Grunnleggende Feil med IT Arkitektur: Vi må kunne være mye mer systematisk, logisk, faktabasert og kostnadsbevisst

Onsdag 11.2.2015 Oslo
Software 2015 Konferanse, Radisson BLU Konferansesenter
13:20 - 13:40 (20 minutter)
Rom 1

Tom Gilb
Tom@Gilb.com, www.Gilb.com, @ImTomGilb
Forfatter av 'Competitive Engineering'
Kolbotn, Norge

The architecture is there to satisfy requirements

Mine ‘Arkitektur Ingeniørfag’ Prinsipper(9 Feb 2015)

1. Arkitektur er en ’forhåpning’ som må påvises i praksis.

2. Dine samlede arkitektur forslag har som hensikt å tilfredsstille mange egenskapsmål, innenfor flere begrensninger.

3. Det presise utfall i egenskapsnivåer, vil bero på dine spesifikasjoner, den virkelig implementasjon i praksis, og det miljø som benyttes.

4. Arkitekten er selv ansvarlig for utfallet av sine spesifikasjoner, dersom de følges nøye.

5. Arkitekten er selv ansvarlig for å planlegge, slik at man unngår eller styrer risikoene for svik i egenskaper og ressursbruk.

6. IT Arkitekten er ansvarlig for alle egenskaper og kostnader på system nivå, ikke kun på software.

7. Arkitekten er ansvarlig for et tilstrekkelig detaljnivå for å kunne styre andre, og garantere sluttresultater.

8. Arkitekten er ansvarlig for åpenhet og planlegging for alle risikoer og usikre momenter, både i opprinnelig planlegging, og ved vedlikehold og forandring i eksisterende systemer.

9. Arkitekten er ansvarlig for å holde seg til erfaringsmessig sikre teknologier, slik at tilsiktede resultater garanteres så langt som mulig: de er ikke der for å leke eller eksperimentere – de har ikke en ’forsker’ rolle.

10. Arkitektens forslag skal bekreftes ved praktisk måling, så tidlig og ofte som mulig: isærnye og risikofylte forslag må måles tidlig mht alle kritiske egenskaper og side effekter, slik at de kan erstattes tidlig ved svikt.

.

Forenklet

- Arkitekten bærer hele ansvaret for resultatene,
  - uansett hva som skjer i konstruksjon, endring og drift

Oslo Opera House requirements

- Qualities
- Costs
- Constraints

Oslo Opera House requirements (imagined, for example)

- 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

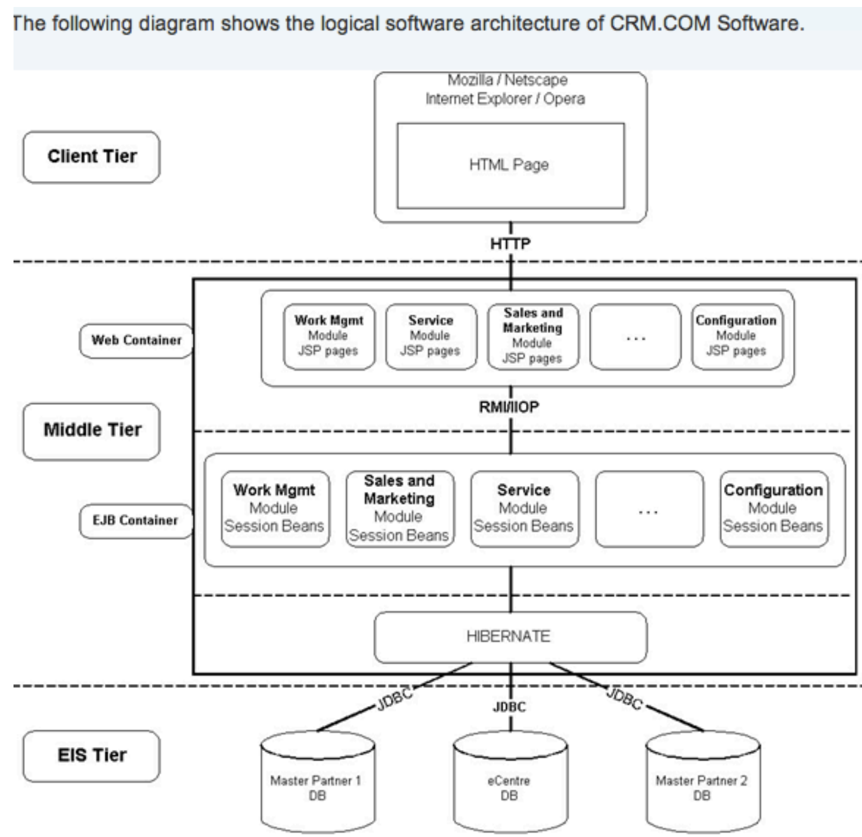
Pseudo Architecture Does not mention goals and constraints

‘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

2 '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.

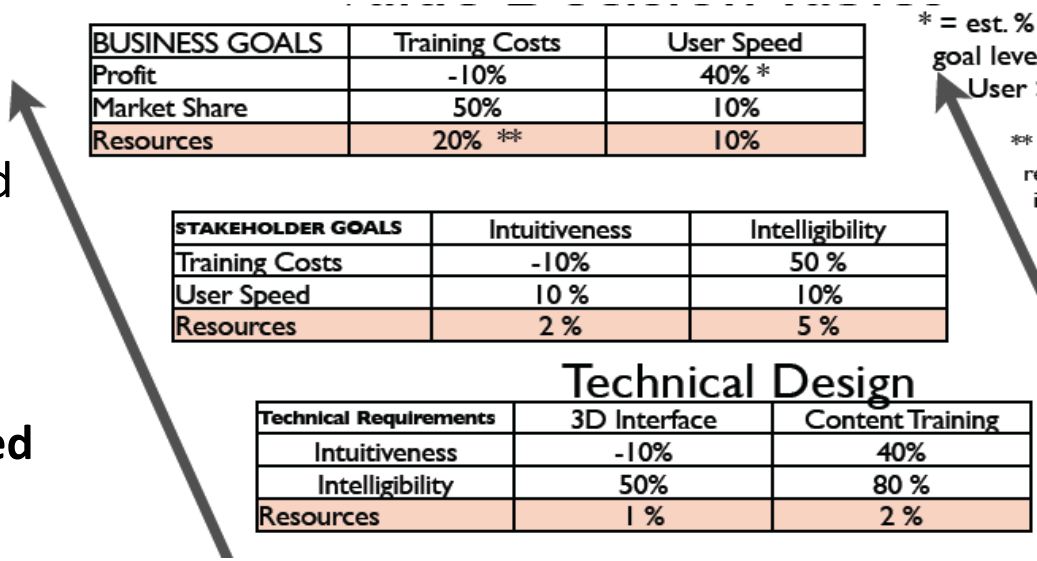
- System Architecture should contain **goals/requirements** artifacts, and structure and behavior artifacts **based on** those goals.

