Tutorial Human Side of Systems Architecting

by *Gerrit Muller*   Buskerud University Collge and Embedded Systems Institute
e-mail: `gerrit.muller@embeddedsystems.nl`
`www.gaudisite.nl`

Abstract

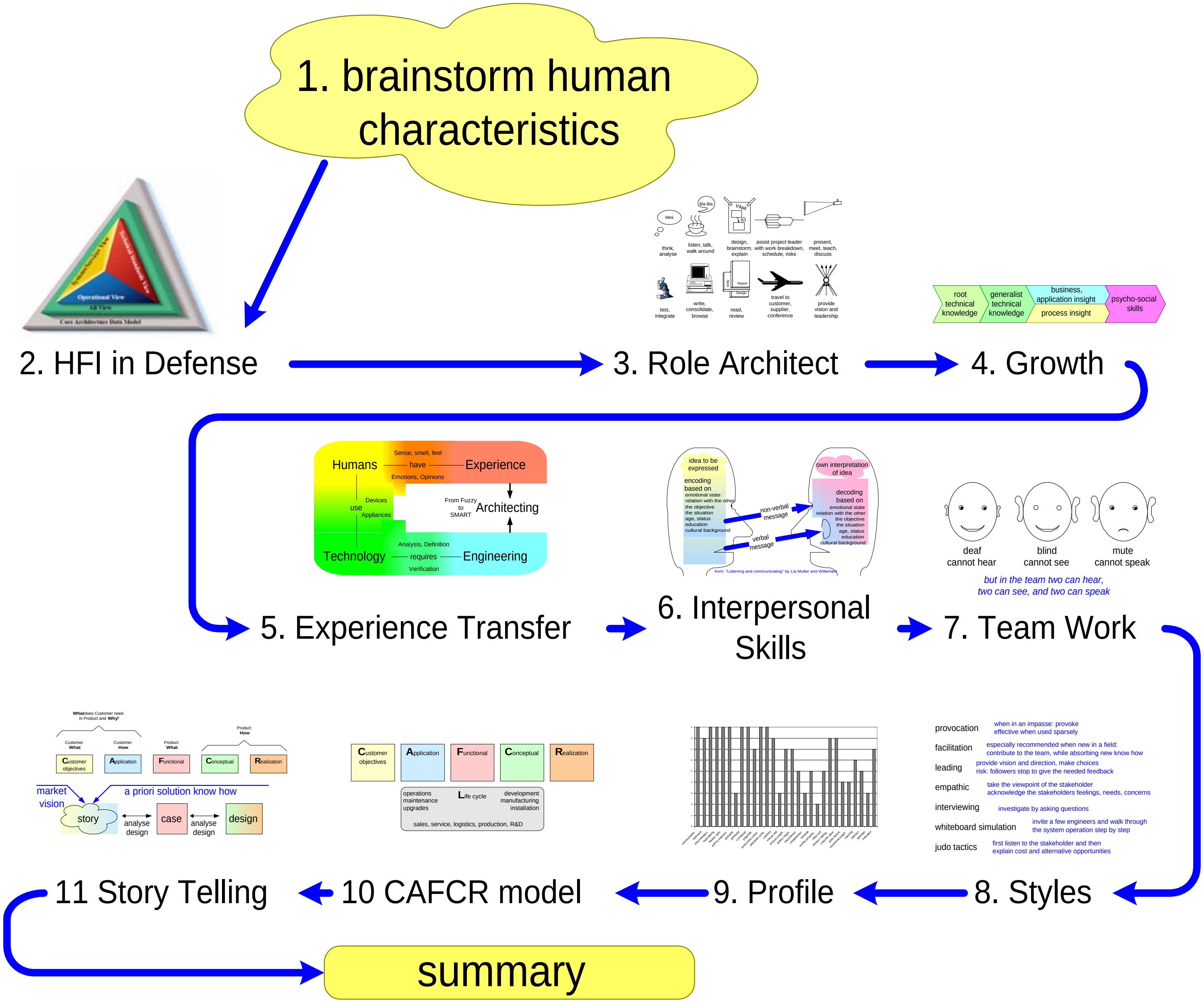
Architects play a crucial role in creating systems that fit well in the human needs. The creation of these systems requires many human interactions between all stakeholders. The background of architects, however, is completely different, mostly technical. We bring insight in the human aspects of systems architecting and we provide an approach with related tools to address the human aspects.

Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

6th March 2009
status: planned
version: 0.1

Figure Of Contents™



Brainstorm

What human characteristics

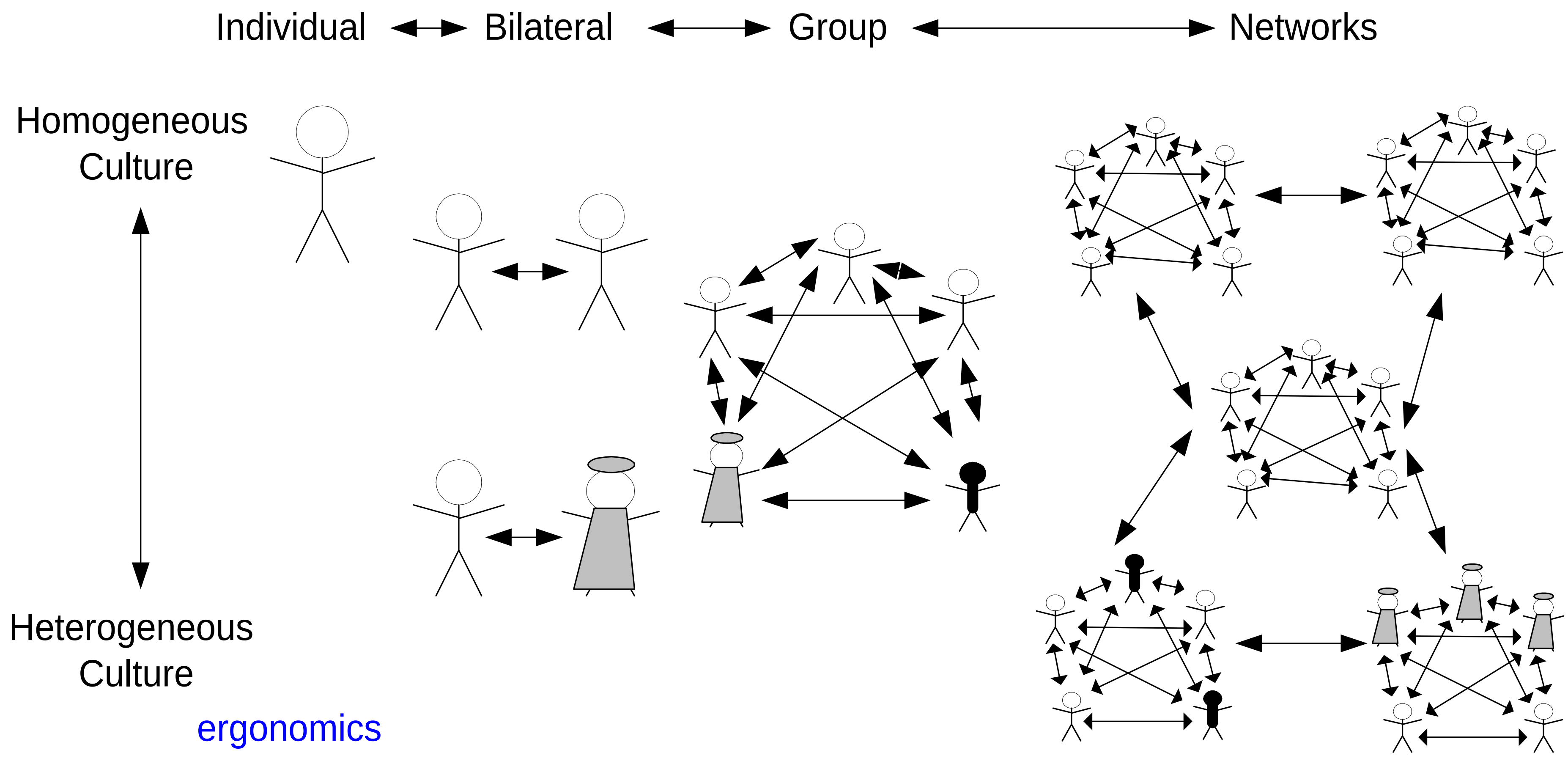
do you know

that impact the

use, specification and design

of systems?

