value delivery to some stakeholders, a plan good enough to approve and execute in practive.
- **Participants**: Project Management, architects prepared to decompose architecture in practice. The weeks team for this start up study.
- **End of Day Process:** meet 30 minutes with any responsible interested managers to present the outputs, and to get preliminary corrections and go-ahead.

Day 5

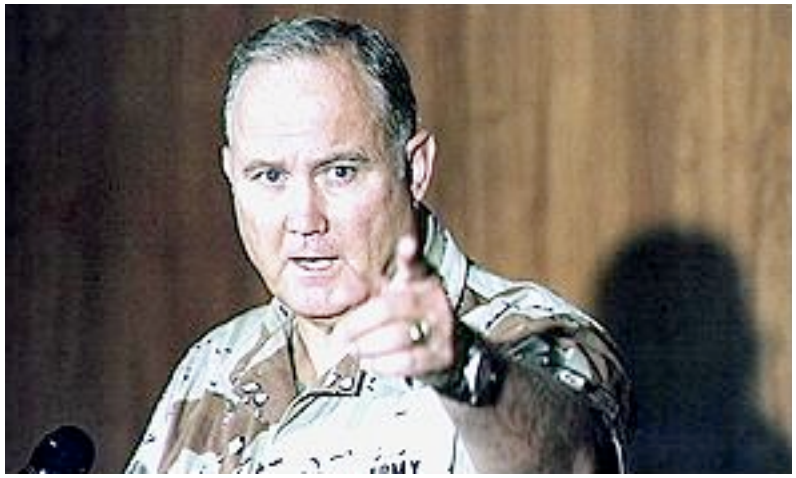
- Boss approves doing the next week

And Now A True War Story

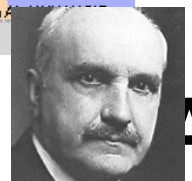
- About Why Bad IT Requirements
  - Can lose a war in Iraq
  - Or at least make it drag on for years



The Persinscom IT System Case



He who does not learn from history
Is doomed to repeat it

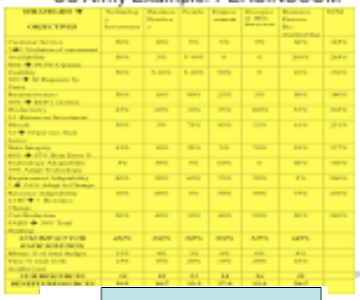


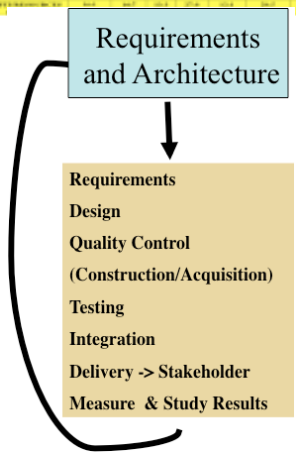
A Man Who understood tha

The Evo Planning Week at DoD

- **Monday**
  - Define top Ten critical objectives, quantitatively
  - Agree that thee are the main points of the effort/project
- **Tuesday**
  - Define roughly the top ten most powerful strategies,
  - for enabling us to reach our Goals on Time
- **Wednesday**
  - Make an Impact Estimation Table for Objectives/Strategies
  - Sanity Test: do we seem to have enough powerful strategies to get to our Goals, with a reasonable safety margin?
- **Thursday**
  - Divide into rough delivery steps (annual, quarterly)
  - Derive a delivery step for 'Next Week'
- **Friday**
  - Present these plans to approval manager (Brigadier General Palicci)
  - get approval to deliver next week

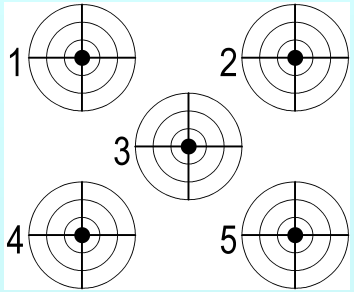
US Army Example: PERSINSCOM





US Army Example: PERSINSCOM: Personnel System

STRATEGIES ➔ OBJECTIVES
Customer Service ?➔0 Violation of agreement
Availability 90% ➔ 99.5% Up time
Usability 200 ➔ 60 Requests by Users
Responsiveness 70% ➔ ECP's on time
Productivity 3:1 Return on Investment
Morale 72 ➔ 60 per mo. Sick Leave
Data Integrity 88% ➔ 97% Data Error %
Technology Adaptability 75% Adapt Technology
Requirement Adaptability ? ➔ 2.6% Adapt to Change
Resource Adaptability 2.1M ➔ ? Resource Change
Cost Reduction FADS ➔ 30% Total Funding



Monday

←The Top Ten

Critical

Objectives

Were decided

Sample of Objectives/Strategy definitions
US Army Example: PERSINSCOM: Personnel System

- *Example of one of the Objectives:*

Customer Service:

Type: Critical Top level Systems Objective

Gist: Improve customer perception of quality of service provided.

Scale: Violations of Customer Agreement per Month.

Meter: Log of Violations.

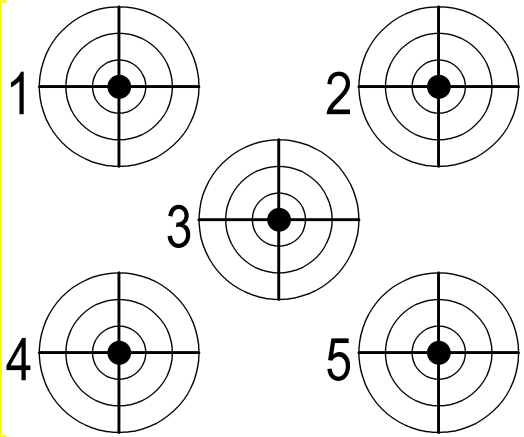
Past [Last Year] Unknown Number ←State of PERSCOM Management Review

Record [NARDAC] 0 ? ← NARDAC Reports Last Year

Fail : <must be better than Past, Unknown number> ←CG

Goal [This Year, PERSINCOM] 0 "Go for the Record" ← Group SWAG

.



US Army Example: PERSINSCOM: Personnel System

STRATEGIES ➔ OBJECTIVES	Technology Investment	Business Practices	People	Empow-erment	*Principles of IMA Management*	Business Process Re-engineering	SUM
Customer Service ?➔0 Violation of agreement							
Availability 90% ➔ 99.5% Up time							
Usability 200 ➔ 60 Requests by Users							
Responsiveness 70% ➔ ECP's on time							
Productivity 3:1 Return on Investment							
Morale 72 ➔ 60 per mo. Sick Leave							
Data Integrity 88% ➔ 97% Data Error %							
Technology Adaptability 75% Adapt Technology							
Requirement Adaptability ? ➔ 2.6% Adapt to Change							
Resource Adaptability 2.1M ➔ ? Resource Change							
Cost Reduction FADS ➔ 30% Total Funding							

Tuesday
The Top Ten
Critical Strategies
For reaching the
←objectives
Were decided

A Strategy (Top Level of Detail)

Technology Investment:

Gist: Exploit investment in high return technology.

Impacts: productivity, customer service and conserves resources.

Wednesday:
Day 3 of 5 of 'Feasibility Study

- We made a rough evaluation
  - of how powerful our strategies might be
  - in relation to our objectives

- Impact Estimation Table
  - 0% Neutral, no ± impact
  - 100% Gets us to Goal level on time
  - 50% Gets us half way to Goal at deadline
  - -10% has 10% negative side effect

STRATEGIES → **OBJECTIVES**	Technology Investment	Business Practices	People	Empow-erment	*Principles of IMA Management*	Business Process Re-engineering	SUM
Customer Service ?→0 Violation of agreement	50%	10%	5%	5%	5%	60%	185%
Availability 90% → 99.5% Up time	50%	5%	5-10%	0	0	200%	265%
Usability 200 → 60 Requests by Users	50%	5-10%	5-10%	50%	0	10%	130%
Responsiveness 70% → ECP's on time	50%	10%	90%	25%	5%	50%	180%
Productivity 3:1 Return on Investment	45%	60%	10%	35%	100%	53%	303%
Morale 72 → 60 per mo. Sick Leave	50%	5%	75%	45%	15%	61%	251%
Data Integrity 88% → 97% Data Error %	42%	10%	25%	5%	70%	25%	177%
Technology Adaptability 75% Adapt Technology	5%	30%	5%	60%	0	60%	160%
Requirement Adaptability ? → 2.6% Adapt to Change	80%	20%	60%	75%	20%	5%	260%
Resource Adaptability 2.1M → ? Resource Change	10%	80%	5%	50%	50%	75%	270%
Cost Reduction FADS → 30% Total Funding	50%	40%	10%	40%	50%	50%	240%
SUM IMPACT FOR EACH SOLUTION	***482%***	***280%***	***305%***	***390%***	***315%***	***649%***	
Money % of total budget	15%	4%	3%	4%	6%	4%	
Time % total work months/year	15%	15%	20%	10%	20%	18%	
SUM RESOURCES	***30***	***19***	***23***	***14***	***26***	***22***	
BENEFIT/RESOURCES RATIO	***16:1***	**14:7**	**13:3**	**27:9**	**12:1**	**29:5**	

MEASURING HAND FOR GLOVE SIZE

US DoD. Persinscom **Impact EstimationTable**:

Designs

Requirements

Design Ideas ->	*Technology Investment*	*Business Practices*	*People*	*Empowerment*	*Principles of IMA Management*	*Business Process Re-engineering*	*Sum Requirements*
	50%	1[illegible]	5%	5%	5%	60%	185%
Availability 90% <-> 99.5% Up time	50%		5–10%	0%	0%	200%	265%
Usability 200 <-> 60 Requests by Users			5–10%	50%	0%	10%	130%
Responsiveness 70% <-> ECP's on time	50%	10%	90%	25%	5%	50%	180%
Productivity	45%						303%
3:1 Return on Investment	50%						251%
Morale 72 <-> 60 per month on Sick Leave							
Data Integrity 88% <-> 97% Data Error %	42%						177%
Technology Adaptability 75% Adapt Technology	5%	30%	5%	60%	0%	60%	160%
Requirement Adaptability ? <-> 2.6% Adapt to Change	80%	20%	60%	75%	20%	5%	260%
Resource Adaptability 2.1M <-> ? Resource Change	10%	80%	5%	50%	50%	75%	270%
Cost Reduction FADS <-> 30% Total Funding	50%	40%	10%	40%	50%	50%	240%
Sum of Performance	*482%*	*280%*	*305%*	*390%*	*315%*	*649%*	
Money % of total budget	15%	4%	3%	4%	6%	4%	36%
Time % total work months/year	15%	15%	20%	10%	20%	18%	98%
Sum of Costs	*30*	*19*	*23*	*14*	*26*	*22*	
Performance to Cost Ratio	*16:1*	*14:7*	*13:3*	*27:9*	*12:1*	**29.5 :1**	

Estimated Impact of

Design
-> Requirements

US Army Example: PERSINSCOM: Personnel System

STRATEGIES ➔ **OBJECTIVES**	Technology Investment	Business Practices	People	Empow-erment	*Principles of IMA Management*	Business Process Re-engineering	SUM
Customer Service ?➔0 Violation of agreement	50%	10%	5%	5%	5%	60%	185%
Availability 90% ➔ 99.5% Up time	50%	5%	5-10%	0	0	200%	265%
Usability 200 ➔ 60 Requests by Users	50%	5-10%	5-10%	50%	0	10%	130%
Responsiveness 70% ➔ ECP's on time	50%	10%	90%	25%	5%	50%	180%
Productivity 3:1 Return on Investment	45%	60%	10%	35%	100%	53%	303%
Morale 72 ➔ 60 per mo. Sick Leave	50%	5%	75%	45%	15%	61%	251%
Data Integrity 88% ➔ 97% Data Error %	42%	10%	25%	5%	70%	25%	177%
Technology Adaptability 75% Adapt Technology	5%	30%	5%	60%	0	60%	160%
Requirement Adaptability ? ➔ 2.6% Adapt to Change	80%	20%	60%	75%	20%	5%	260%
Resource Adaptability 2.1M ➔ ? Resource Change	10%	80%	5%	50%	50%	75%	270%
Cost Reduction FADS ➔ 30% Total Funding	50%	40%	10%	40%	50%	50%	240%
SUM IMPACT FOR EACH SOLUTION	***482%***	***280%***	***305%***	***390%***	***315%***	***649%***	
Money % of total budget	15%	4%	3%	4%	6%	4%	
Time % total work months/year	15%	15%	20%	10%	20%	18%	
SUM RESOURCES	***30***	***19***	***23***	***14***	***26***	***22***	
BENEFIT/RESOURCES RATIO	***16:1***	**14:7**	**13:3**	**27:9**	**12:1**	**29.5 :1**	

Impact Estimation: Value-for-Money Delivery Table

STRATEGIES ➔ **OBJECTIVES**	Technology Investment	Business Practices	People	Empow-erment	*Principles of IMA Management*	Business Process Re-engineering	SUM
Customer Service ?➔0 Violation of agreement	50%	10%	5%	5%	5%	60%	185%
Availability 90% ➔ 99.5% Up time	50%	5%	5-10%	0	0	200%	265%
Usability 200 ➔ 60 Requests by Users	50%	5-10%	5-10%	50%	0	10%	130%
Responsiveness 70% ➔ ECP's on time	50%	10%	90%	25%	5%	50%	180%
Productivity 3:1 Return on Investment	45%	60%	10%	35%	100%	53%	303%
Morale 72 ➔ 60 per mo. Sick Leave	50%	5%	75%	45%	15%	61%	251%
Data Integrity 88% ➔ 97% Data Error %	42%	10%	25%	5%	70%	25%	177%
Technology Adaptability 75% Adapt Technology	5%	30%	5%	60%	0	60%	160%
Requirement Adaptability ? ➔ 2.6% Adapt to Change	80%	20%	60%	75%	20%	5%	260%
Resource Adaptability 2.1M ➔ ? Resource Change	10%	80%	5%	50%	50%	75%	270%
Cost Reduction FADS ➔ 30% Total Funding	50%	40%	10%	40%	50%	50%	240%
SUM IMPACT FOR EACH SOLUTION	***482%***	***280%***	***305%***	***390%***	***315%***	***649%***	
Money % of total budget	15%	4%	3%	4%	6%	4%	
Time % total work months/year	15%	15%	20%	10%	20%	18%	
SUM RESOURCES	***30***	***19***	***23***	***14***	***26***	***22***	
BENEFIT/RESOURCES RATIO	***16:1***	**14:7**	**13:3**	**27:9**	**12:1**	**29.5 : 1**	

Thursday:
Day 4 of 5 of ‘Feasibility Study

- We looked for a way to deliver some stakeholder results, next week
- 1 1 1 1 1 1
  - 1 increase from 0%
  - 1 stakeholder
  - 1 quality
  - 1 week
  - 1 Function
  - 1 Design

STRATEGIES ➔ **OBJECTIVES**	Technology Investment	Business Practices	People	Empow-erment	*Principles of IMA Management*	Business Process Re-engineering	SUM
Customer Service ?➔0 Violation of agreement	50%	10%	5%	5%	5%	60%	185%
Availability 90% ➔ 99.5% Up time	50%	5%	5-10%	0	0	200%	265%
Usability 200 ➔ 60 Requests by Users	50%	5-10%	5-10%	50%	0	10%	130%
Responsiveness 70% ➔ ECP’s on time	50%	10%	90%	25%	5%	50%	180%
Productivity 3:1 Return on Investment	45%	60%	10%	35%	100%	53%	303%
Morale 72 ➔ 60 per mo. Sick Leave	50%	5%	75%	45%	15%	61%	251%
Data Integrity 88% ➔ 97% Data Error %	42%	10%	25%	5%	70%	25%	177%
Technology Adaptability 75% Adapt Technology	5%	30%	5%	60%	0	60%	160%
Requirement Adaptability ? ➔ 2.6% Adapt to Change	80%	20%	60%	75%	20%	5%	260%
Resource Adaptability 2.1M ➔ ? Resource Change	10%	80%	5%	50%	50%	75%	270%
Cost Reduction FADS ➔ 30% Total Funding	50%	40%	10%	40%	50%	50%	240%
SUM IMPACT FOR EACH SOLUTION	***482%***	***280%***	***305%***	***390%***	***315%***	***649%***	
Money % of total budget	15%	4%	3%	4%	6%	4%	
Time % total work months/year	15%	15%	20%	10%	20%	18%	
SUM RESOURCES	***30***	***19***	***23***	***14***	***26***	***22***	
BENEFIT/RESOURCES RATIO	***16:1***	**14:7**	**13:3**	**27:9**	**12:1**	**29:5**	