<u>Real</u> Architecture diagram

Kai Gilb, 'Bring' Case

3 levels of organisation connected by impact numbers

(highly simplified)



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

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

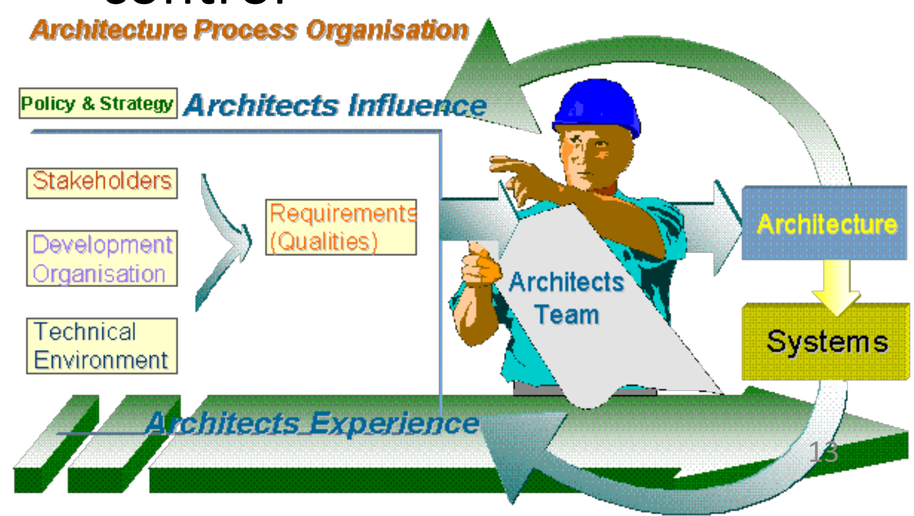
Why are these Architecture essentials, essential?

Why?

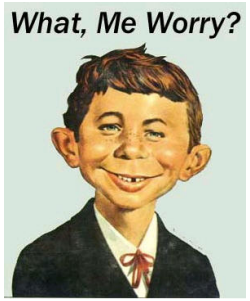
- Failure to reach even one 'critical' objective can mean total system failure
  - Example: reliability
- Failure to respect even a single constraint can mean total system failure
  - Example: cost