Overview of Human Aspects



Individual ↔ Bilateral ↔ Group ↔ Networks

Homogeneous Culture

Heterogeneous Culture

ergonomics

psychology

group dynamics

political science

psychiatry

pedagogy

didactics

sociology

criminology

medicin

Human Factors in Defense

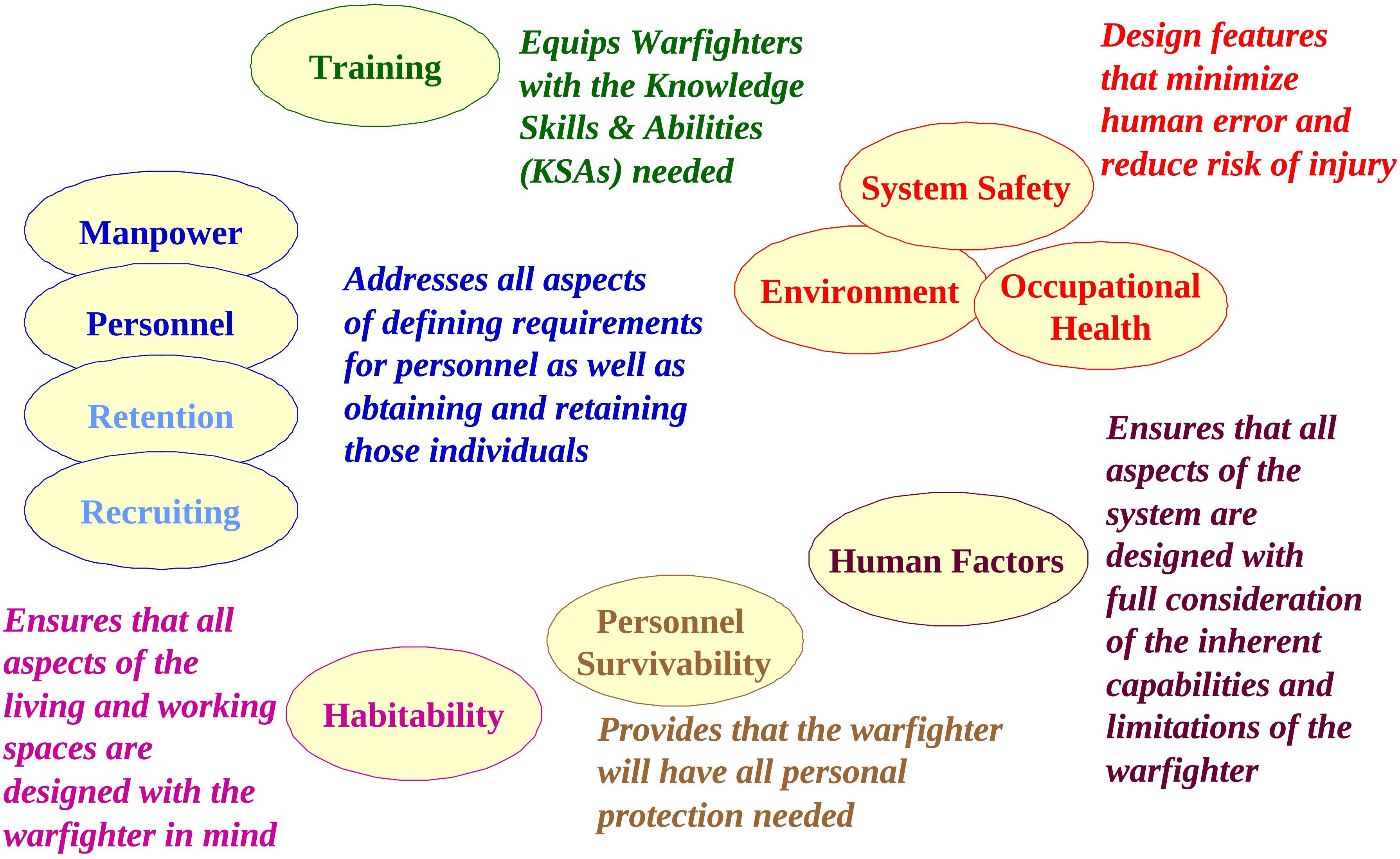
by *Gerrit Muller* Embedded Systems Institute
e-mail: `gerrit.muller@embeddedsystems.nl`
`www.gaudisite.nl`

Abstract

The defense industry has recognized the importance of human factors for system design. Some processes and procedures are available to address these needs. In this paper we provide a brief overview of ongoing *Human Factors* or *Human Systems Integration* activities in Defense.

6th March 2009
status: preliminary draft
version: 0

Human Systems Integration DoD Acquisition



Training

Equips Warfighters with the Knowledge Skills & Abilities (KSAs) needed

Manpower

Personnel

Retention

Recruiting

Addresses all aspects of defining requirements for personnel as well as obtaining and retaining those individuals

System Safety

Environment

Occupational Health

Design features that minimize human error and reduce risk of injury

Human Factors

Ensures that all aspects of the system are designed with full consideration of the inherent capabilities and limitations of the warfighter

Personnel Survivability

Provides that the warfighter will have all personal protection needed

Habitability

Ensures that all aspects of the living and working spaces are designed with the warfighter in mind

Human Systems Integration in DoD Acquisition by Ms. Nancy Dolan CNO N125
https://acc.dau.mil/GetAttachment.aspx?id=25755&pname=file&aid=3181&lang=en-US

Human Engineering from Naval Perspective

1. Mission Analysis
2. Requirements Analysis
3. Function Analysis
4. Function Allocation
5. Task Design and Analysis
6. Human Interface and Team Development
7. Performance, Workload, and Training Level Estimation
8. User and Requirements Reviews

from ONR (Office of Naval Research)/SC-21 Manning Affordability Initiative
www.hf.faa.gov/docs/508/docs/Human_System_Engineering_(NSWC).pdf

Human Views for MODAF

HV-A: Personnel Availability

HV-B: Quality Objectives and Metrics

HV-C: Human Interaction Structure

HV-D: Organisation

HV-E: Human Functions and Tasks

HV-F: Roles and Competencies

HV-G: Dynamic Drivers of Human Behaviour

from The Human View Handbook for MODAF
www.hfidtc.com/MoDAF/HV Handbook First Issue.pdf

The Role and Task of the System Architect

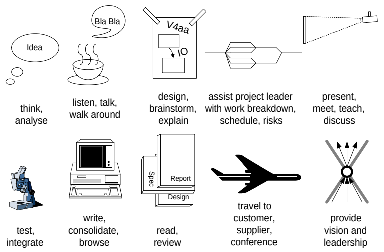
by *Gerrit Muller* Embedded Systems Institute
e-mail: gerrit.muller@embeddedsystems.nl
www.gaudisite.nl

Abstract

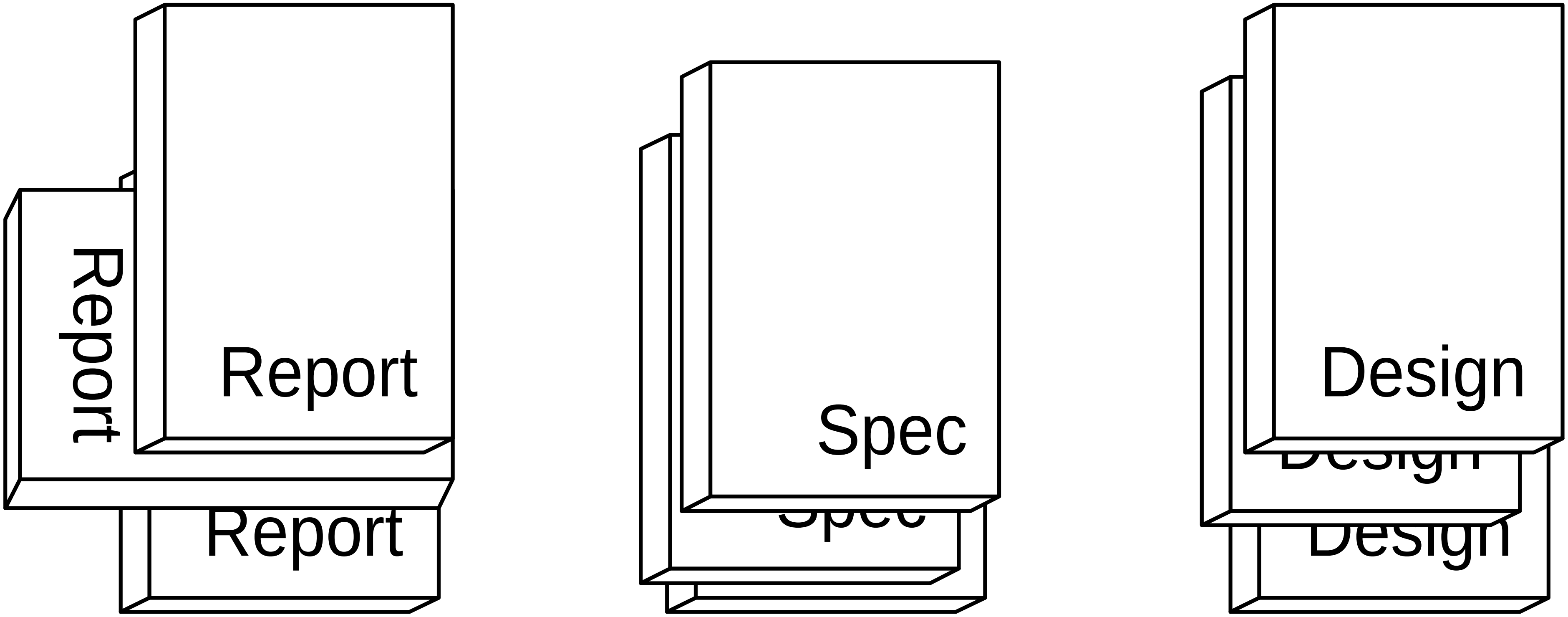
The role of the system architect is described from three viewpoints: deliverables, responsibilities and activities. This description shows the inherent tension in this role: a small set of hard deliverables, covering a fuzzy set of responsibilities, hiding an enormous amount of barely visible day-to-day work.

Distribution

6th March 2009
status: draft
version: 1.3


Idea
Bla Bla
V4aa
IO
think, analyse
listen, talk, walk around
design, brainstorm, explain
assist project leader with work breakdown, schedule, risks
present, meet, teach, discuss
Spec
Report
Design
test, integrate
write, consolidate, browse
read, review
travel to customer, supplier, conference
provide vision and leadership

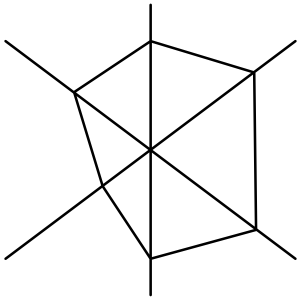
Deliverables of the System Architect



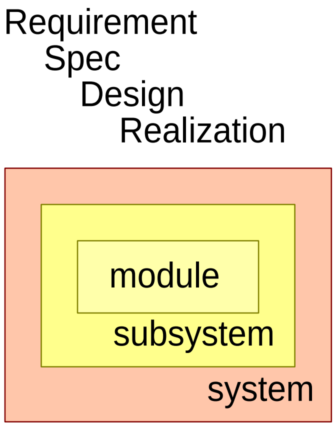
List of Deliverables

- Requirements (**what** is needed)
- Specification (**what** will be realized)
- Design (**how** the system will be realized)
- Verification Specification (**how** the system will be verified)
- Verification Report (the result of the verification)
- Feasibility Report (the results of a feasibility study)
- Roadmap

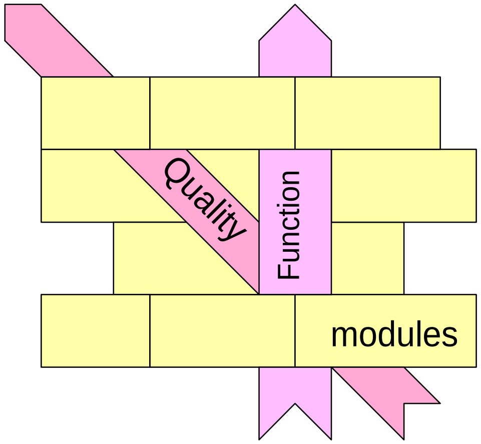
Responsibilities of the System Architect