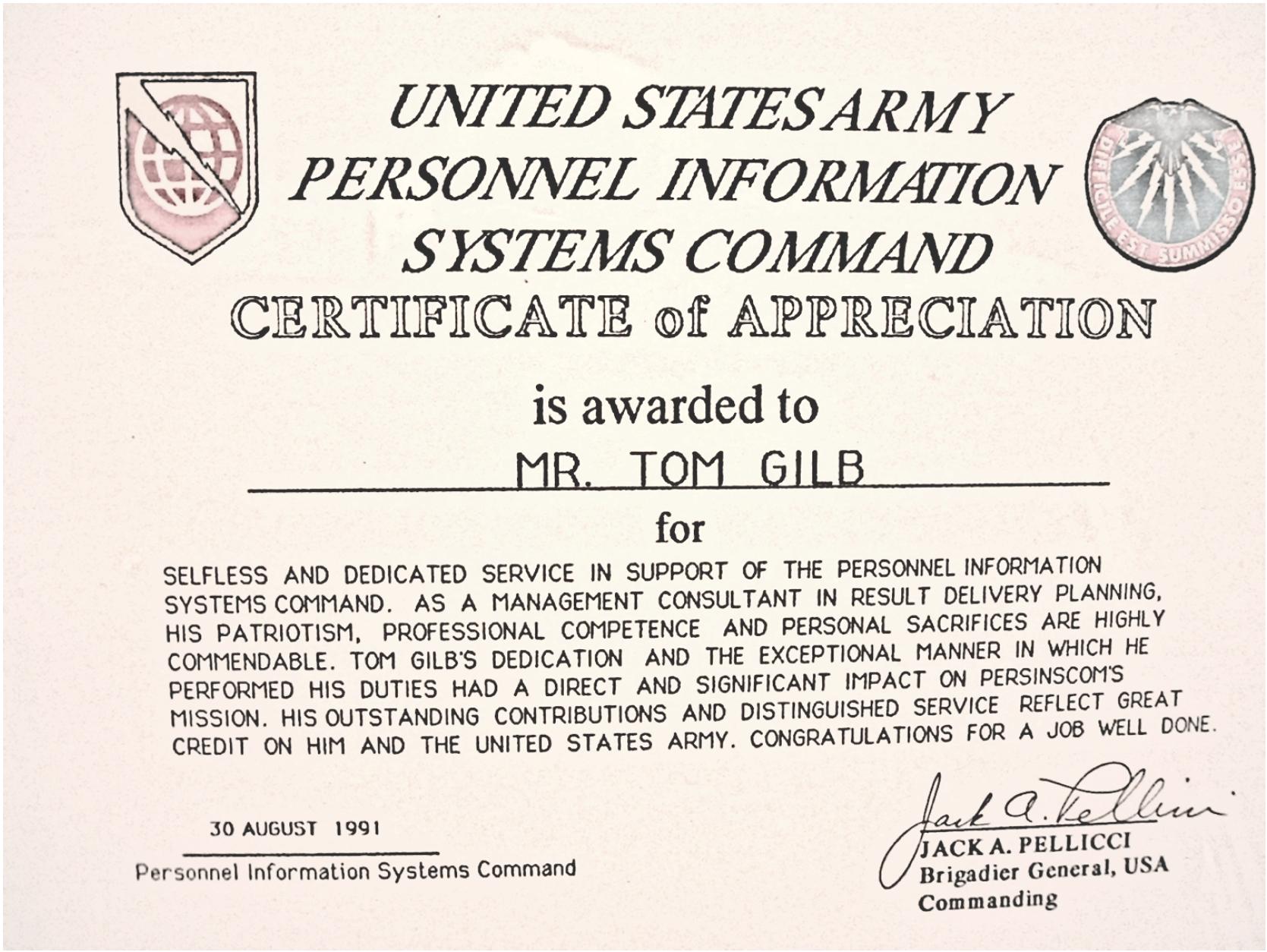
Next weeks Evo Step??

- **"You won't believe we never thought of this, Tom!'**

- **The step:**
  - **When the Top General Signs in**
  - **Move him to the head of the queue**
    - **Of all people inquiring on the system.**

- **Can you deliver it next week?**
  - **Its already done: 1If General, move to head of queue'**

The Reward for Service



UNITED STATES ARMY
PERSONNEL INFORMATION
SYSTEMS COMMAND

CERTIFICATE of APPRECIATION

is awarded to

MR. TOM GILB

for

SELFLESS AND DEDICATED SERVICE IN SUPPORT OF THE PERSONNEL INFORMATION SYSTEMS COMMAND. AS A MANAGEMENT CONSULTANT IN RESULT DELIVERY PLANNING, HIS PATRIOTISM, PROFESSIONAL COMPETENCE AND PERSONAL SACRIFICES ARE HIGHLY COMMENDABLE. TOM GILB'S DEDICATION AND THE EXCEPTIONAL MANNER IN WHICH HE PERFORMED HIS DUTIES HAD A DIRECT AND SIGNIFICANT IMPACT ON PERSINSCOM'S MISSION. HIS OUTSTANDING CONTRIBUTIONS AND DISTINGUISHED SERVICE REFLECT GREAT CREDIT ON HIM AND THE UNITED STATES ARMY. CONGRATULATIONS FOR A JOB WELL DONE.

30 AUGUST 1991

Personnel Information Systems Command

JACK A. PELLICCI
Brigadier General, USA
Commanding

Requirements Exercises
Short Version

- Stakeholders
- Stakeholder & Critical Value
- Most critical quality impacting a critical value
- Ambition Level
- Scale of measure
- Goal Level
- Tolerable Level

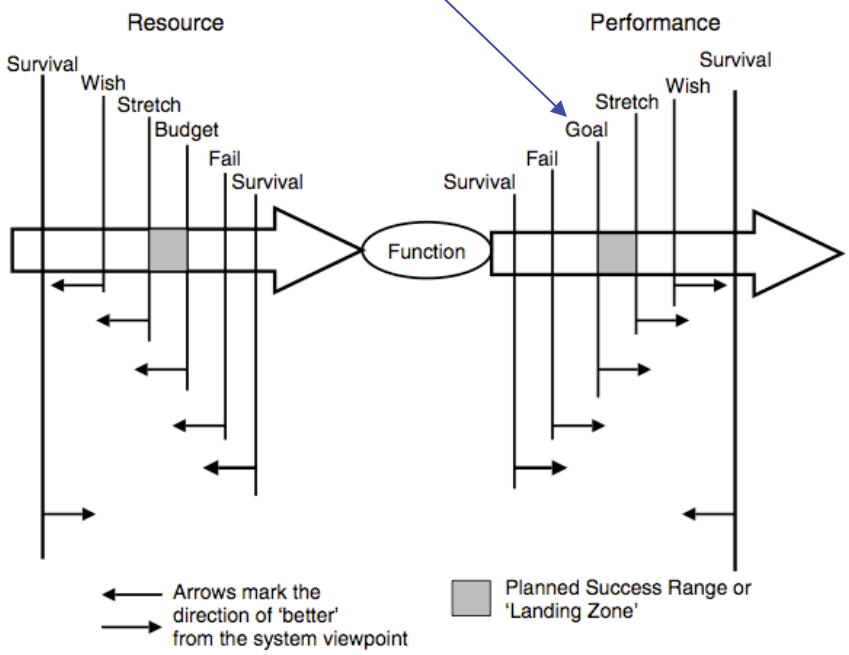
”Stakeholder”

- ***Stakeholders** include*
  - ***any person,***
  - ***organizational grouping***
  - ***or other entity,***
  - ***internal or external to a given development project,***
  - ***of any kind***
  - ***which observably has requirements (performance goals, function or constraints) regarding a system,***
    - ***whether these requirements are known, accepted, formalized, specified or not yet does not disqualify a stakeholder from potentially influencing architecture to satisfy its requirements.***
    - ***This is a much needed generalization of the concept of ‘client’. (‘Architect satisfies client needs’)***

Conditions for A Goal Level

When is a goal level really valid? <-CE 366, *109

1. Technically possible - within state of art
2. Economically Possible - resources exist
3. Costs consistent with other Requirements
4. Effective, and effect necessary to satisfy stakeholder needs
5. Profitable: value over cost
6. Prioritized: by any rules of priority
    1. Effectiveness
    2. Profitability
    3. Politics
7. All [Conditions] in the Goal statement are ‘true’



(Quality) Requirements Specification Template with <hints>

HOW WE SPECIFY SCALAR ATTRIBUTE PRIORITY

<name tag of the objective>

Ambition: **<give overall real ambition level in 5-20 words>**

Version: **<dd-mm-yy each requirements spec has a version, at least a date>**

Owner: **<the person or instance allowed to make official changes to this requirement>**

Type: **<quality|objective|constraint>**

Stakeholder: **{ , , }** **"who can influence your profit, success or failure?"**

Scale: **<a defined units of measure, with [parameters] if you like>**

Meter **[<for what test level?>]**

====Benchmarks ========== the Past

Past **[ ]** **<estimate of past> <--<source>**

Record **[<where>, <when >, <estimate of record level>]** **<-- <source of record data>**

Trend **[<future date>, <where?>]** **<prediction of level>** **<-- <source of prediction>**

===== Targets ========== the future needs

Wish **[ ]** **<-- <source of wish>**

Goal **[…] <target level>** **<-- Source**

Value **[Goal] <refer to what this impacts or how much it creates of value>**

Stretch **[ ] <motivating ambition level>** **<-- <source of level>**

========== Constraints ==========

Fail **[ ]** **<-- <source>** **'Failure Point'**

Survival **[ ]** **<- <source of limit>** **'Survival Point'**

Design Exercises
Short Version

- Most powerful design for reaching the Goal level on time
- Components of this design

Impact Estimation
Short Version

- Cost of this design
- Estimated % of impact on the goal (100% = Goal on time)
- Side effects on other Performance/Quality/ Cost aspects

Value Delivery Step
Short version

- Identify the smallest (days to implement) implementable component of your design that can have some impact on your Goal
- How much impact % will it have?
- How many days to implement will it take

Ask for free digital copy! (tom@gilb.com)

Questions and Discussion

- On Real Architecture

Advanced Reserve Slides

- Which we do not plan to present at this Conference
- But are in reserve
- They can give you more detail
- And might be used to answer questions in more detail

Software and Systems Engineering

- Our opinion about Software Architecture applies fully to the higher level of the system of which our ‘code’ is a component

- i.e. it is a *systems* engineering perspective

Rationale: (for the Architecture definition)

- ***Rationale: this definition has the following intents by the author (TG):***
- ***to bring in the concept that architecture is related to multiple requirements,***
- ***and must be judged in terms of***
  - ***its satisfaction,***
  - ***and optimization degree,***
  - ***for multiple performance goals,***
  - ***within multiple constraints.***
    - ***This seems missing in other definitions [Maier02, Art of Architecting]***
- ***to avoid the notion that architecture is done by one instance,***
  - ***it can exist and have evolved, even in a 'new' system.***
- ***to avoid the notion that architecture***
  - ***is formally specified (this can be stated as an adjective, 'architecture specification', see below)***
- ***to differentiate architecture from other design***
  - ***by invoking the notion that it has the power to constrain the decisions of other engineering levels***

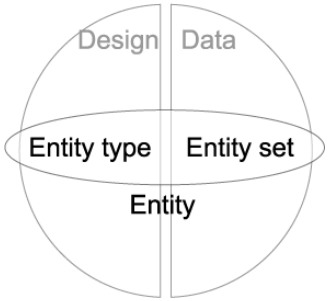
Rejected Architecture Notions

- *In particular I reject some notions common in other definitions of architecture:*
- ***structure** (MIL STD 498, Maier02 p285) : this term is commonly used to define architecture.*
  - *Even in Civil Architecture it is at best one category of the architecture.*
  - *In systems engineering it is practically, but not totally, irrelevant.*
  - *It hides the more central notion of a 'design artifact',*
    - *which is something that determines system properties or enables them*
    - *. (this point is also made by IEEE Architecture Working Group [Maier02, p285-6])*
- ***component, interfaces & connections**: same principle as for 'structure',*
  - *these describe specific but narrow classes of design artifacts.*
  - *This in practice leads to the exclusion of the more general concept of 'anything which satisfies the requirements'.*
  - *It certainly does not include concepts like training, operator selection, motivation, human communication, contracts, policies and other 'non-hardware',*
    - *which can be every bit as dramatic in influencing the architecture's impact on the system requirements.*

Interpretations of terms used in the definition of 'The Architecture':

"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."