And if they are missing…

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



What a Difference



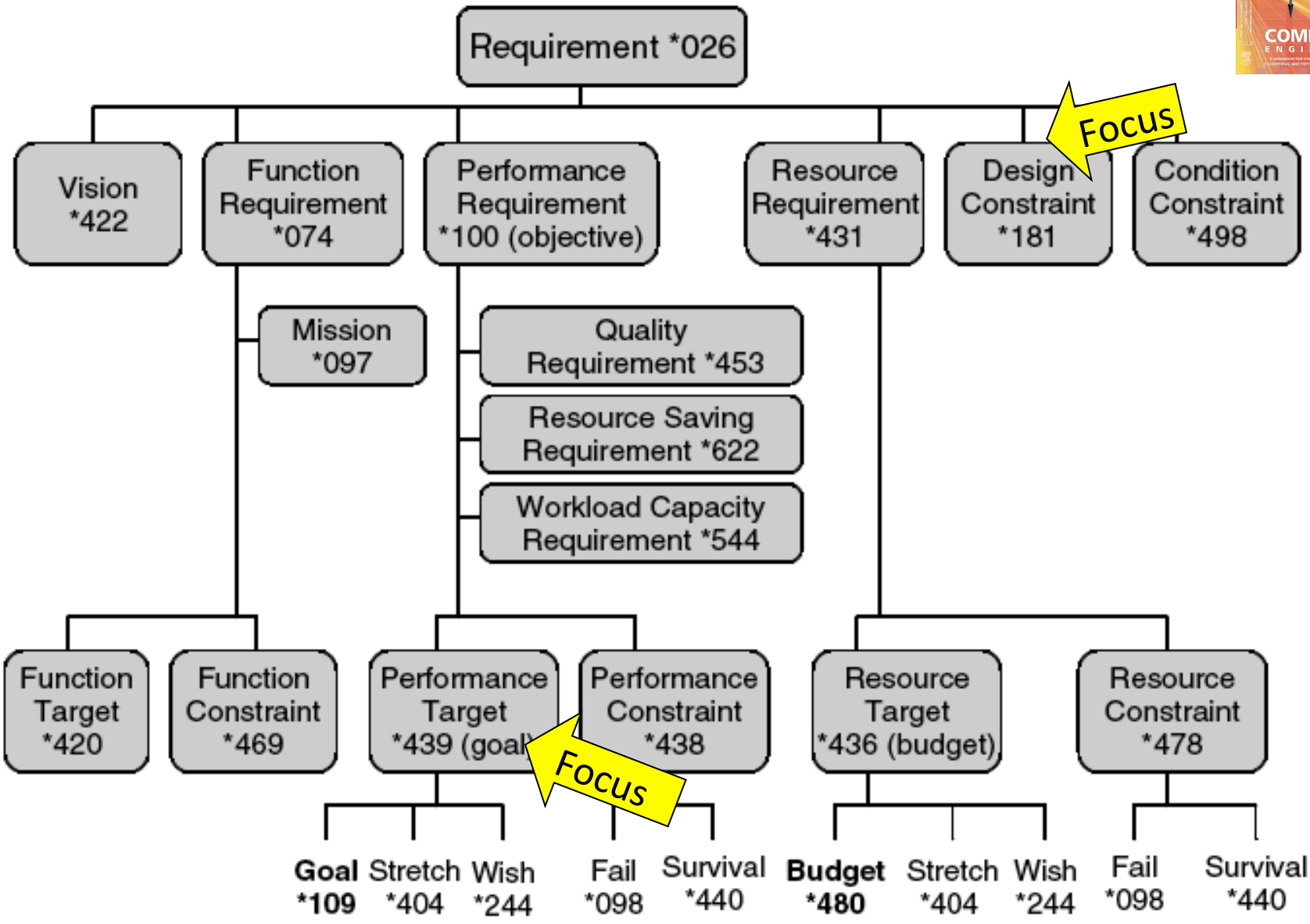
A Real Architect

- **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**

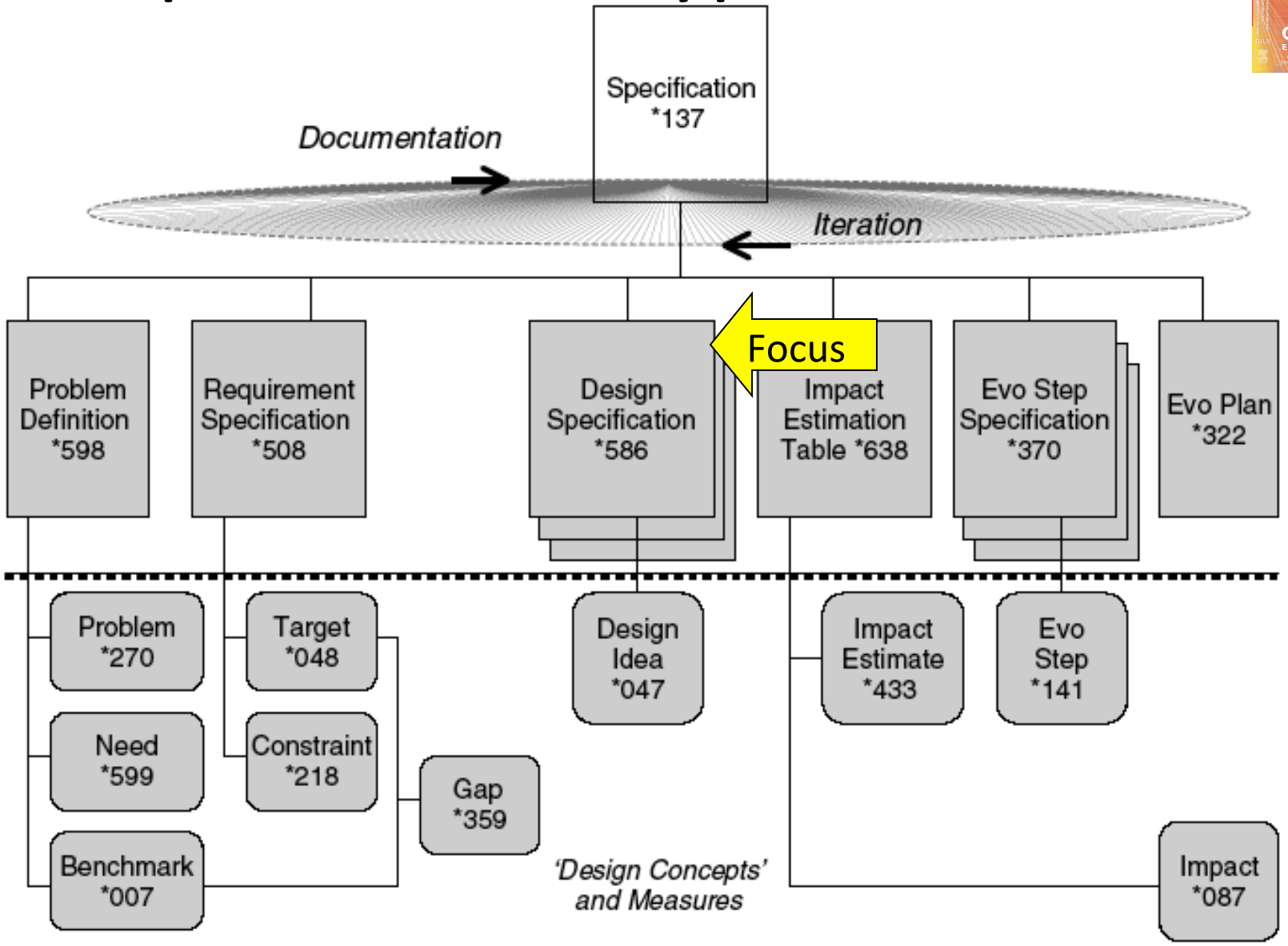
A False Architect

- **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 responsibility for the performance and quality attributes or costs of their suggested architecture: no skin in the game.**

Requirement Concepts for Architects



Specification Types for Architects



Architecture Specification Rules

se http://www.gilb.com/dl60 for praktiske eksempler på fortolkning 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.

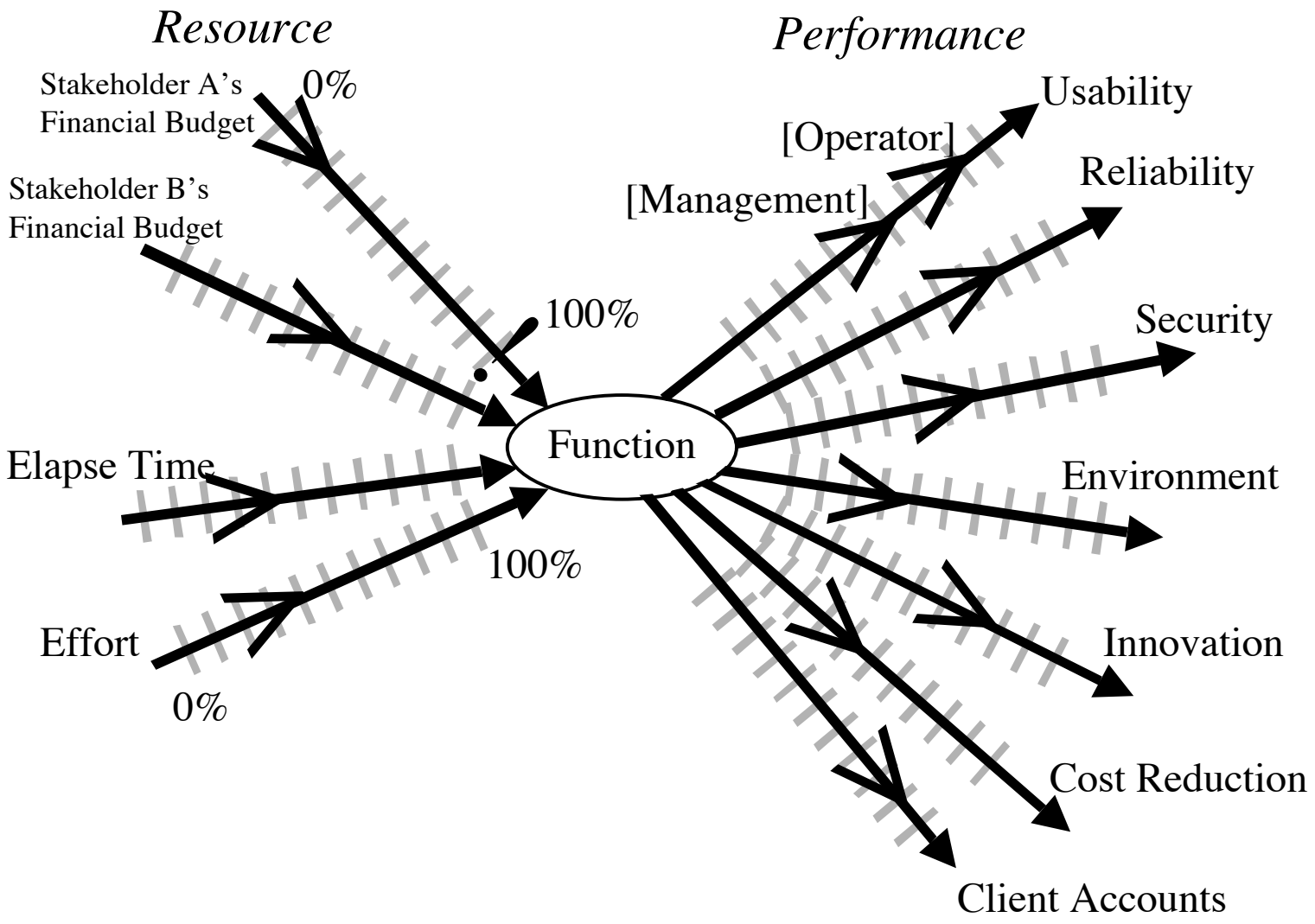
R5: Impacts: For each design idea, specify at least one main performance attribute impacted by it. Use an impact arrow '->' or the Impacts parameter.