Balance



Consistency

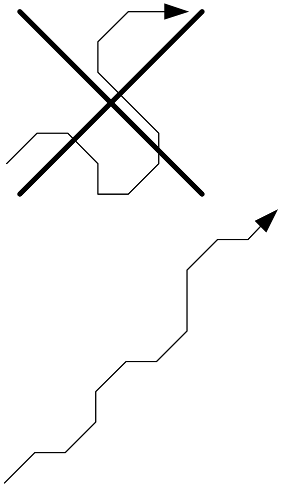


Decomposition
Integration

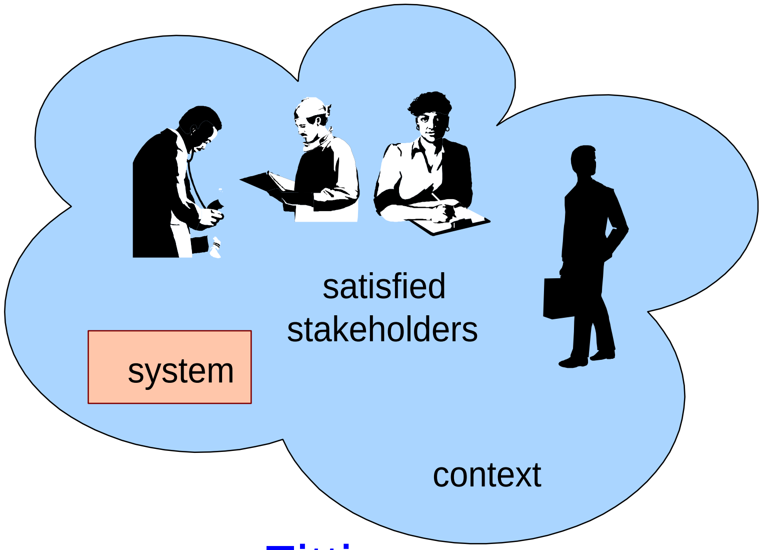
Overview

KISS

Elegance
Simple



Integrity

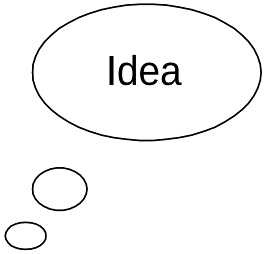


Fitting

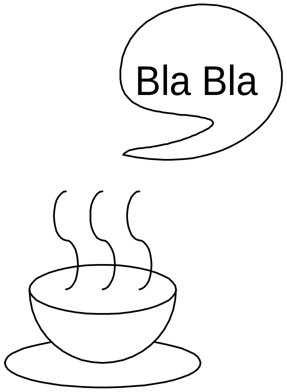
Examples of Secondary Responsibilities

responsibility	primary owner
business plan, profit	business manager
schedule, resources	project leader
market, salability	marketing manager
technology	technology manager
process, people	line manager
detailed designs	engineers

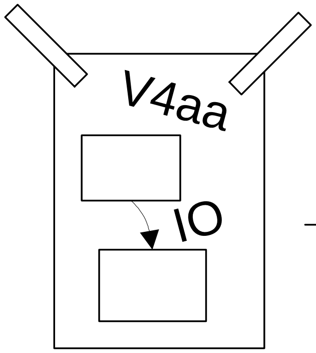
What does the System Architect **do**?



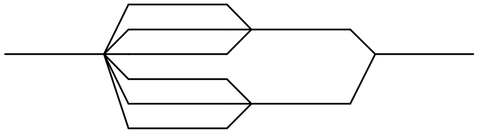
think,
analyse



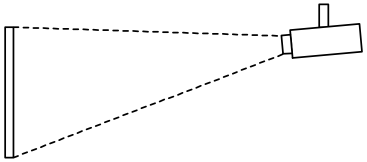
listen, talk,
walk around



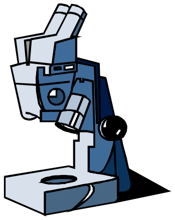
design,
brainstorm,
explain



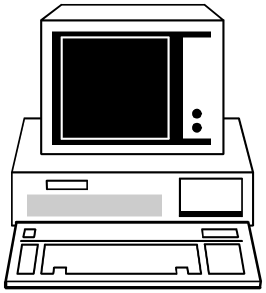
assist project leader
with work breakdown,
schedule, risks



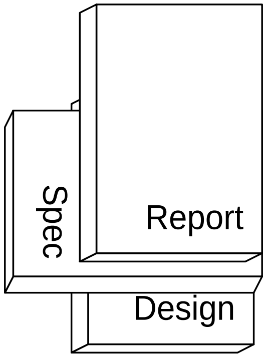
present,
meet, teach,
discuss



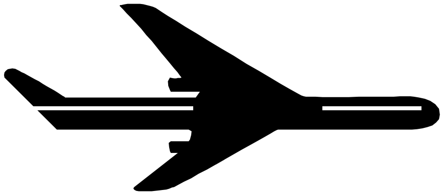
test,
integrate



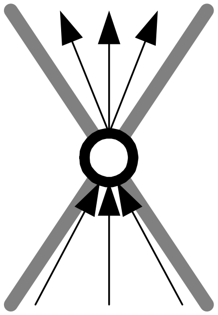
write,
consolidate,
browse



read,
review

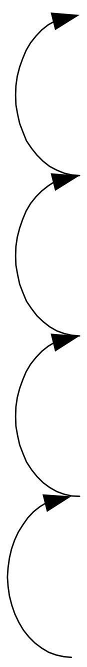


travel to
customer,
supplier,
conference



provide
vision and
leadership

From Detail to Overview



consolidation in deliverables

meetings

informal contacts

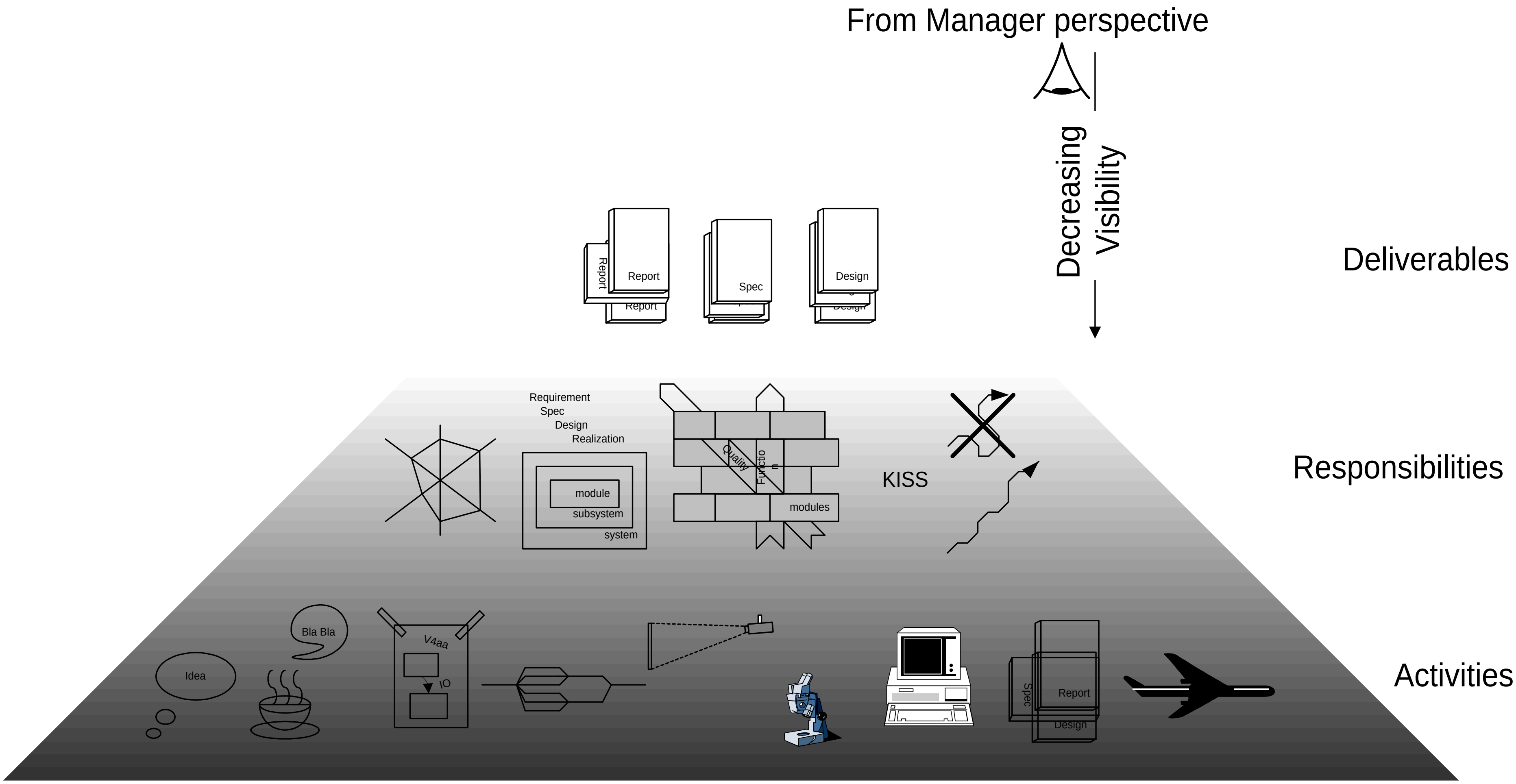
sampling scanning

	Quantity per year (order of magnitude)	architect time per item
driving views	10	100 hrs
shared issues	10^2	1 hr
touched details	10^4	0.5..10 min
seen details	$10^5..10^6$	0.1 .. 1 sec
product details	$10^7..10^{10}$	
real world facts	infinite	

Reality or Virtuality?

Abstractions only exist for concrete facts.

Visible Output versus Invisible Work



From Manager perspective

Decreasing Visibility

Deliverables

Responsibilities

Activities

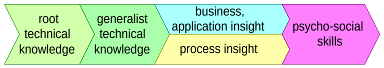
The Awakening of a System Architect

by *Gerrit Muller* Embedded Systems Institute
e-mail: `gerrit.muller@embeddedsystems.nl`
`www.gaudisite.nl`

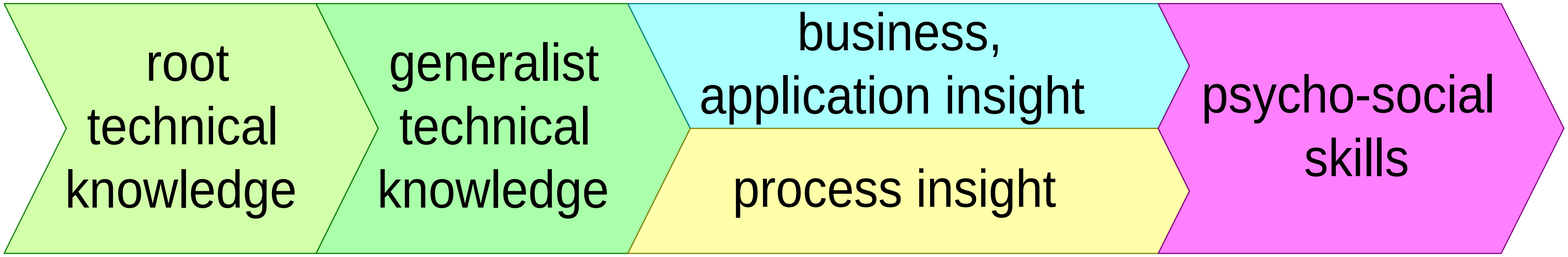
Abstract

The typical phases of a system architect development are described, beginning at the fundamental technology knowledge, with a later broadening in technology and in business aspects. Finally the subtlety of individual human beings is taken into account.