*What do we mean by the "**Set**" (of entities):*

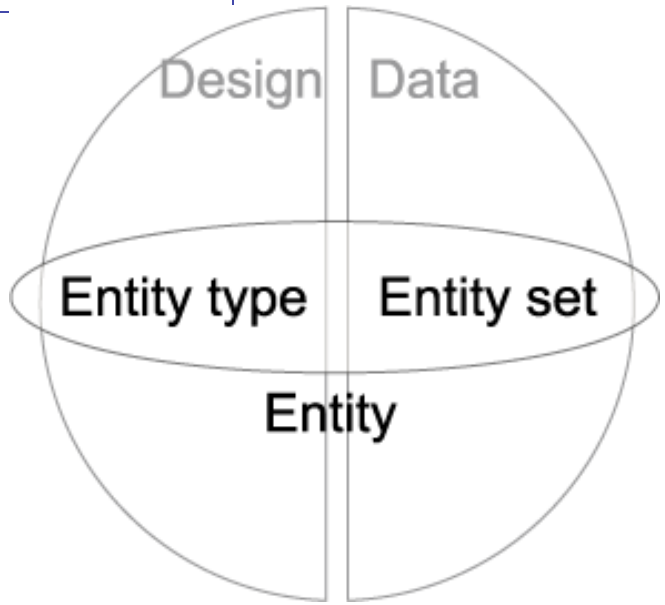


- *the notion of a set of entities,*
- *the notion of the architecture as a 'set' of arbitrarily different devices*
  - *for impacting*
  - *or controlling*
  - *the attributes of a system.*

- 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.

Why do we use the term "Entities":

- *this is intended to be **extremely broad** in scope*
  - *covering **everything imaginable** and discernable*
  - *which is **intended to satisfy** requirements,*
  - *and which is **intended to constrain other design**, operational environment, or life cycle activity.*
- *In particular it goes **way beyond the traditional notion** of structure, and organization.*
- *It for example includes notions of agreements, contracts, social mores, and motivation -*
  - *which never seem to get mentioned in the conventional definitions.*
- *It is also intended to cover all discernible mechanisms which are operating at this level,*
  - *no matter who selected them, when they were selected, or if the formal 'architects' are aware of them.*
- *Entities are **not** necessarily design **specifications** (*586).*
- *They are the existing design concepts (*047) themselves, no matter how they are represented, or determined.*



" in fact exist ":

- ***the design artifacts may 'exist' because of***
  - ***Conscious selection (design), tradition, accident or unintentionally, - even foolishly,***
  - ***by anybody or anything –***
    - ***including cultures, legal systems, political systems, and nature – even the formal 'architect'.***
  - ***But the point is that they are in fact in existence***
    - ***in either a real system or a model of such a system.***
  - ***The selection is not necessarily a conscious act for formal engineering***
  - ***but the design artifact is observably in place and in force – irrespective of its history.***

Implication

- An architect,
- Doing an architecture process
- May add conscious and intentional architecture entities
- To an *existing* architecture
- Containing earlier, less conscious or unconscious architecture entities

Design Process

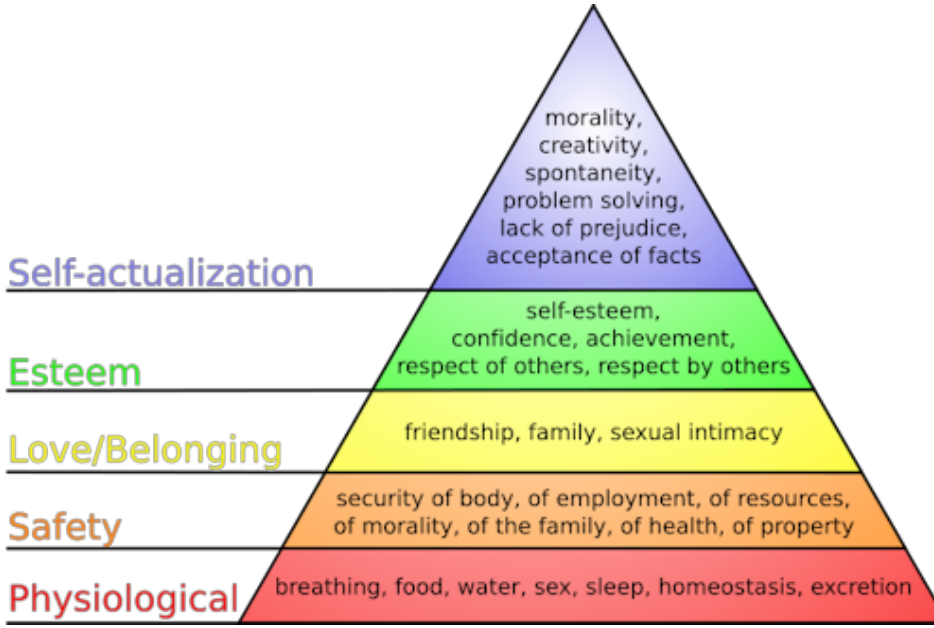
Concept *046 July 18, 2003

- 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.

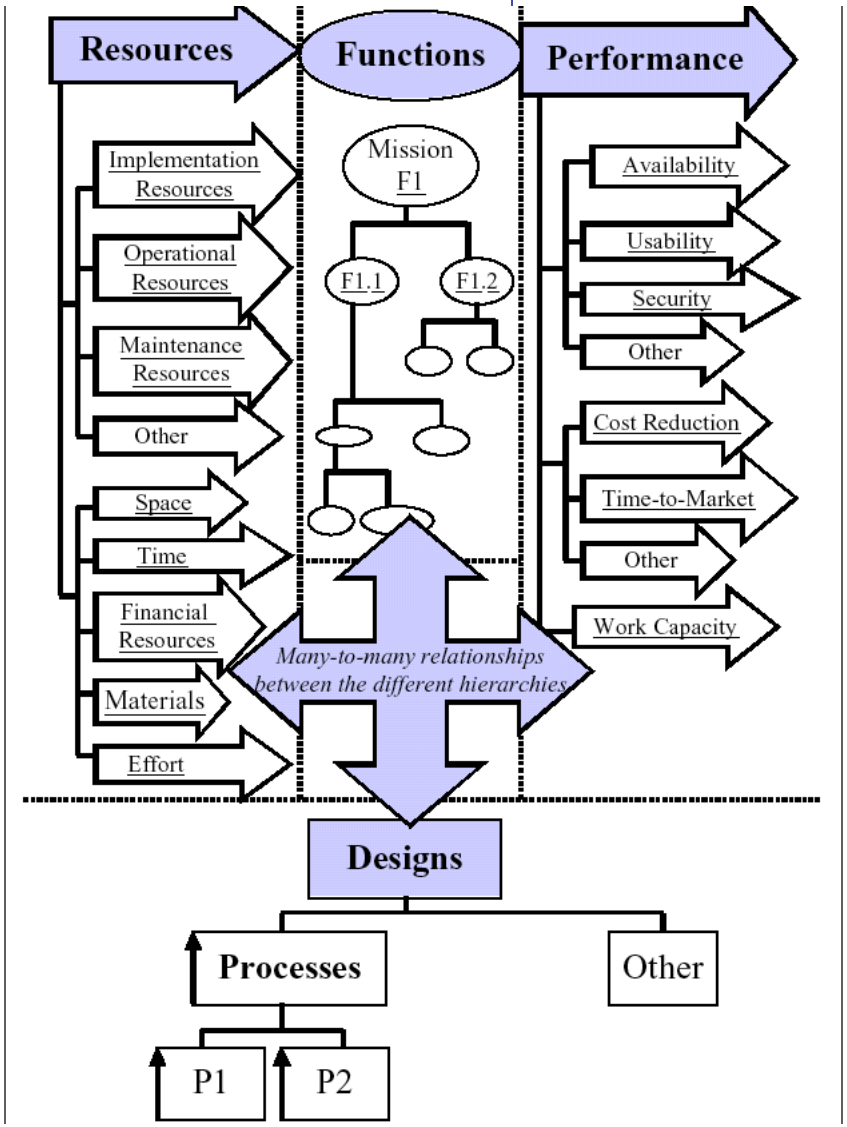
“Satisfy”: design process tries to

- ***satisfy*** is intended in the broadest sense.
- It means there is a *discernible relation* between some **design artifacts**, and some **requirements** –
- and that the purpose, intent, or at least actual effect of the design artifacts is
  - to some degree
  - to impact some performance levels, in the direction of goals,
  - and/or to avoid violating or threatening some constraints.
- There is **no notion of full satisfaction** or optimization implied or intended here.
- The degree of satisfaction actually delivered will be **limited** by priorities, resources and technology.
  - And the satisfaction will vary in time, as requirements change, and the system environment changes



System:

- the “system” is
  - any arbitrarily delineated system
  - or sub-system
  - that anyone chooses to
    - study
    - or deal with
    - that has requirements attached to it
      - formally and informally.