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



My Personal Definition

http://www.gilb.com/tiki-download_file.php?fileId=47

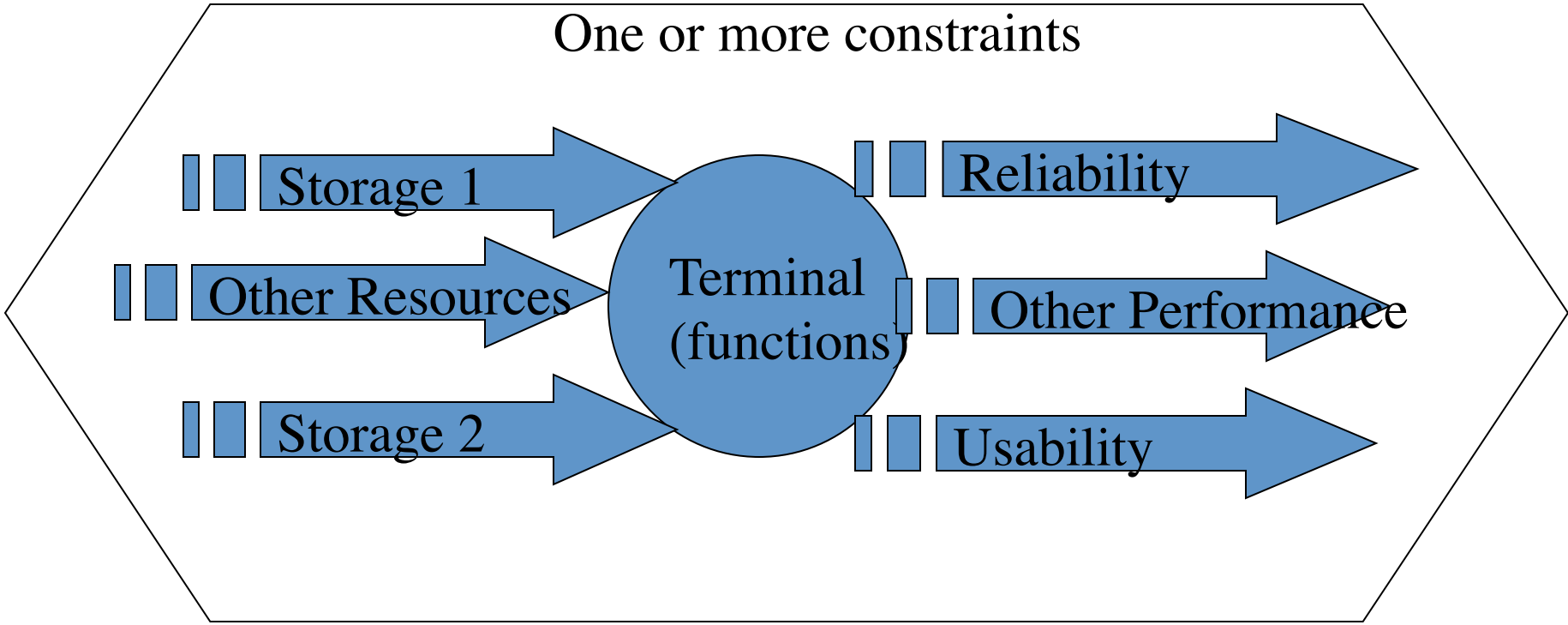
- **Architecture (collective noun):**
  - Concept *192. May 9 2005 (detail in ppt note!)

- 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.

- Et arkitektur er
- Virkelige ting og spesifikasjoner som påvirker systemegenskaper direkte eller indirekte

Evo and Requirements, Conceptually

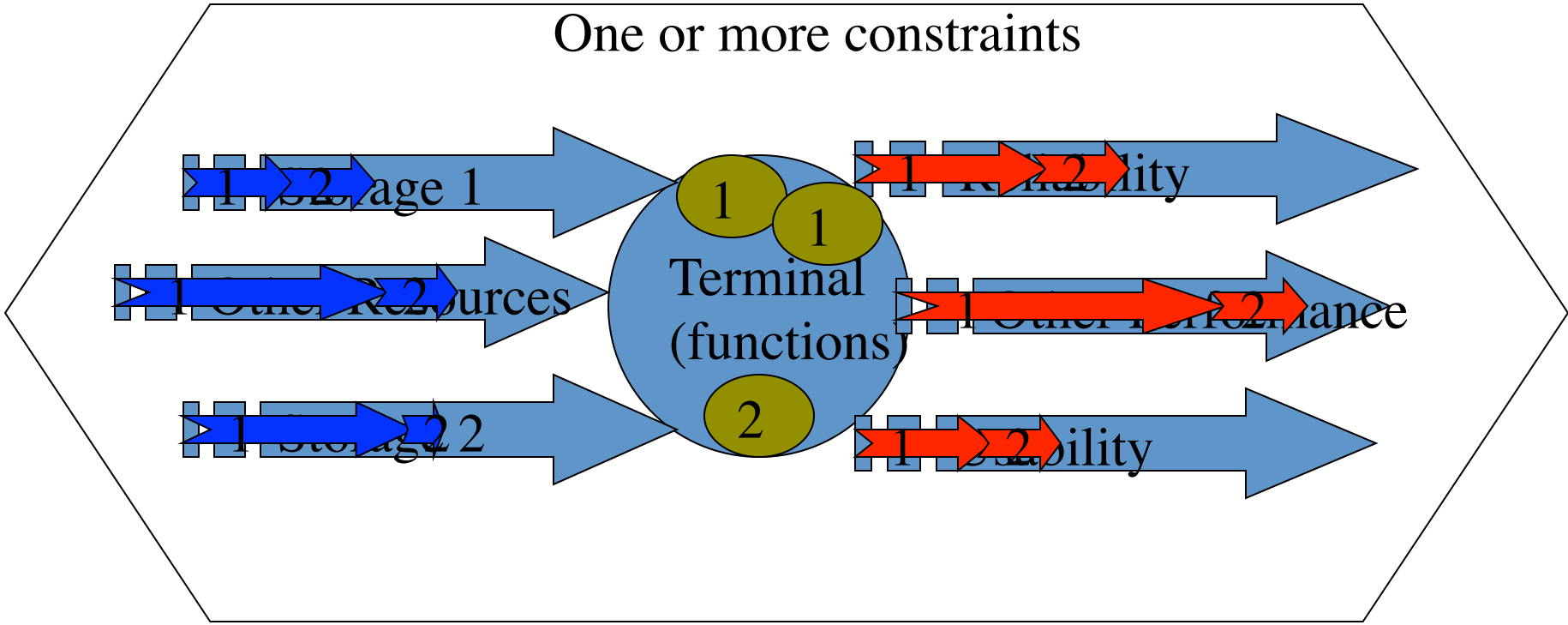
Requirements are the framework for Evo development