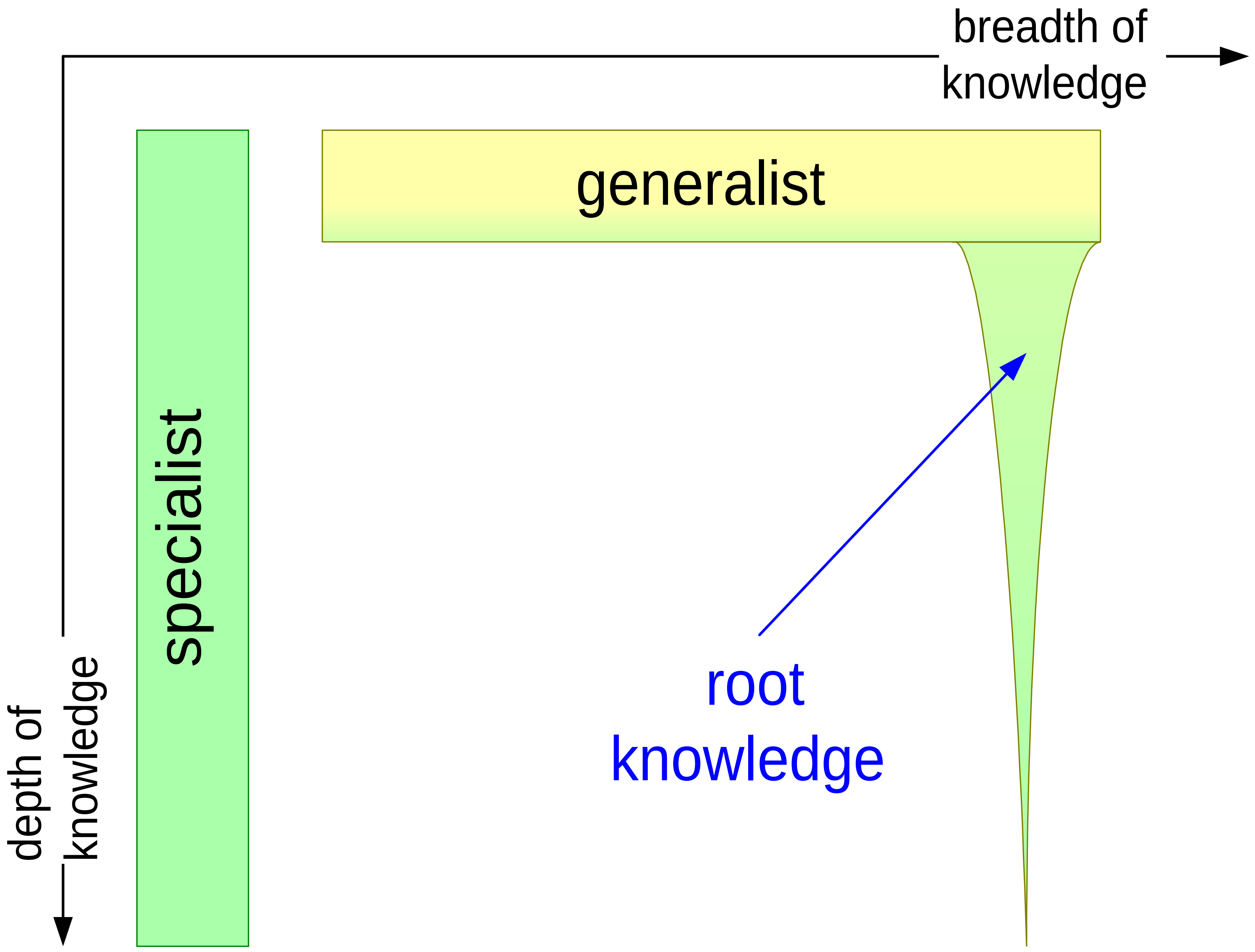
6th March 2009
status: draft
version: 1.0