Performance:

- *the attributes of a system*
  - *which describe 'how well' its function is carried out.*
  - *One first level decomposition is into*
    - *work capacity,*
    - *quality and*
    - *savings.*

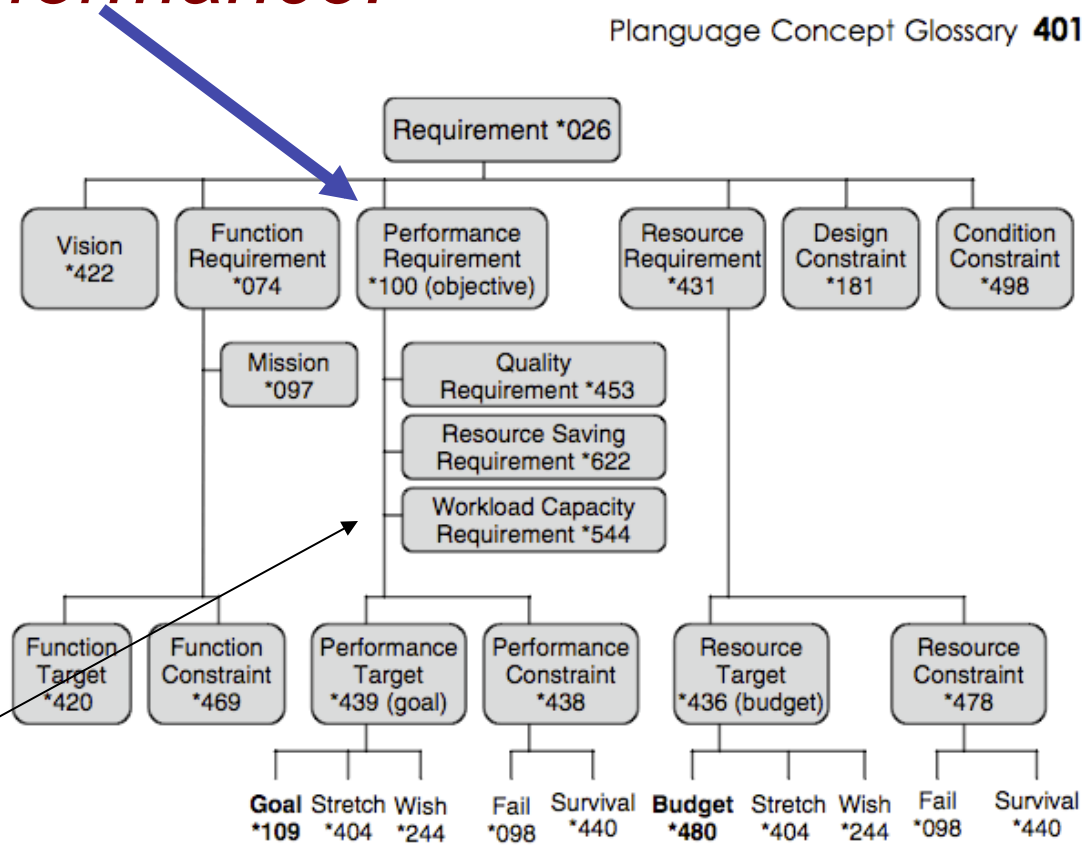
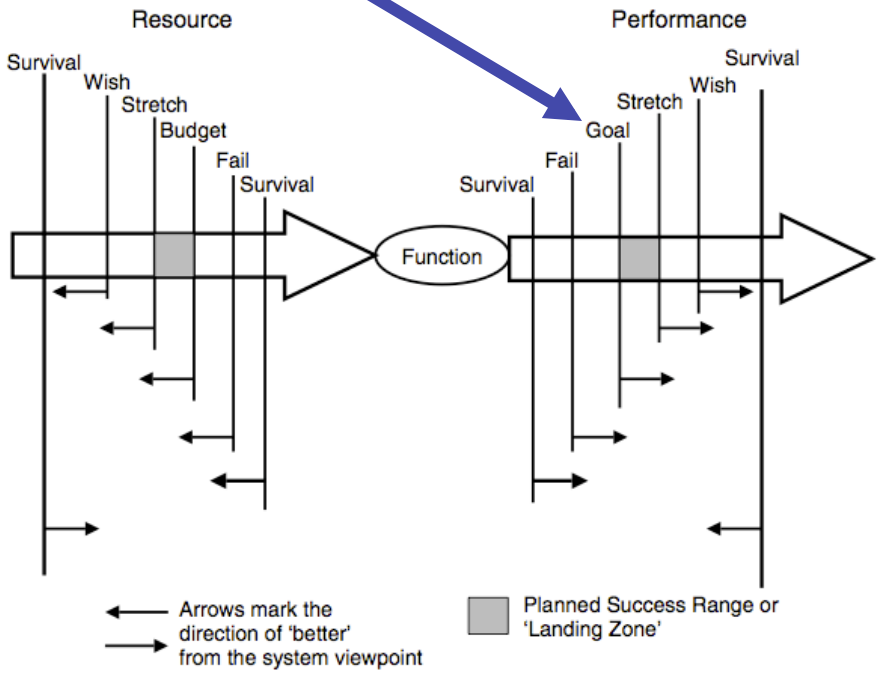


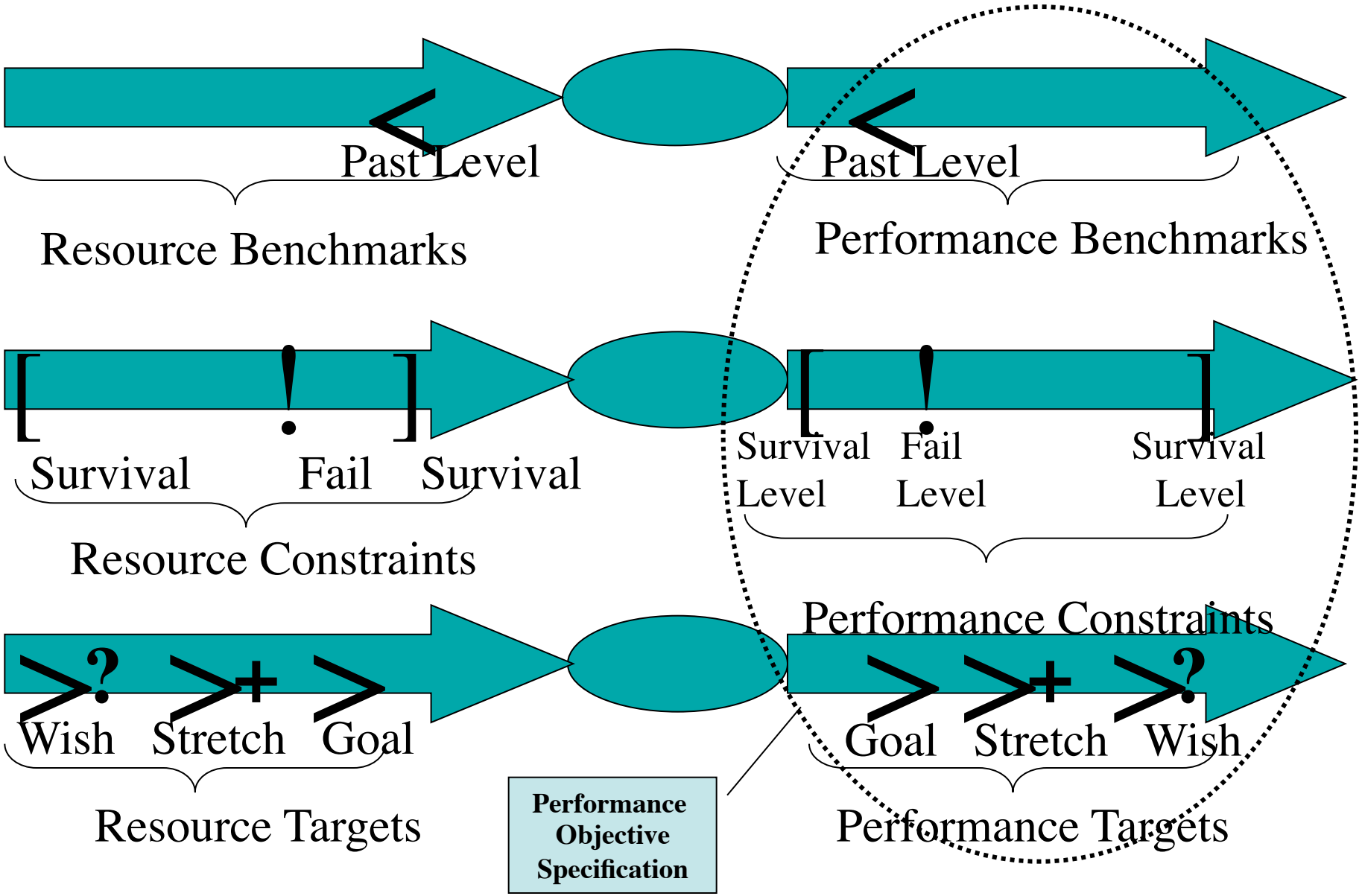
Figure G20
Requirement Concepts.