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



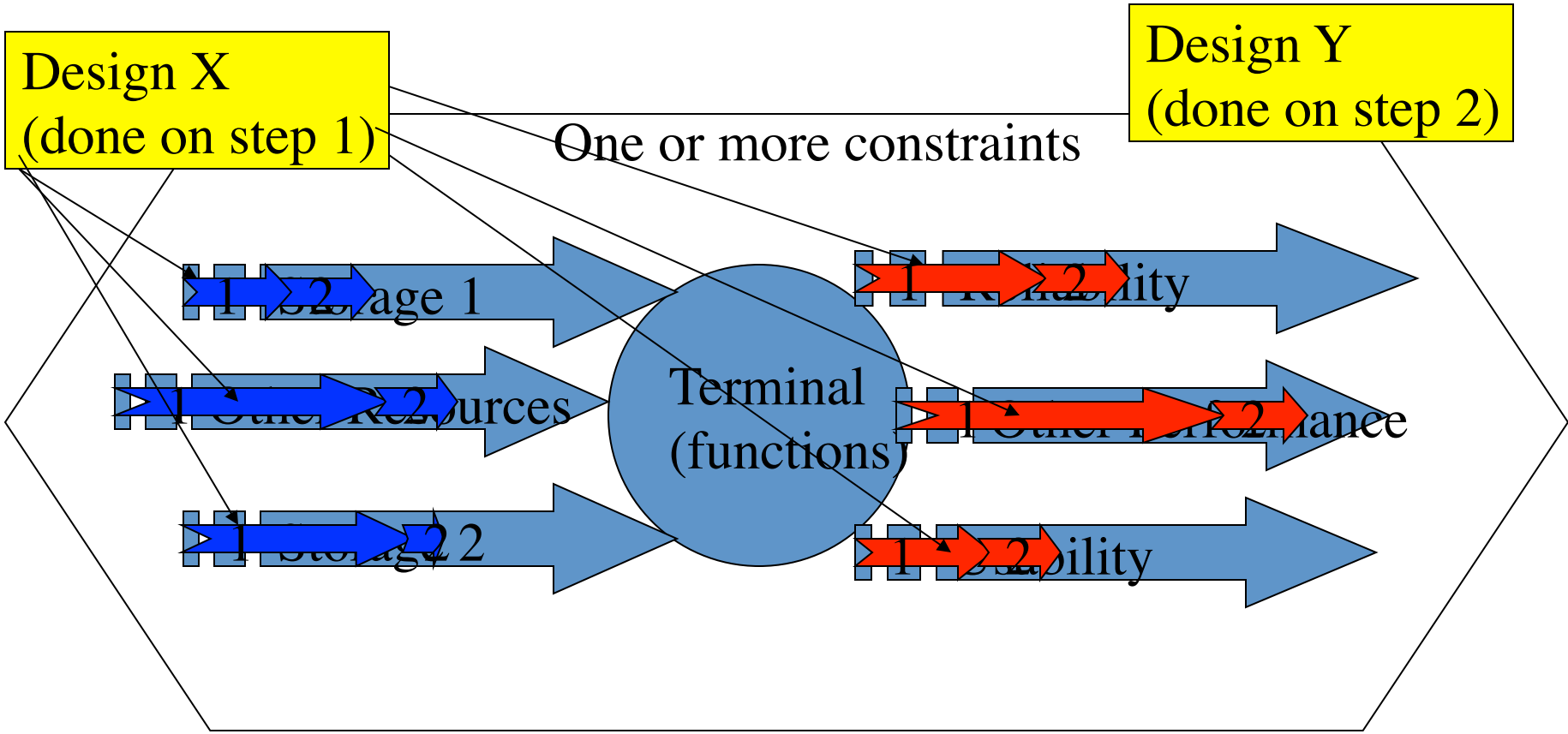
Evo development gradually delivers function and performance, while eating up resources

n

n

n

'Design' is what delivers performance, and costs resource



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

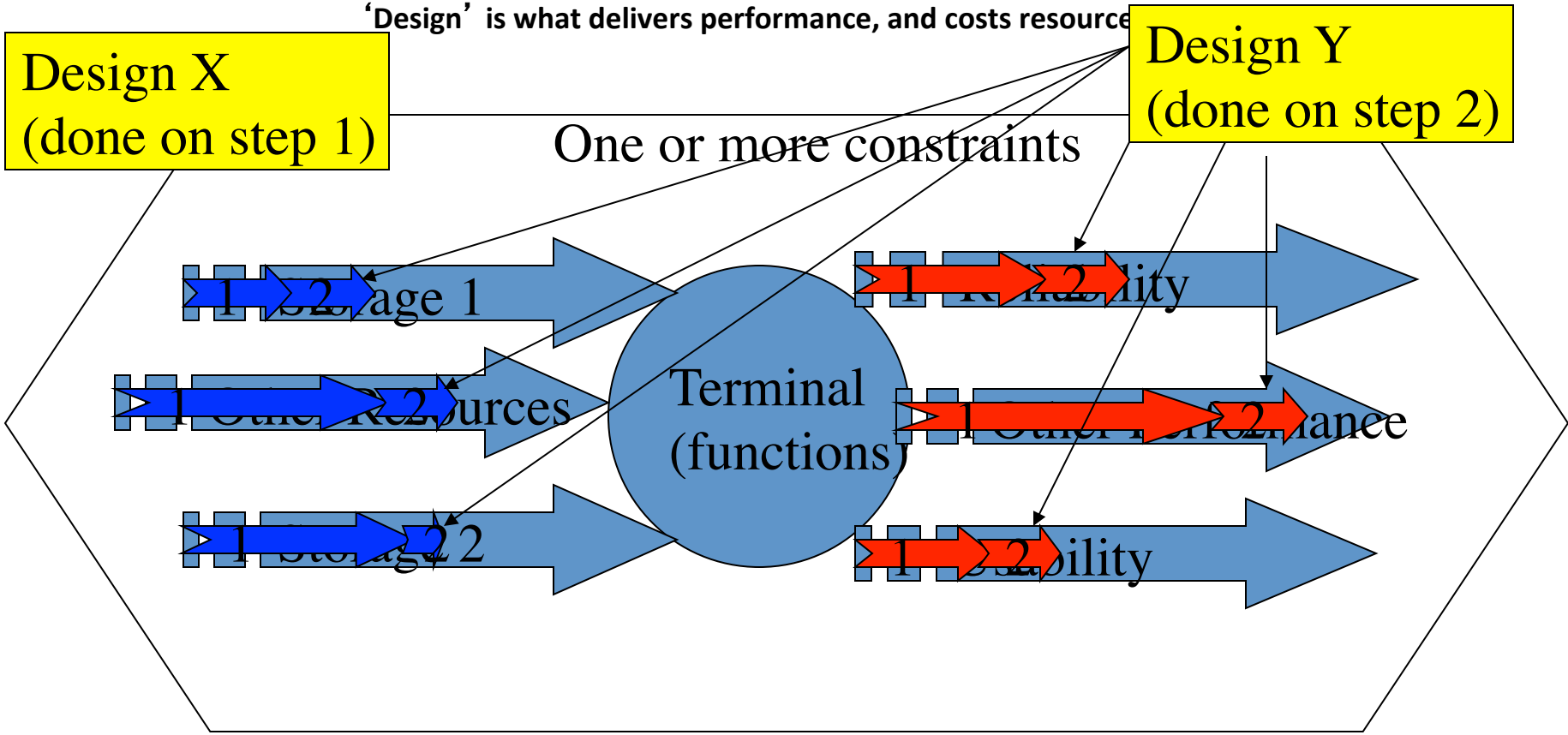
n

n

Design _
(done on step n)