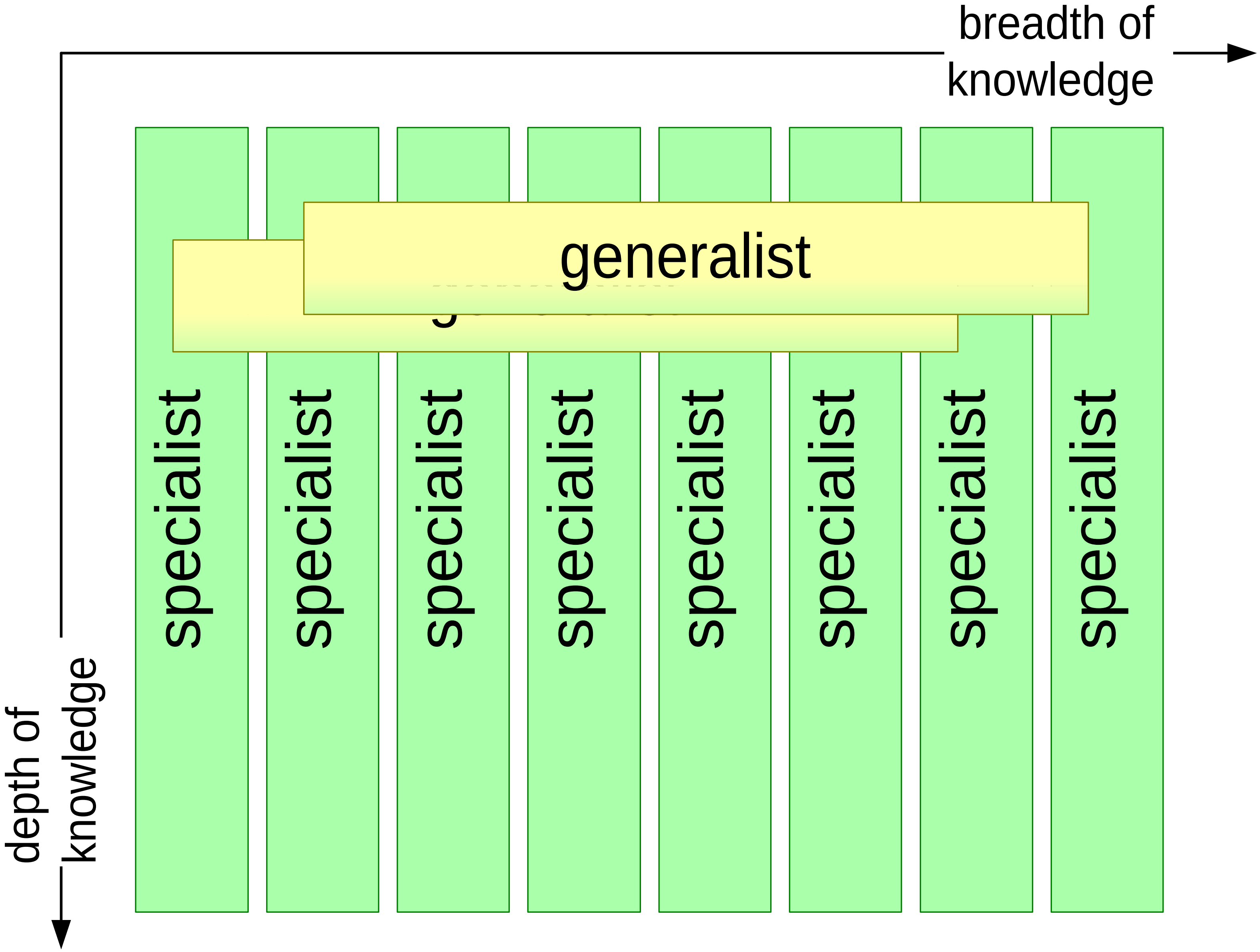
Typical Growth of a System Architect



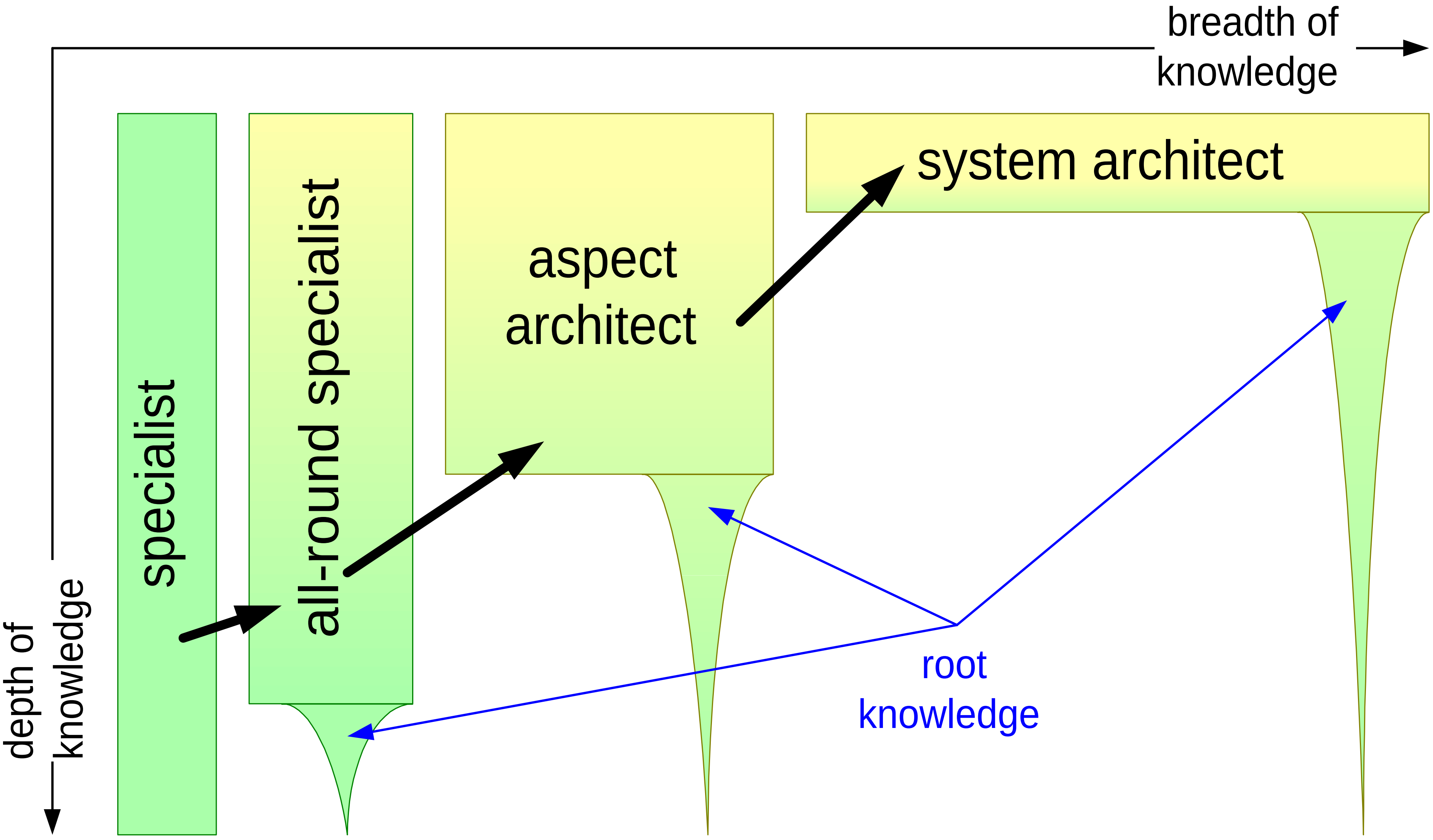
Generalist versus Specialist



Generalists and Specialists are Complementary



Spectrum from Specialist to System Architect



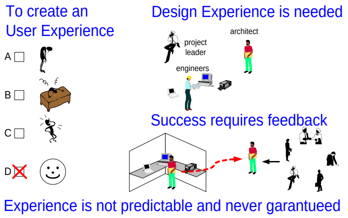
Architecting for Humans; How to Transfer Experience?

by *Gerrit Muller*   Embedded Systems Institute
e-mail: `gerrit.muller@embeddedsystems.nl`
`www.gaudisite.nl`

Abstract

The ultimate goal of Product Creation is to create products which give the user a great experience. User experience is very intangible. Product engineering focuses on tangible requirements. Successfull product require both sound engineering as well as creative design. The question is how to obtain a workforce, which is capable of both activities?

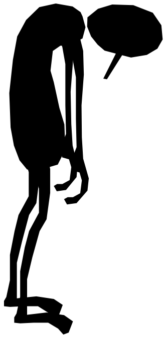
The education of successfull engineers is limited to engineering methods. Additional skills are acquired by experience. Unfortunately experience cannot be transfered from one engineer to the next. Such a transfer is approximated by active personal development.


To create an
User Experience
Design Experience is needed
architect
project
leader
engineers
A □
B □
C □
D ☒
Success requires feedback
Experience is not predictable and never garantueed

Distribution

6th March 2009
status: finished
version: 1.3

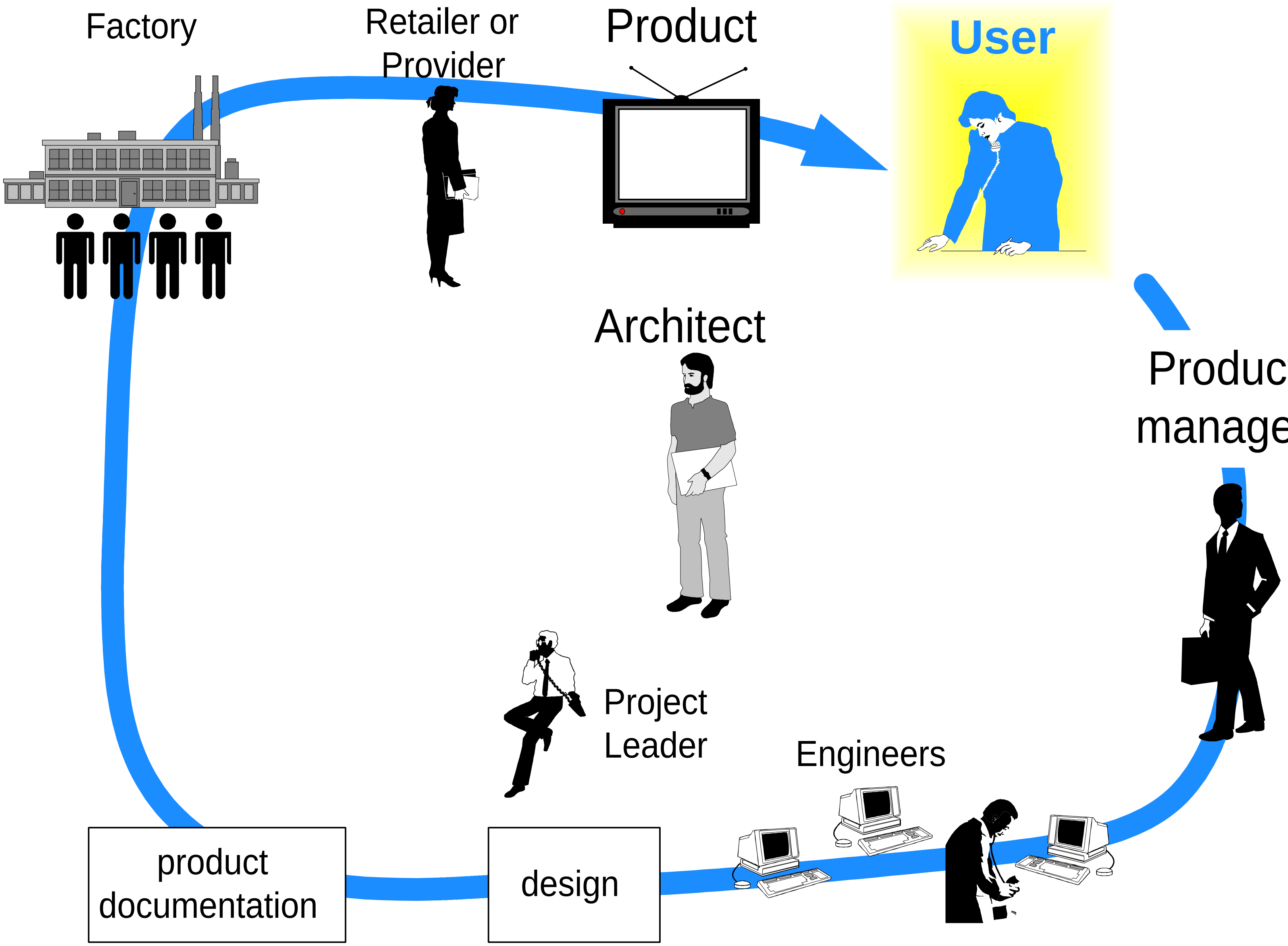
Did you ever program a VCR or PVR?