Goals:

- *goals are*
  - *levels of performance*
  - *which some set of stakeholders value and sponsor.*
- *They are*
  - *specifiable levels*
  - *on defined scales of measure.*
- *They are*
  - *the architectural basis*
  - *for judging the need for design artifacts*
    - *to control and enable*
    - *the detailed engineering of a system*
    - *to deliver to those levels*
      - *when and as needed.*

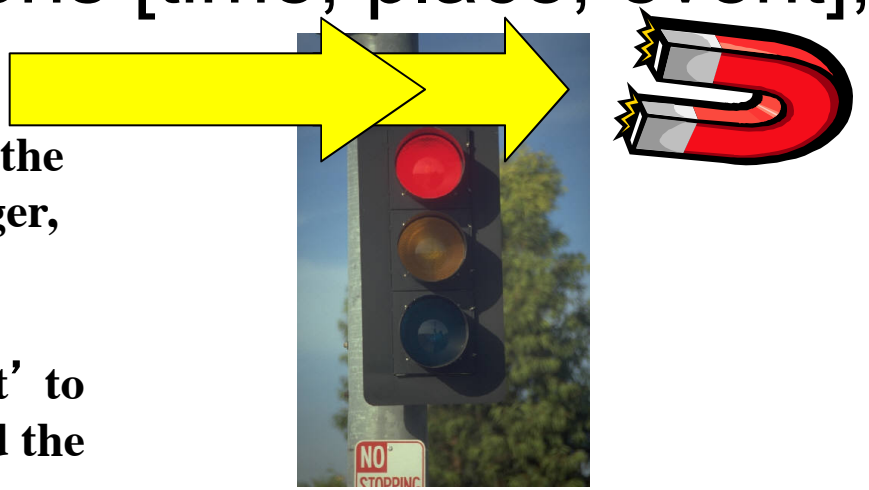


Scale Parameter Concepts



Goal (parameter): --->--------
>
Concept *109. April 7 2002

- A Goal parameter states a future, ‘sufficient’, performance or budget level requirement, on a defined Scale, under specified conditions [time, place, event], for an attribute.



A Goal acts as a magnet on the designer and project manager,

until it is reached.

Then it acts like a ‘red light’ to stop using resources beyond the Goal level

Constraints:

- ***constraints are***
  - ***any class of requirement***
  - ***which intentionally restricts the freedom***
  - ***of an architect or designer of any kind***
  - ***to select design artifacts***
    - ***either at the architectural level***
    - ***or the engineering,***
    - ***operational***
    - ***Or other life cycle levels***
      - ***(such as disposal, or maintenance).***
- ***Constraints are of several types,***
  - ***and few are absolute***
  - ***all can be judged for their relative priority and traded off.***
- ***The major types of constraints are***
  - ***resource budgets (including budgeted levels and worst case levels)***
  - ***performance constraints (worst acceptable levels of any performance attribute)***
  - ***restrictions (things the system must not do)***
  - ***demands (things the system must do)***
  - ***design constraints (any restrictions regarding design which are inputs to a given level of architecture).***

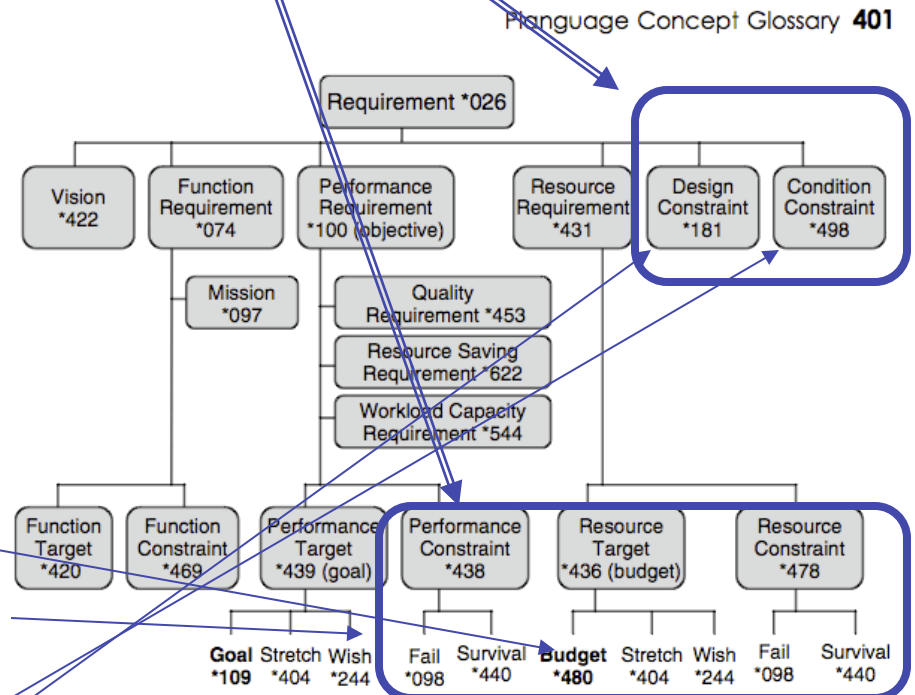


Figure G20
Requirement Concepts.

"Constrain"

- *means that the requirements,*
  - *– if known or perceived in any way,*
  - *– limit the ability of the architect to choose design artifacts,*
  - *– and impose upon the architect*
    - *the necessity of designing artifacts*
    - *which limit the ability of other design engineers*
    - *to avoid satisfying requirements.*

“Influence”

- *means that the requirements are somehow taken into consideration,*
- *even if they are prioritized so low that their real influence is at one given moment zero.*
- *They may have the potential to be reconsidered*
  - *later and*
  - *under different circumstances.*
- *They are possibly latent later in the system life cycle.*

”Related (Engineering Decisions)”

- *these include*
  - *all other architecture and requirements decisions*
  - *decisions by any engineering specialty*
    - *or other decision-making entity*
    - *that is controllable by the architectural level of decision-making*
      - *to any degree*
      - *by any means.*
  - *Decisions made after initial system delivery*
    - *by any other entities*
    - *which can influence the attributes of the system*
    - *or some offspring of it.*
    - *These specifically include*
      - *customers,*
      - *markets,*
      - *trade associations,*
      - *license holders,*
      - *military alliances,*
      - *trade blocs*
      - *and the like.*

Engineering Decisions:

- *are decisions*
  - *by any engineering process,*
  - *scientific or art,*
  - *about any notion of design artifact*
  - *intended to influence the outcome*
  - *according to their level of requirements.*

Interesting specializations

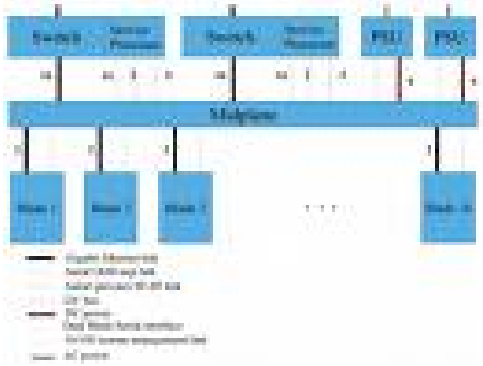
- *Perceivable Architecture: the architecture which*
  - *is somehow directly or indirectly perceivable in a real system,*
  - *as determining the range of performance and cost attributes possible.*
  - *This applies regardless of who, if anyone, consciously specified the architecture design artifacts.*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

Federal Aviation Definition [Architecture]

- ***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 these components?)***
  - ***relationships between components (How do components relate in space and time?)***
  - ***dynamic interplay description (How is control passed between and among components?)***
  - ***flows (How does data or in-process product flow in space and time?)***
  - ***resources (What resources are consumed where, in the process or system?)***
    - *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.*

IEEE definition of Architecture

- *Architecture*
  - *The organizational structure of a system or component.*
  - *Source: [IEEE 90] in [SEI-95-MM-003]*