Evo and Requirements, Conceptually

'Design' is what delivers performance, and costs resource



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

n

n

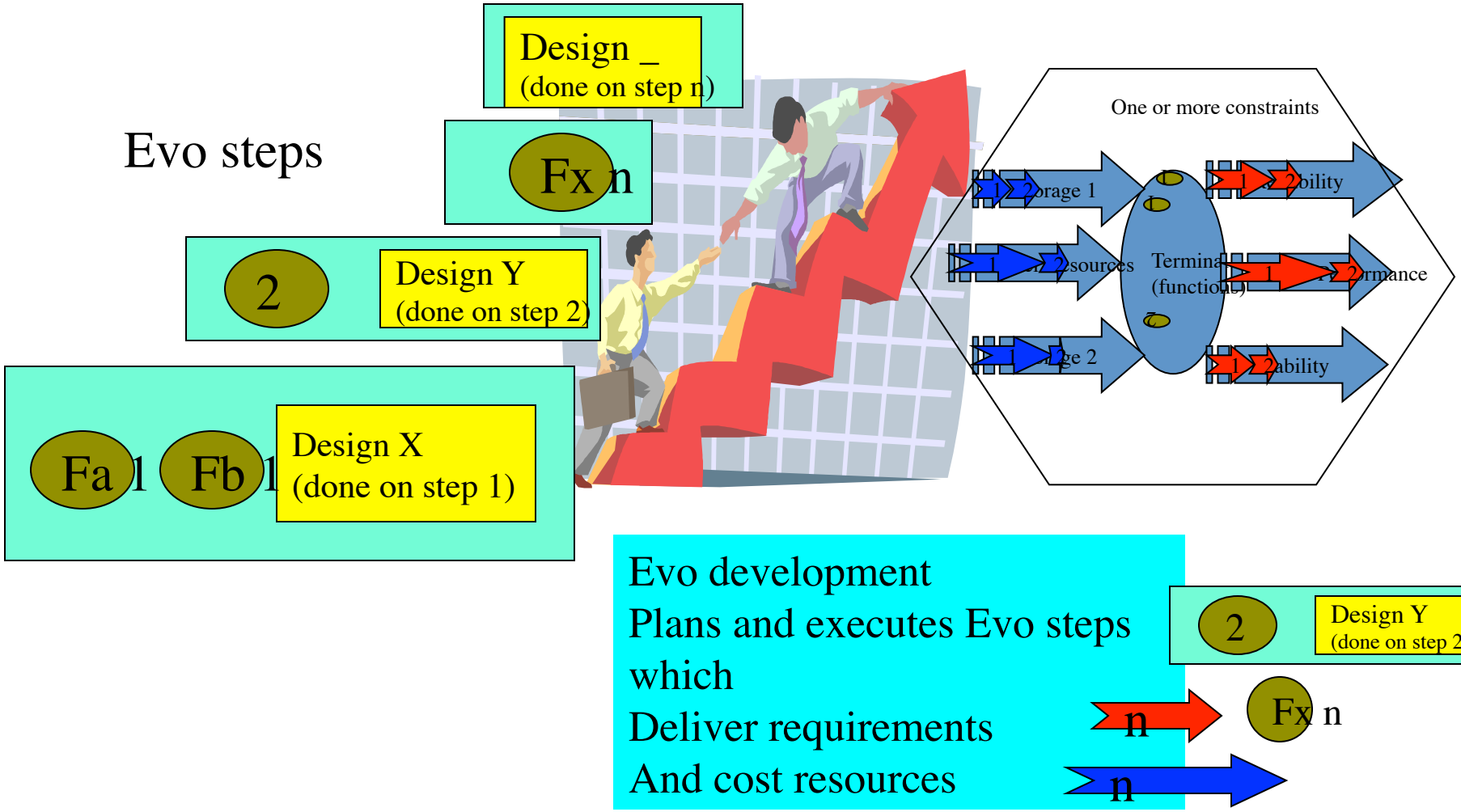
Design _
(done on step n)

Evo and Requirements, Conceptually

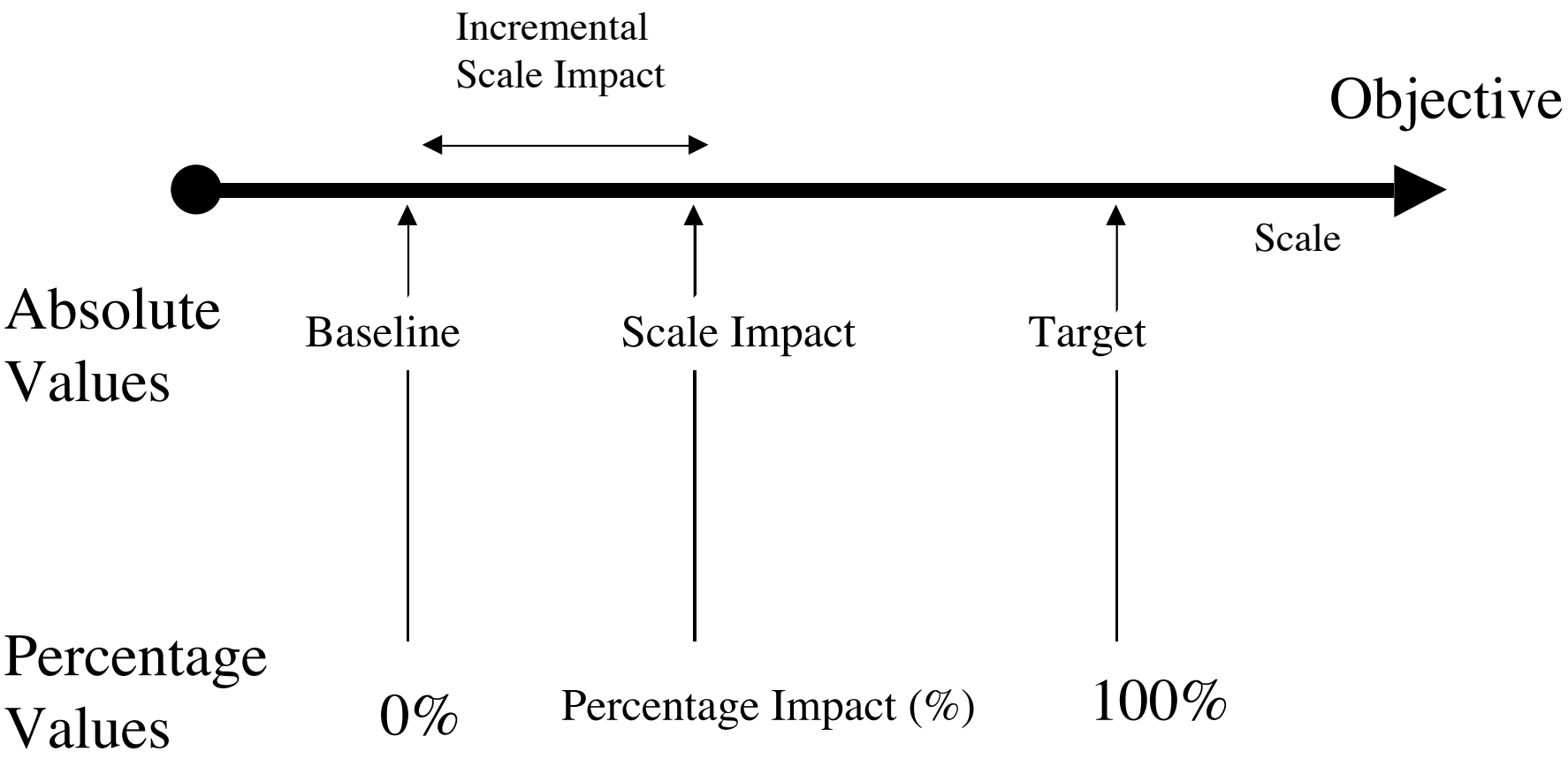
'Design' is what 'delivers performance', and 'costs resource'