A □  depressed

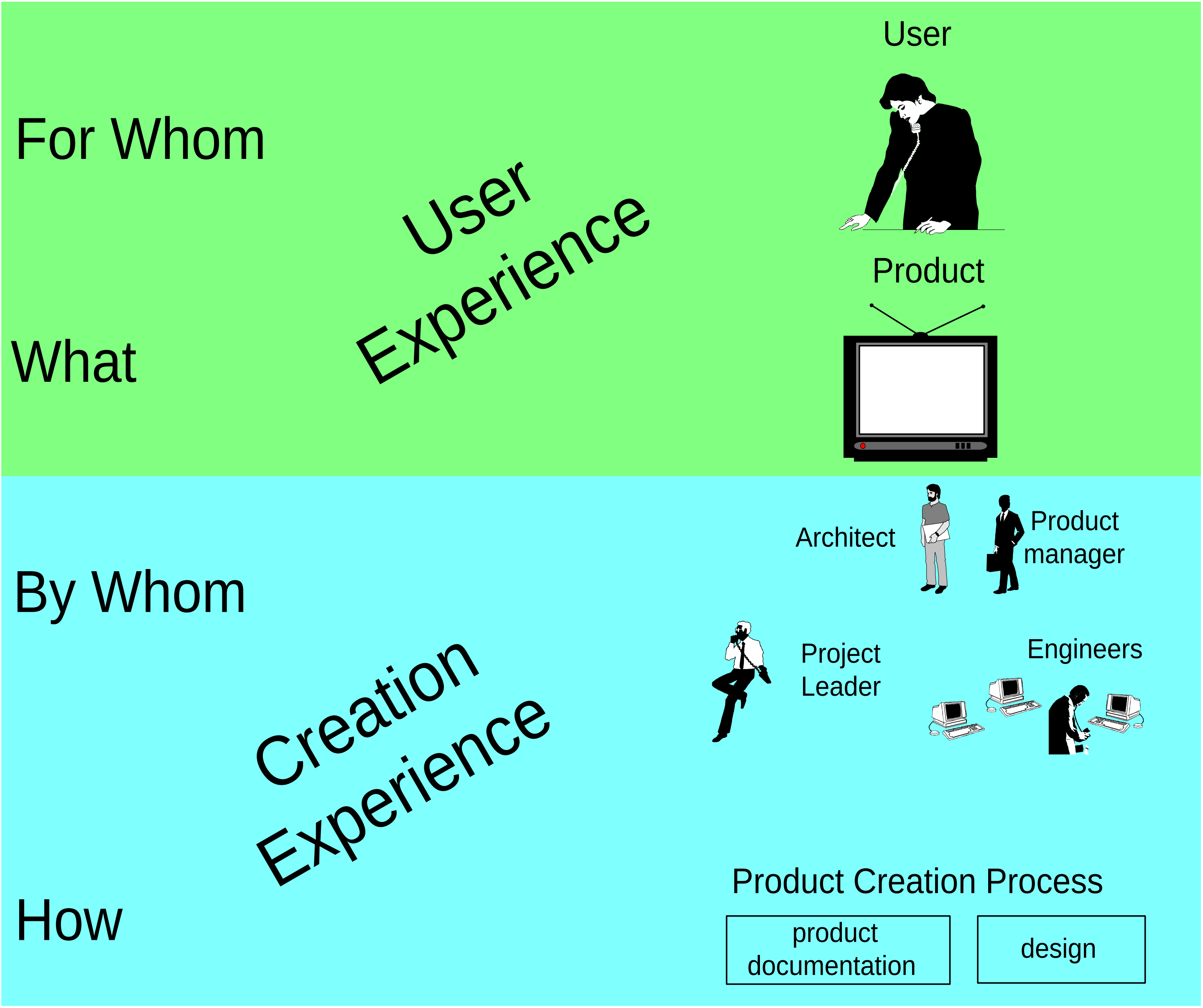
B □ desparate

C □ hysteric

Product Creation Cycle



2 Levels of Experience



For Whom

What

User Experience

User

Product

By Whom

How

Creation Experience

Architect

Product manager

Project Leader

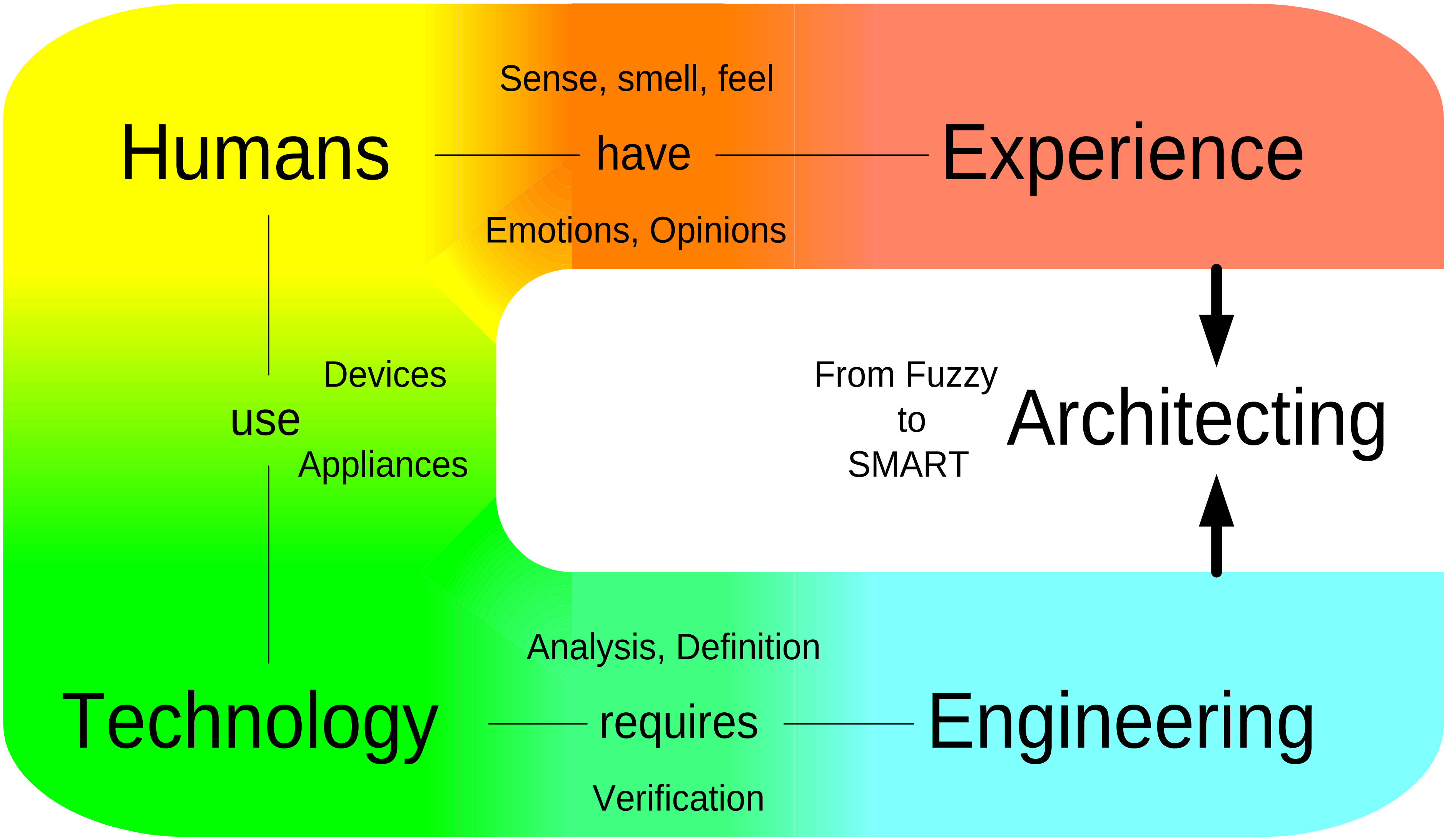
Engineers

Product Creation Process

product documentation

design

Bridging the gap between Experience and Engineering



Sense, smell, feel

Humans — have — Experience

Emotions, Opinions

Devices

use

Appliances

From Fuzzy
to
SMART

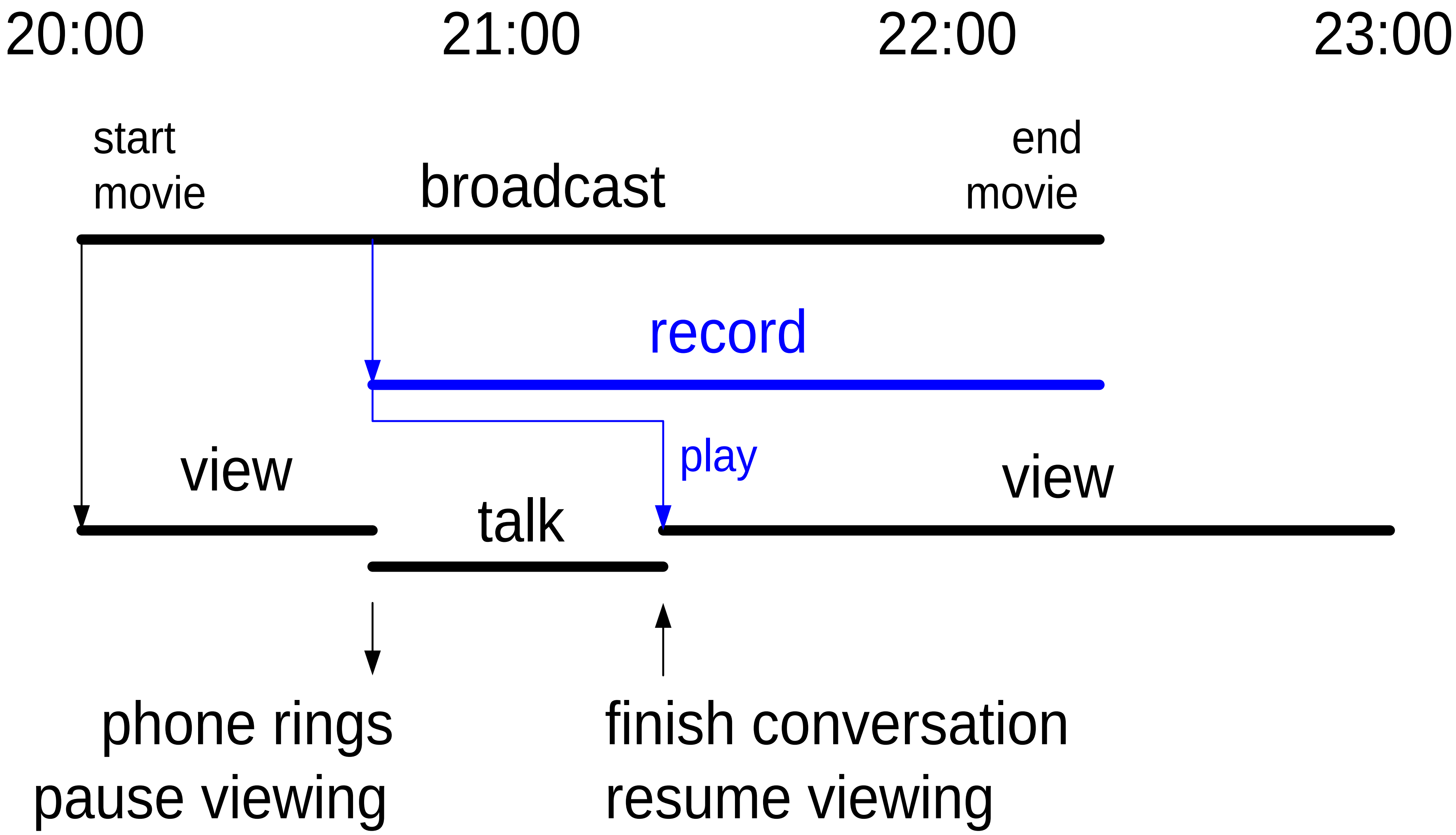
Architecting

Analysis, Definition

Technology — requires — Engineering

Verification

Example Time Shift recording



20:00

21:00

22:00

23:00

start movie

broadcast

end movie

record

play

view

talk

view

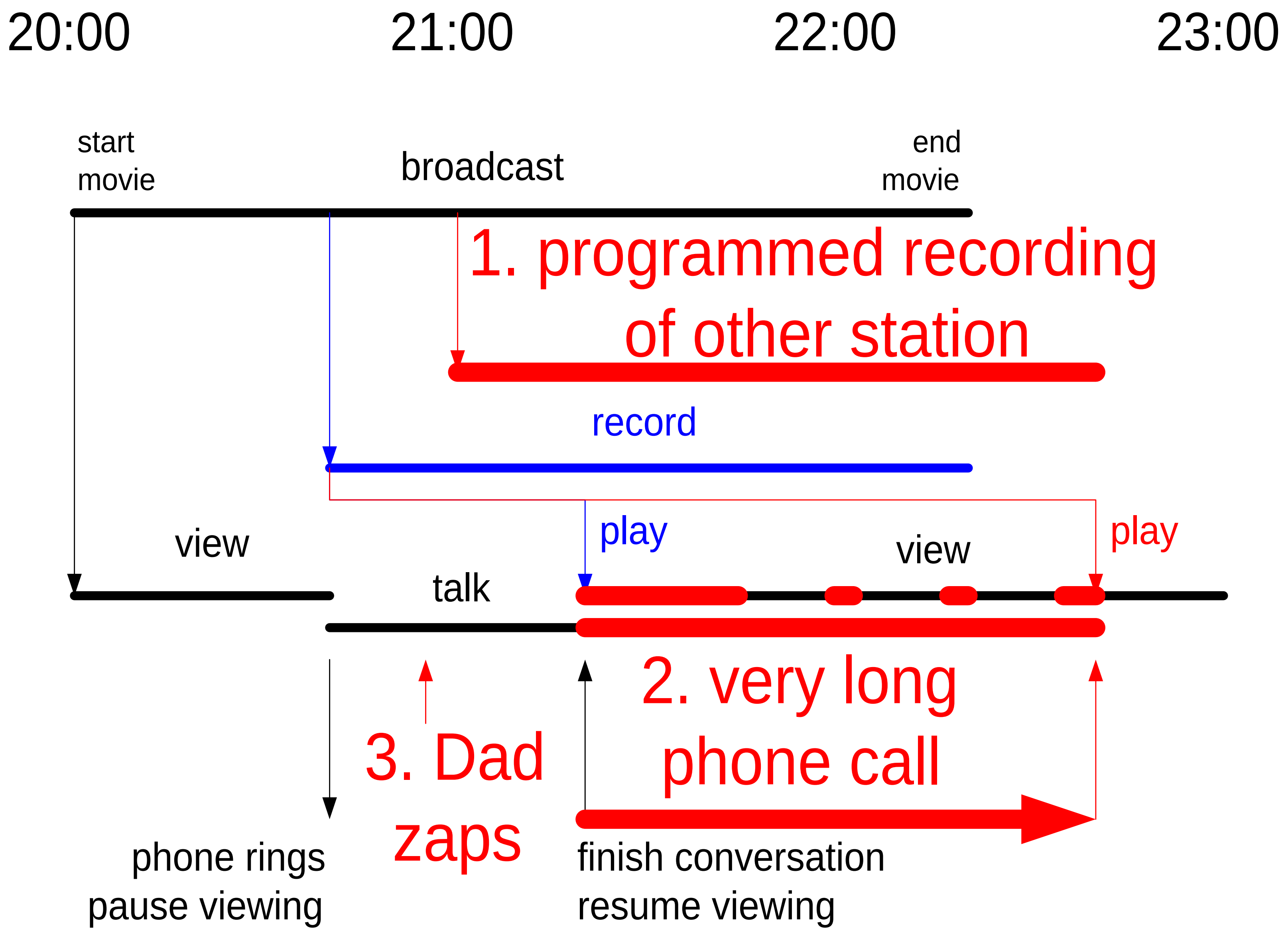
phone rings
pause viewing

finish conversation
resume viewing

Construction limits intrude in Experience

- number of tuners
- number of simultaneous streams (recording and playing)
- amount of available storage
- management strategy of storage space

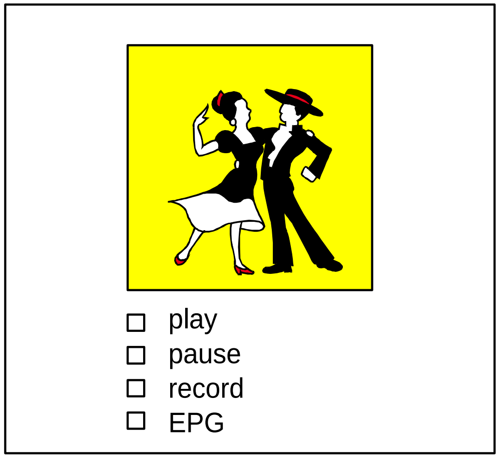
What if?


20:00
21:00
22:00
23:00
start
movie
broadcast
end
movie
1. programmed recording
of other station
record
view
play
view
play
talk
2. very long
phone call
3. Dad
zaps
phone rings
pause viewing
finish conversation
resume viewing

OOTI workshop 2001

Visual Basic Prototype:

enables "experiencing"



Requirements specification

Many tables, mostly addressing details

2.1.1 Real-time data requirements

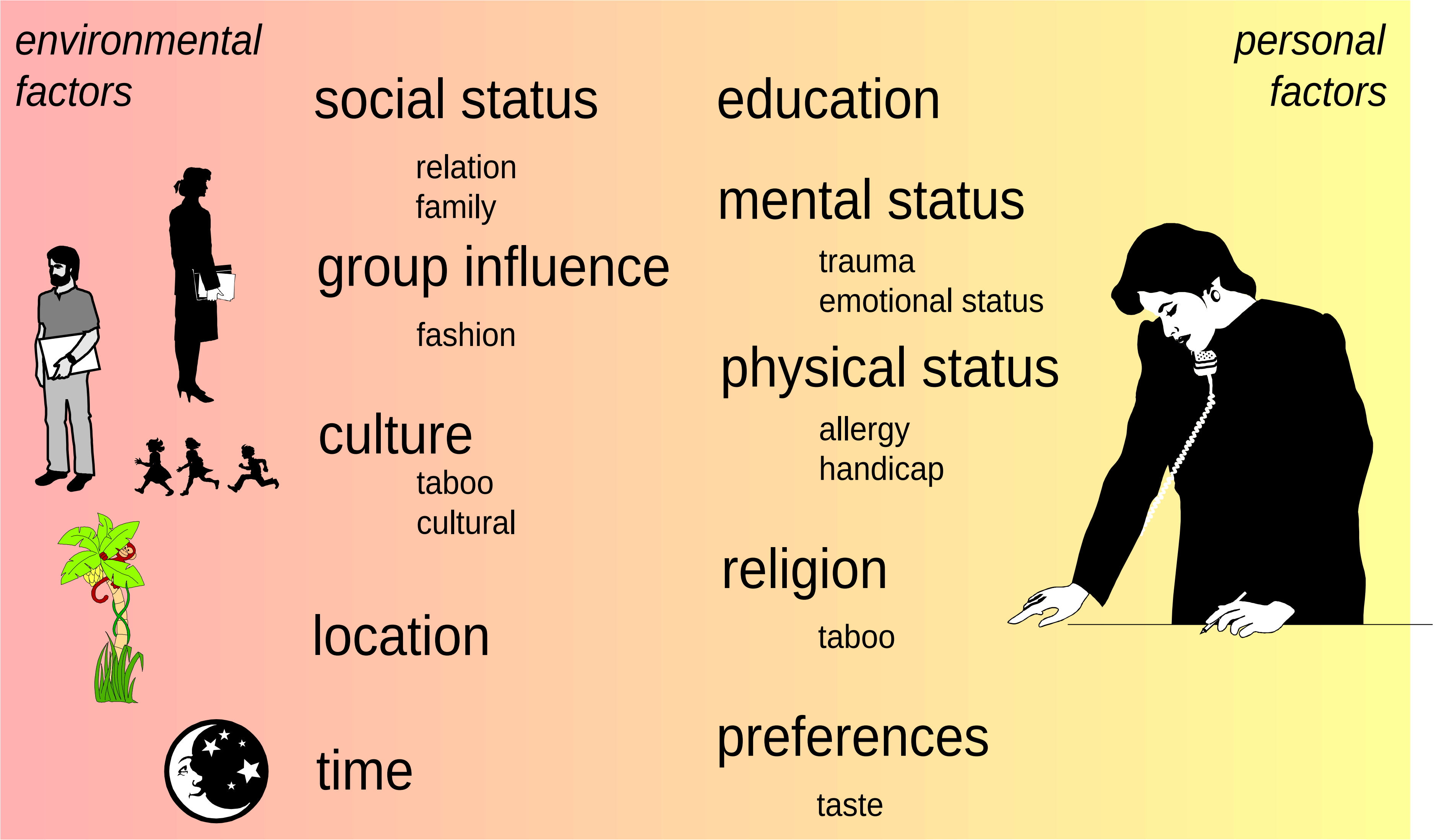
2.1.2 Implementation detail

2.1.3 Non-real time data requirements

1.1 Software Requirements