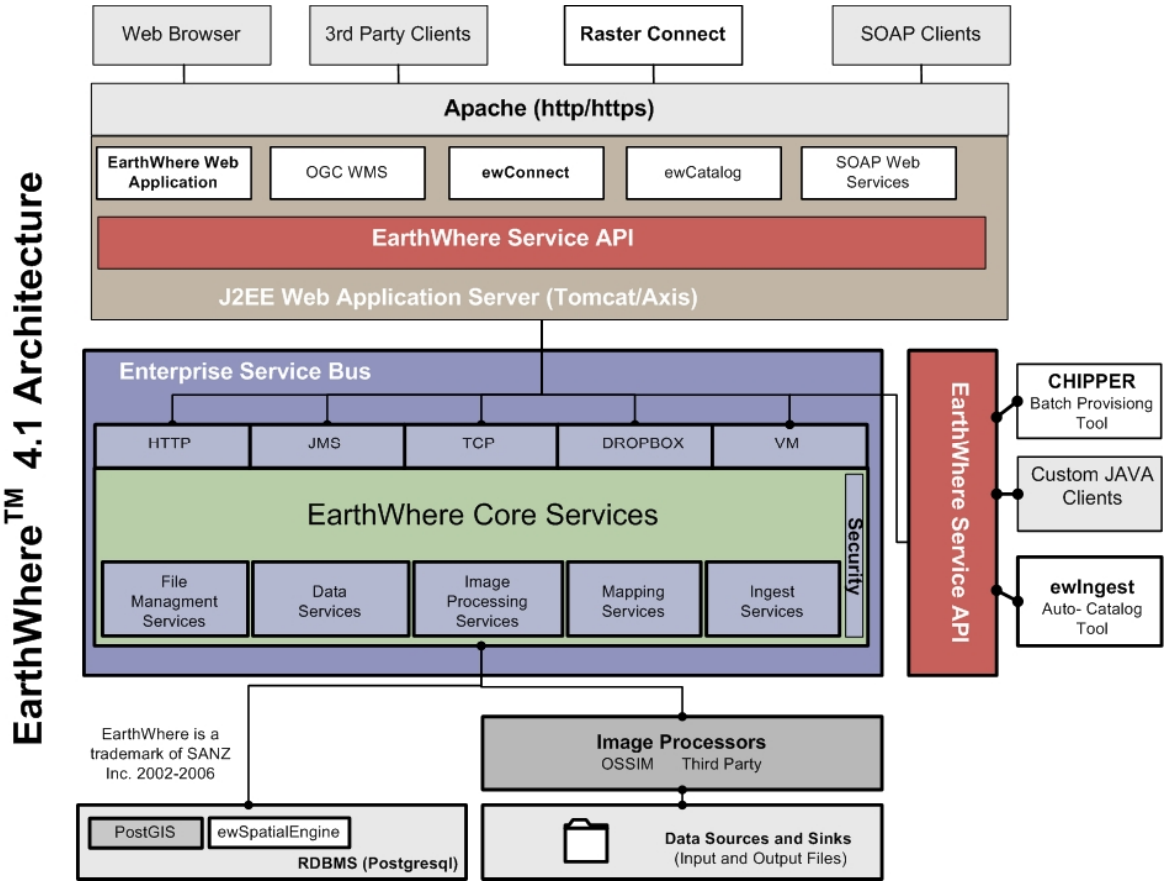
Architectural Description

Concept *618

Architectural description is

– **“a collection of products to *document* an architecture.”**

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



Architecture Specification

– **Architecture Specification**

Concept *617 June 17, 2003

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

Defining a Design/Solution/Architecture/Strategy
(Planguage, CE Design Template)
1. enough detail to estimate, 2. some impact assertion, 3. Assumptions, Risks, Issues

Orbit Application Base: (formal Cross reference Tag)

Type: Primary Architecture Option

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

Version: Nov. 30 20xx 16:49, updated 2.Dec by telephone and in meeting. 14:34

Status: Draft

Owner: Brent Barclays

Expert: Raj Shell, London

Authority: for differentiating business environment characteristics, Raj Shell, Brent Barclays(for overview)

Source: <Source references for the information in this specification. Could include people>. Various, can be done later BB

Gist: risk and P/L aggregation service, which also provides work flow/adjustment and outbound and inbound feed support. Currently used by Rates ExtraBusiness, Front Office and Middle Office, USA & UK.

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

D1: ETL Layer. Rules based highly configurable implementation of the ETL Pattern, which allows the data to be onboarded more quickly. Load and persist new data very quickly. With minimal development required. -> Business-Capability-Time-To-Market, Business Scalability

D2: high performance risk and P/L aggregation processing (Cube Building). -> Timeliness, P/L Explanation, Risk & P/L Understanding, Decision Support, Business Scalability, Responsiveness.

D3: Orbit supports BOTH Risk and P/L -> P/L Explanation, Risk & P/L Consistency, Risk & P/L Understanding, Decision Support.

D4: a flexible configurable workflow tool, which can be used to easily define new workflow processes -> Books/Records Consistency, Business Process Effectiveness, Business Capability Time to Market.

D5: a report definition language, which provides 90+% of the business logic contained with Orbit, allows a quick turnaround of new and enhanced reports with minimal regression testing and release procedure impact. -> P/L Explanation, Risk & P/L Understanding, Business Capability Time to Market, Business Scalability.

D6: Orbit GUI. Utilizes an Outlook Explorer metaphor for ease of use, and the Dxx Express Grid Control, to provide high performance Cube Interrogation Capability. -> Responsiveness, People Interchangeability, Decision Support, Risk & P/L Understanding.

D7: downstream feeds. A configurable event-driven data export service, which is used to generate feeds. -> Business Process Effectiveness

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

Assumptions: <Any assumptions that have been made>.

A1: **FCCP is assumed to be a part of Orbit.** FCxx does not currently exist and is Dec 20xx 6 months into Requirements Spec. <- Picked up by TsG from dec 2 discussions AH MA JH EC.

Consequence: FCxx must be a part of the impact estimation and costs rating.

A2: **Costs**, the development costs will not be different. All will base on a budget of say $nn mm and 3 years. The o+

costs may differ slightly, like $n mm for hardware. MA AH 3 dec

A3:Boss X will continue to own Orbit. TSG DEC 2

A4: the schedule, 3 years, will constrained to a scope we can in fact deliver, OR we will be given additional budget. If not “I would have a problem” <- BB

A5: the cost of expanding Orbit will not be prohibitive. <- BB 2 dec

A6: we have made the assumption that we can integrate Oribit with PX+ in a sensible way, even in the short term <- BB

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

D1: FCxx replaces Px+ in time. ? tsg 2.12

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

R1. FCxx is delayed. Mitigation: continue to use Pxx <- tsg 2.12

R2: the technical **integration** of Px+ is not as easy as thought & we must redevelop Oribit

R3: the and or scalability and cost of **coherence** will not allow us to meet the delivery.

R4: **scalability** of Orbit team and infrastructure, first year especially <- BB. People, environments, etc.

R5: re Cross Desk reporting Requirement, major impact on technical design. **Solution not currently known**. Risk no solution allowing us to report all P/L

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

I1: Do we need to put the fact that we own Orbit into the objectives (Ownership). MA said, other agreed this is a huge differentiator. Dec 2.

I2: what are the time scales and scope now? Unclear now BB

I3: what will the success factors be? We don’t know what we are actually being asked to do. BB 2 dec 20xx

Design Spec Enlarged 1 of 2

Spec Headers

Orbit Application Base: (formal Cross reference Tag)

Type: Primary Architecture Option

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

Version: Nov. 30 20xx 16:49, updated 2.Dec by telephone and in meeting. 14:34

Status: Draft (PUBLIC EXAMPLE EDIT)

Owner: Brent Barclays

Expert: Raj Shell, London

Authority: for differentiating business environment characteristics, Raj Shell, Brent Barclays(for overview)

Source: <Source references for the information in this specification. Could include people>. Various, can be done later BB

Gist: risk and P/L aggregation service,

which also provides work flow/ adjustment and outbound and inbound feed support. Currently used by Rates Extra Business, Front Office and Middle Office, USA & UK.

Detailed Description and -> Impacted Objectives

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

D1: ETL Layer. Rules based highly configurable implementation of the ETL Pattern, which allows the data to be onboarded more quickly. Load and persist new data very quickly. With minimal development required. -> Business-Capability-Time-To-Market, Business Scalability

D2: high performance risk and P/L aggregation processing (Cube Building). -> Timeliness, P/L Explanation, Risk & P/L Understanding, Decision Support, Business Scalability, Responsiveness.

D3: Orbit supports BOTH Risk and P/L -> P/L Explanation, Risk & P/L Consistency, Risk & P/L Understanding, Decision Support.

D4: a flexible configurable workflow tool, which can be used to easily define new workflow processes -> Books/Records Consistency, Business Process Effectiveness, Business Capability Time to Market.

D5: a report definition language, which provides 90+% of the business logic contained with Orbit, allows a quick turnaround of new and enhanced reports with minimal regression testing and release procedure impact. -> P/L Explanation, Risk & P/L Understanding, Business Capability Time to Market, Business Scalability.

D6: Orbit GUI. Utilizes an Outlook Explorer metaphor for ease of use, and the Dxx Express Grid Control, to provide high performance Cube Interrogation Capability. -> Responsiveness, People Interchangeability, Decision Support, Risk & P/L Understanding.

D7: downstream feeds. A configurable event-driven data export service, which is used to generate feeds -> Business Process Effectiveness, Business Capability Time to Market.

Design Spec Enlarged 2 of 2

==== Priority & Risk Management =========

Assumptions*: <Any assumptions that have been made>.*

A1: **FCCP is assumed to be a part of Orbit.** FCxx does not currently exist and is Dec 20xx 6 months into Requirements Spec. <- Picked up by TsG from dec 2 discussions AH MA JH EC.

Consequence: FCxx must be a part of the impact estimation and costs rating.