Function is selected or built to deliver more function

Evo steps are packages of either function and/or design



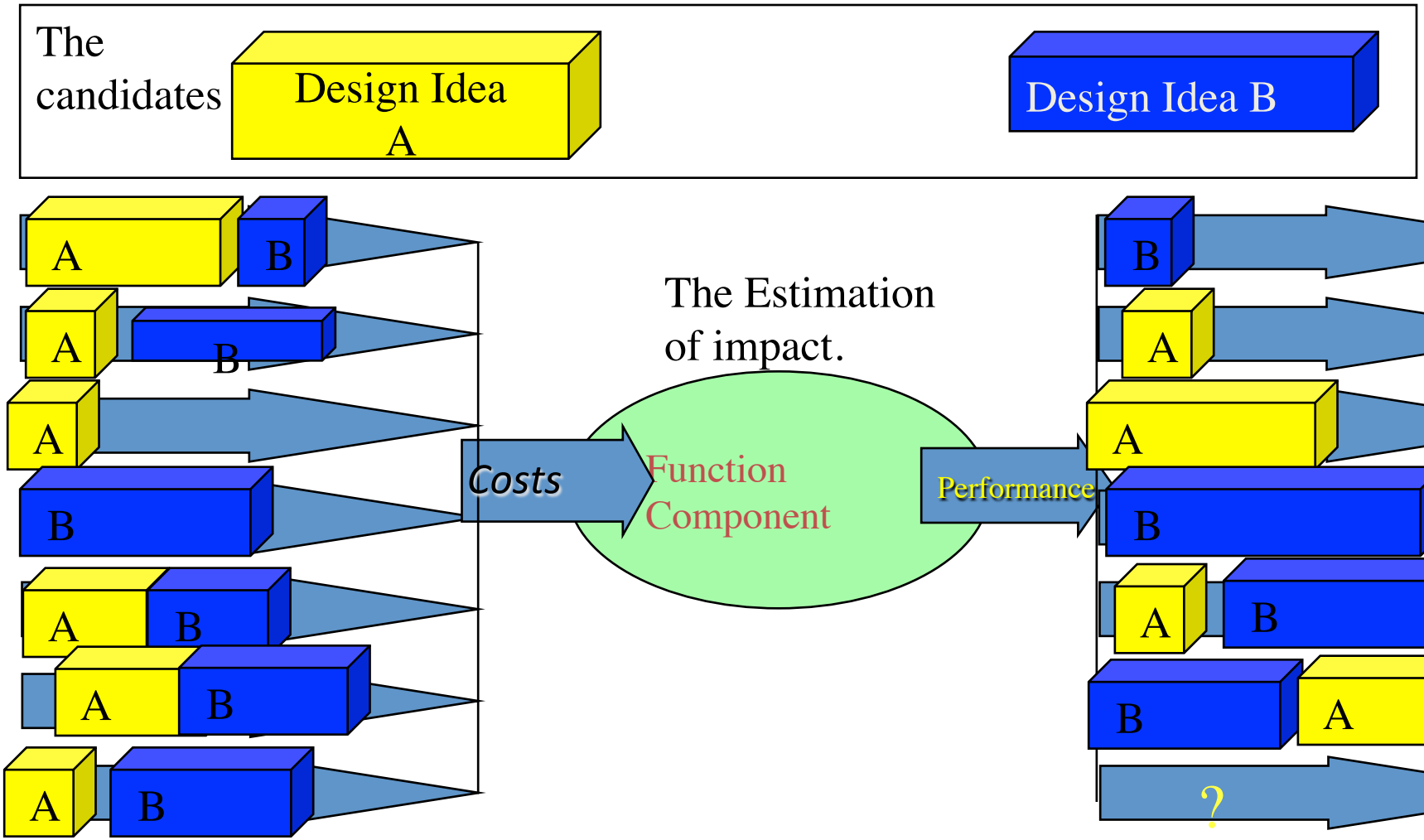
Impact Estimation Basic Concepts



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	:6 mo	X		X	X
Mid-range	Min BoM for The Corp phone	<$9	3	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		6	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	>2 yrs			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

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%	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]%	5%	10%	5%	5%	0%	0%
Platformisation Technology	25%	10%	30%	0%	[illegible]	10%	0%	5%	0%	10%	0%	5%
Interface	5%	15%	15%	0%	5%	0%	5%	0%	0%	10%	0%	10%
Operator preference	0%	[illegible]	[illegible]	[illegible]	[illegible]	20%	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%	[illegible]	[illegible]	0%	0%	40%	0%	0%	0%	5%	20%	5%
Competitiveness	10%	15%	20%	0%	10%	20%	10%	10%	20%	10%	10%	10%
User experience	5%	[illegible]	0%	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	£ [illegible]	£ 0.81	£ 1.21	£ 2.68	£ 0.79	£ 0.62	£ 0.60	
ROI Index (100=average)	106	359	109	33	[illegible]	148	107	10	152	202	174	

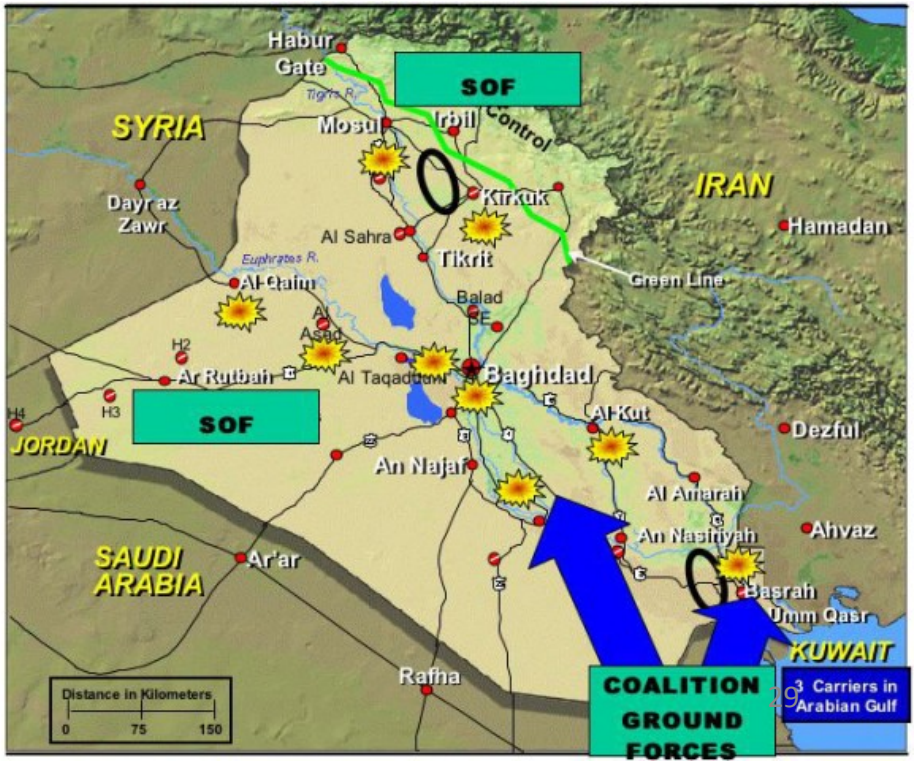
Strategy Impacts on Objectives

Cost

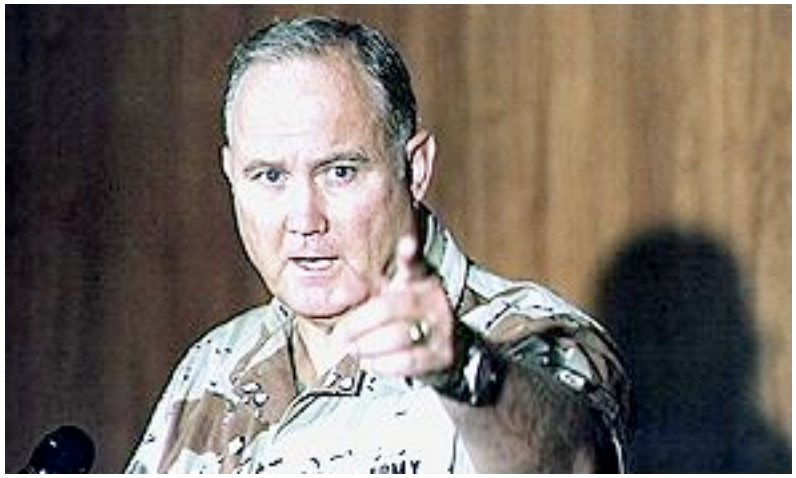
Benefit/Cost ratio

And Now A True War Story

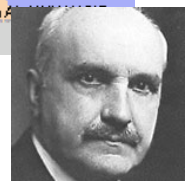
- About Why Bad IT Requirements
  - Can lose a war in Iraq
  - Or at least make it drag on for years