1.1.1	Real-time data requirements	*1.1.1.1 Access to the non-real-time data must be done in such a way that it does not interfere with the real-time data* *1.1.1.2 There must be no disruptions in output of video signal during the operation of VCR* 1.1.1.3 *Responsiveness for non real-time data is less then 150ms (the time for writing a block on HDD) for 2KB of non-video data*	
1.1.2	Implementation detail	*1.1.2.1 Management of HDD content must only be possible through the TOC in order to prevent unauthorized access to content of HDD* *1.1.2.2 Visual feedback is provided to the user via On-Screen Display* 1.1.2.3 *User input is provided via the RC*	
1.1.3	Non-real time data requirements	*1.1.3.1 User must be able to pause and unpause a title, played from HDD, while (s)he is watching it* *1.1.3.2 User can jump forward and backward in a title, from HDD, during watching of this title* *1.1.3.3 Names of titles should be derived from the information from the EPG (name of the program to be recorded, time and date of registration)*	

Factors influencing the User Experience



environmental factors

social status

- relation
- family

group influence

- fashion

culture

- taboo
- cultural

location

time

personal factors

education

mental status

- trauma
- emotional status

physical status

- allergy
- handicap

religion

- taboo

preferences

- taste

How to ”SMART”en Experience?

- define
- measure
- predict
- verify

Infinite Experience Space

People	Number of People on earth	$O(10^9)$
		*
Time	Human lifespan in seconds	$O(10^9)$
		*
Location	Square meters of planet earth	$O(10^{14})$
		*
...	...	...

Size of experience space ∞

It is not that bad :-)