A2: **Costs**, the development costs will not be different. All will base on a budget of say $ nn mm and 3 years. The ops costs may differ slightly, like $n mm for hardware. MA AH 3 dec

A3:Boss X will continue to own Orbit. TSG DEC 2

A4: the schedule, 3 years, will constrained to a scope we can in fact deliver, OR we will be given additional budget. If not “I would have a problem” <- BB

A5: the cost of expanding Orbit will not be prohibitive. <- BB 2 dec

A6: we have made the assumption that we can integrate Oribit with PX+ in a sensible way, even in the short term <- BB

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

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

R1. FCxx is delayed. Mitigation: continue to use Pxx<- tsg 2.12

R2: the technical **integration** of Px+ is not as easy as thought & we must redevelop Oribit

R3: the and or scalability and cost of **coherence** will not allow us to meet the delivery.

R4: **scalability** of Orbit team and infrastructure, first year especially <- BB. People, environments, etc.

R5: re Cross Desk reporting Requirement, major impact on technical design. **Solution not currently known**. Risk no solution allowing us to report all P/L

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

I1: Do we need to put the fact that we own Orbit into the objectives (Ownership). MA said, other agreed this is a huge differentiator. Dec 2.

I2: what are the time scales and scope now? Unclear now BB

I3: what will the success factors be? We don’t know what we are actually being asked to do. BB 2 dec 20xx

I4: for the business other than flow options, there is still a lack of clarity as to what the requirements are and how they might differ from Extra and Flow Options. BB

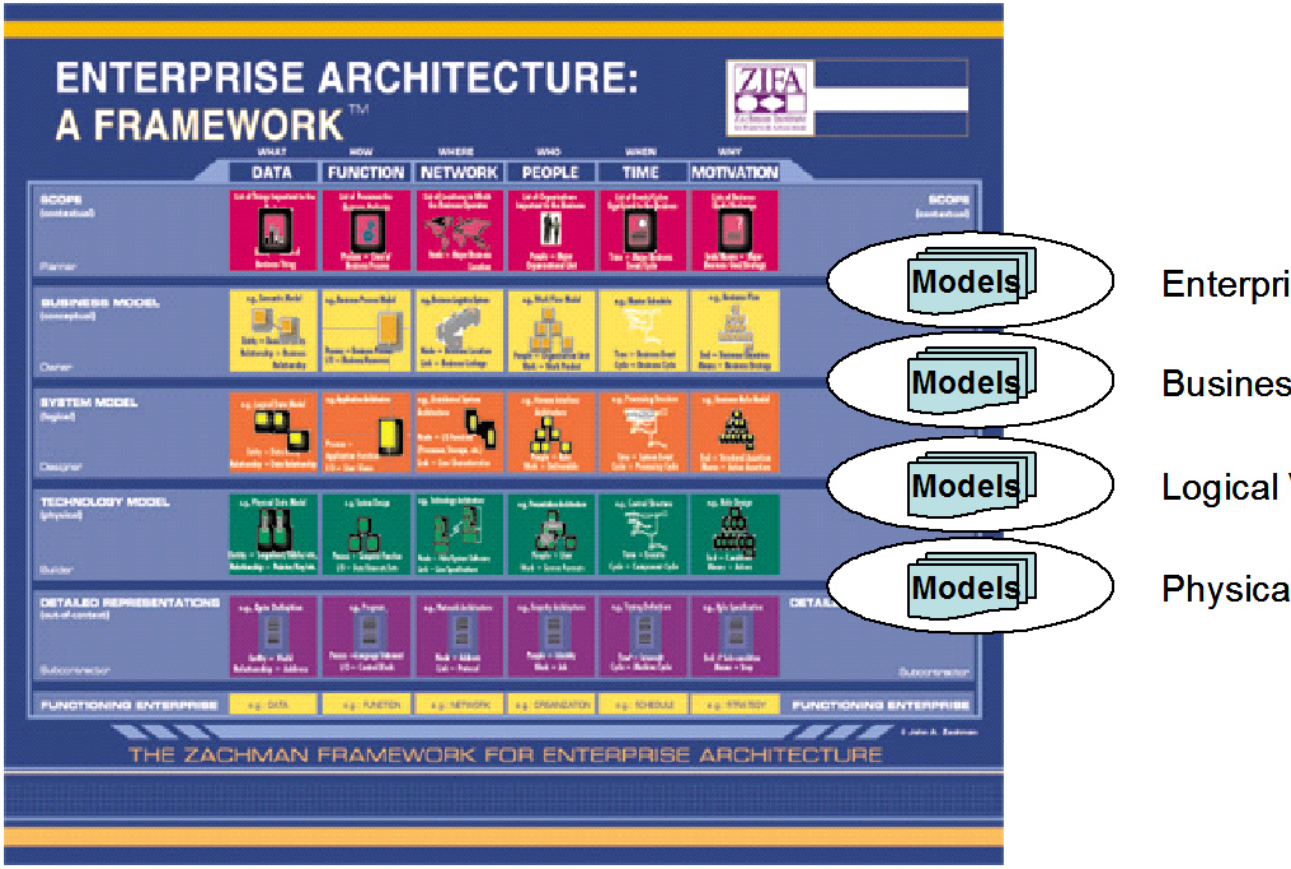
I5: the degree to which this option will be seen to be

Systems Architect

– **Systems Architect**

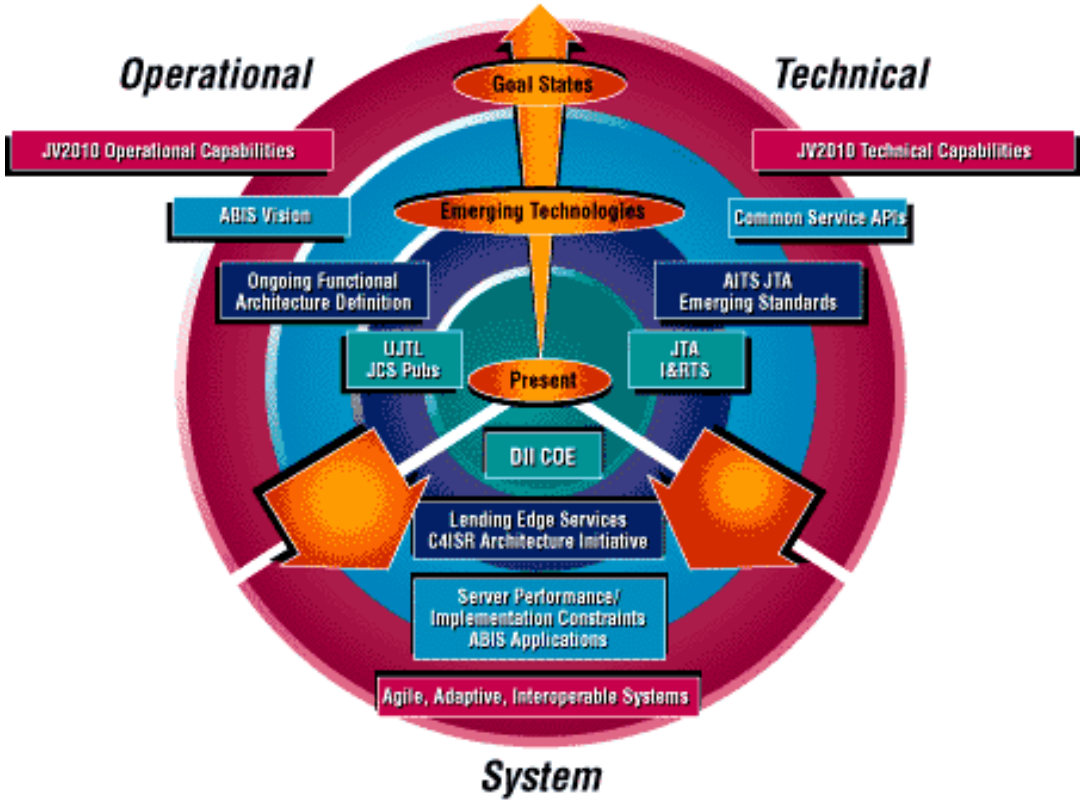
Concept *193 May 6, 2003

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



Systems Architecture

- **Systems Architecture**
  - **Concept *564 May 28, 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.

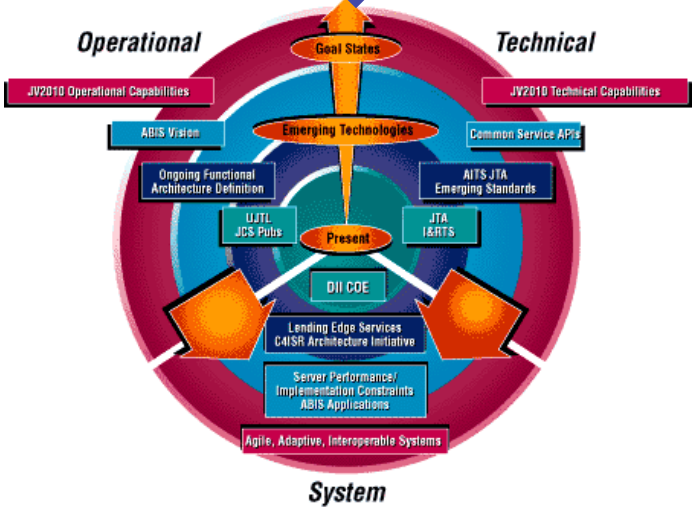


Systecture

– **Systecture** **© Gilb**

Concept *564 May 27, 2003

- See Systems Architecture *564.
- Systecture is
  - a conjunction of the term
  - *'system architecture'.*



Systect

- **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).