The Persinscom IT System Case



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

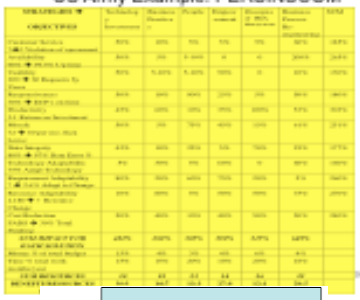


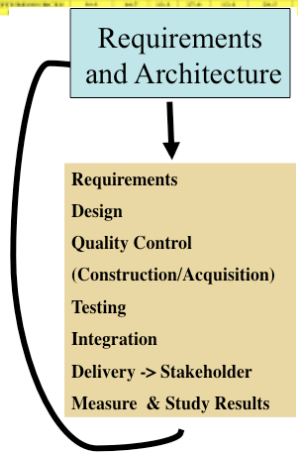
A Man Who understood that
"a bird in the hand is worth two in the Bush" <-tsg

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

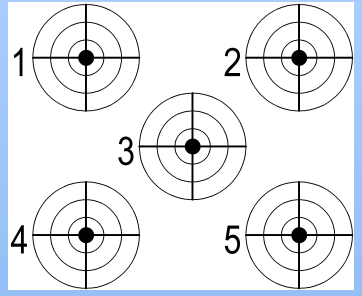
US Army Example: PERSINSCOM





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.

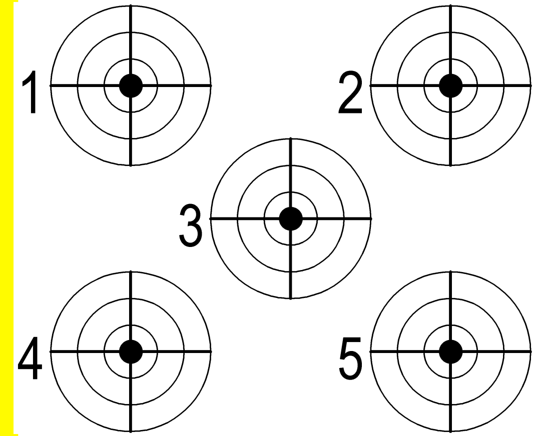
Past [Last Year] Unknown Number 🡐State of PERSCOM Management Review

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)

<u>Technology Investment</u>:

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

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'**

Gilb's Principles of Software Architecture Engineering:

(30 Sept 2013)

1. HYPOTHESIS: Architecture is a hypothesis, that needs to be proven by implementation and measurement

2. MULTIPLICITY: All architecture components will contribute a variety of attributes: *Performance, qualities, costs, and conformance-to-constraints*; to satisfy stakeholder requirements in a balanced way.

3. RESULTS: The exact architecture attributes will be determined by the exact engineering design and implementation, and the real system environment it is implemented in.

4. RESPONSIBILITY: The architect is responsible for *all* attributes of their architecture suggestions

5 .RISKS: The architect is responsible for understanding and mitigating all risks of deviation from good levels of attributes.

6. SYSTEM: The architect is responsible, for all attributes (performance, qualities, costs, constraints) at a defined *systems* level. Total system, IT system, Software System:

7. DETAIL: The architect is responsible for *defining* the architecture and its implementation and operation, to a level of detail, that ensures adequate control over the expected attributes: no blaming others for bad results.

8. HONESTY: The architect is responsible for dealing responsibly with all *unknowns and uncertainties* directly: by making them transparent and planning mitigation and alternatives; as well as following the results of implementation, of later changes, and operation,

9.KNOWNS: The architect, like other engineers, is responsible for preferring conservative architecture, that which is *known* to work: selecting innovative risk-filled architecture only when necessary.

10. PROOF: The set of architecture ideas should, as far as possible, be implemented and measured in all attributes, in an early and frequent high-value stream of components, to working systems, so that realistic assessment of effects, side effects and costs can be made; and the component adjusted or replaced if necessary without major damage.

Mine Arkitektur Ingeniørfag Prinsipper(9 Feb 2015)

1. Arkitektur er en forhåpning som må påvises I praksis.
2. Dine samlede arkitektur forslag har som hensikt å tilfredsstille flerfoldige egenskapsmål innenfor flere begrensninger.
3. Det presise utfall i egenskapsnivåer vil bero på dine spesifikasjoner, den virkelig implementasjon i praksis, og det miljø som benyttes.
4. Arkitekten er selv ansvarlig for utfallet av sine spesifikasjoner, dersom de følges nøye.
5. Arkitekten er selv ansvarlig for å planlegge slik at man unngår eller styrer risikoene for svik i egenskaper og ressursbruk.
6. IT Arkitekten er ansvarlig for alle egenskaper og kostnader på system nivå, ikke kun på software.
7. Arkitekten er ansvarlig for et tilstrekkelig detaljnivå for å kunne styre andre og garantere sluttresultater.
8. Arkitekten er ansvarlig for åpenhet og planlegging for alle risikoer og usikre momenter, både i opprinnelig planlegging, og ved vedlikehold og forandring i eksisterende systemer.
9. Arkitekten er ansvarlig for å holde seg til erfaringsmessig sikre teknologier, slik at tilsiktede resultater garanteres så langt som mulig: de er ikke der for å leke eller eksperimentere – de har ikke en 'forsker' rolle.
10. Arkitektens forslag skal bekreftes ved praktisk måling så tidlig og ofte som mulig: spesielt nye og risikofylte forslag må måles tidlig mht alle kritiske egenskaper og side effekter, slik at de kan erstattes ved svikt.

.

Mer detaljerte info

Software 2015 talen min er kun 20 minutter. Men for de som vil ha en utvidet versjon og mer detaljerte kunnskaper, kan vi tilby følgende

1. Fullstendig sett med slides
   http://www.gilb.com/dl603
2. Video på engelsk av talen (90 minutter, Keynote)
   http://vimeo.com/user22258446/review/79092608/600e7bd650
3. Kort artikkel
   http://www.gilb.com/dl658
4. Gratis 2 dagers kurs i London
   https://events.bcs.org/book/1366/
5. Gratis Kurs i Norge Se OSWA Meetup tilbud. På engelsk normalt.
   Se evt http://gilb.com/CourseSchedule
   http://www.meetup.com/Oslo-Software-Architecture/events/207367122/
   link er til Okt 2014 kurset. Vi planlegger 2015 nå.
6. Krav Kurs i Dataforening på Norsk, ikke gratis
   https://www.dataforeningen.no/value-requirement.5570836-330190.html
7. Ny uferdig bokmanuskript:
   http://tinyurl.com/ValuePlanning

8. Over 100 artikler, case studies, slides, bøker mv: Gilb.com/downloads

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

Tilbake!