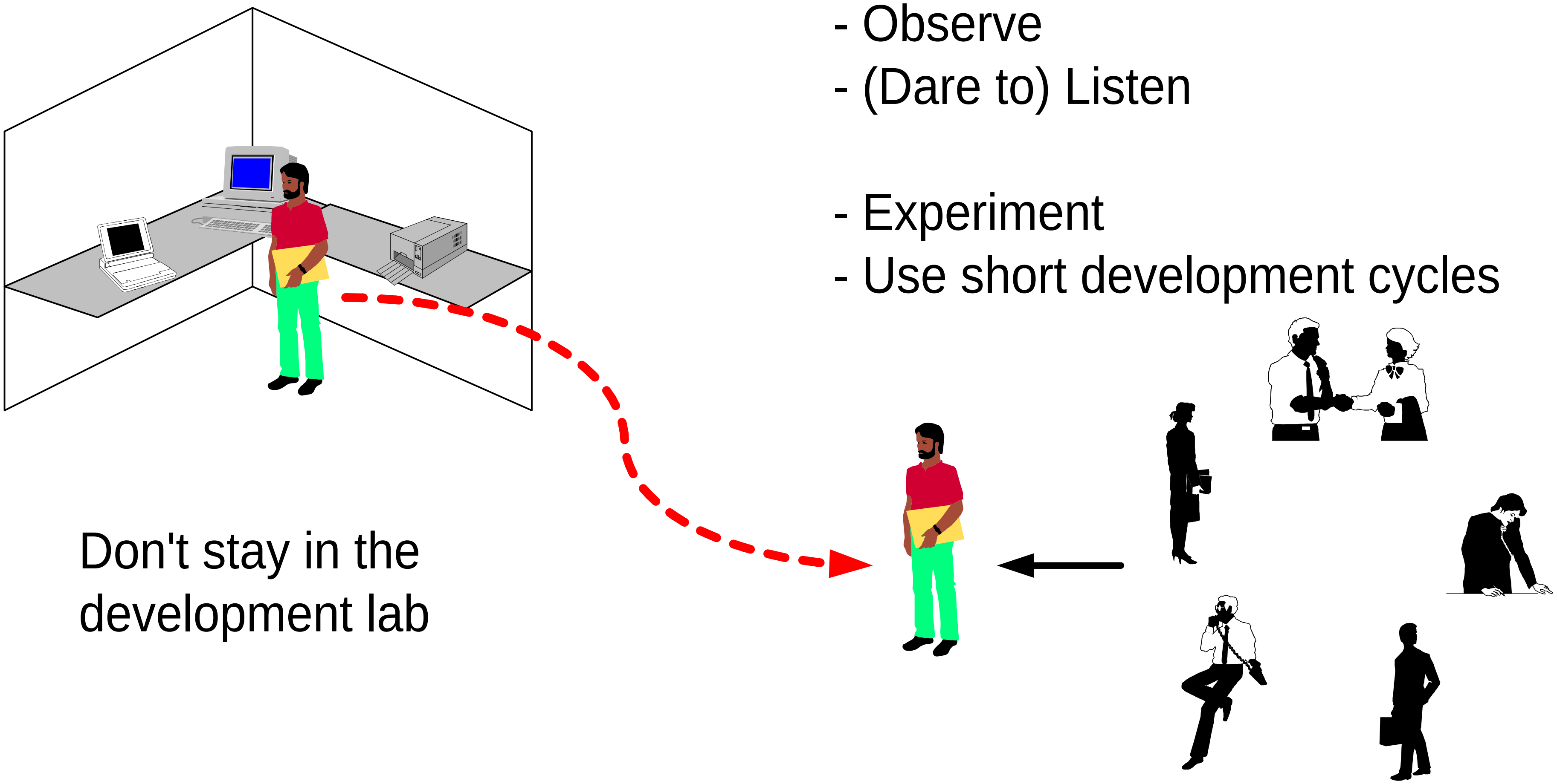
Many nice and successfull products exist!

Key Success Factor: Feedback

Obtain feedback from real users:

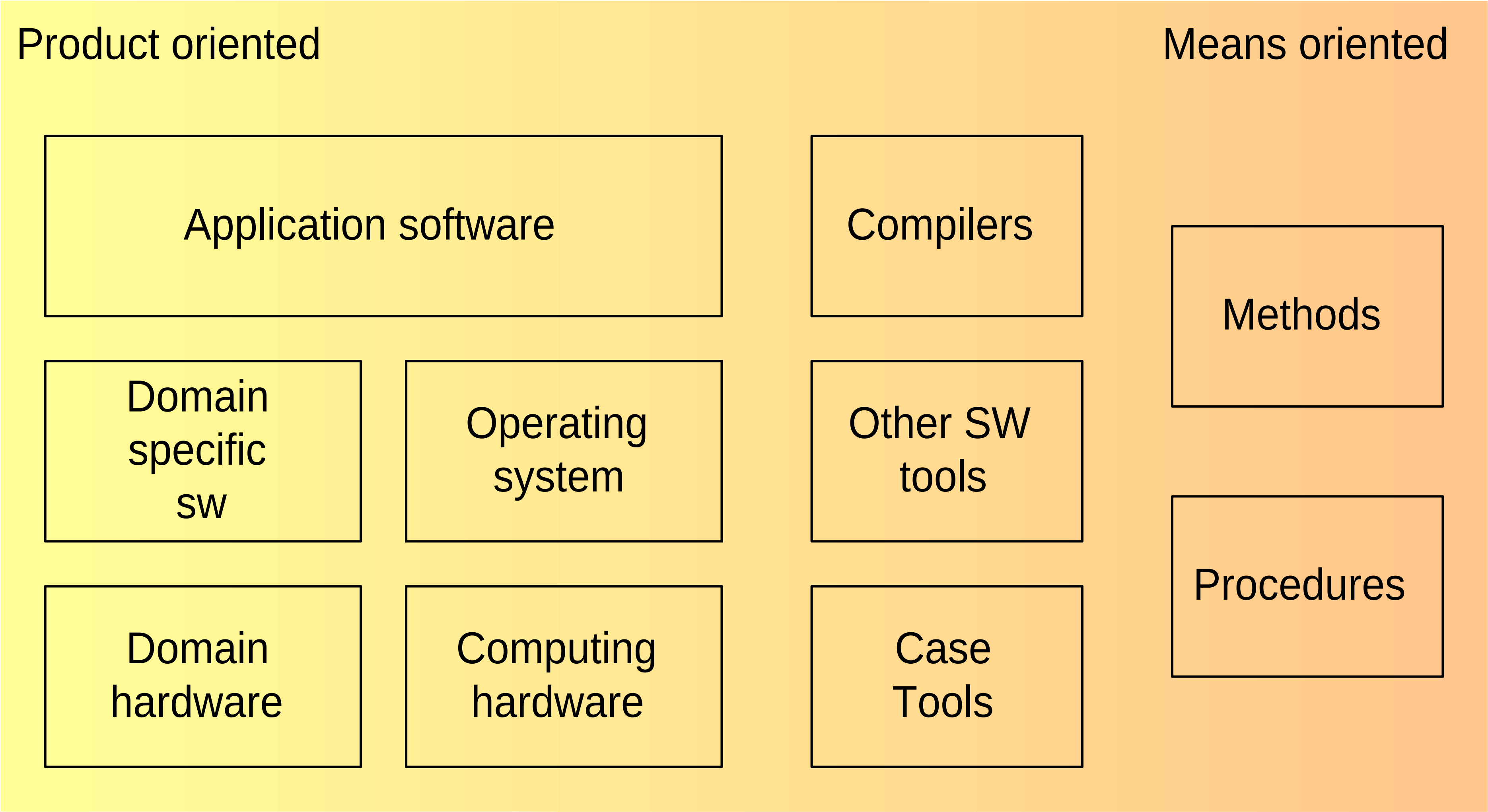
- Observe
- (Dare to) Listen

- Experiment
- Use short development cycles



Don't stay in the development lab

The world of the construction



Product oriented

Means oriented

Application software

Compilers

Methods

Domain specific SW

Operating system

Other SW tools

Procedures

Domain hardware

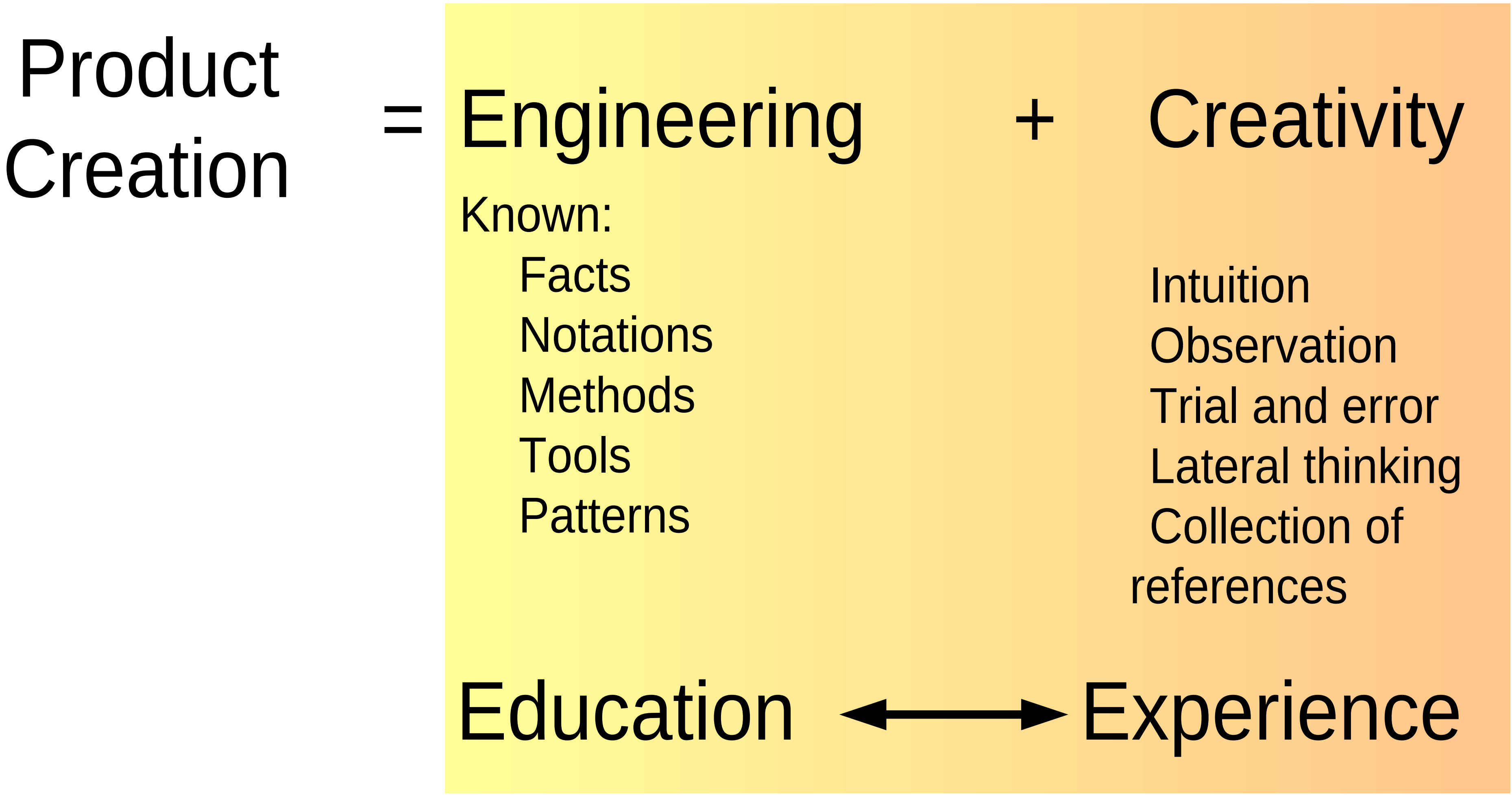
Computing hardware

Case Tools

Engineers are educated in construction disciplines

- Programming languages
- Operating systems
- Algorithms
- Data structures
- Formal specification and verification techniques
- Analysis, simulation techniques

Product Creation is much more than Engineering



Product Creation = Engineering + Creativity

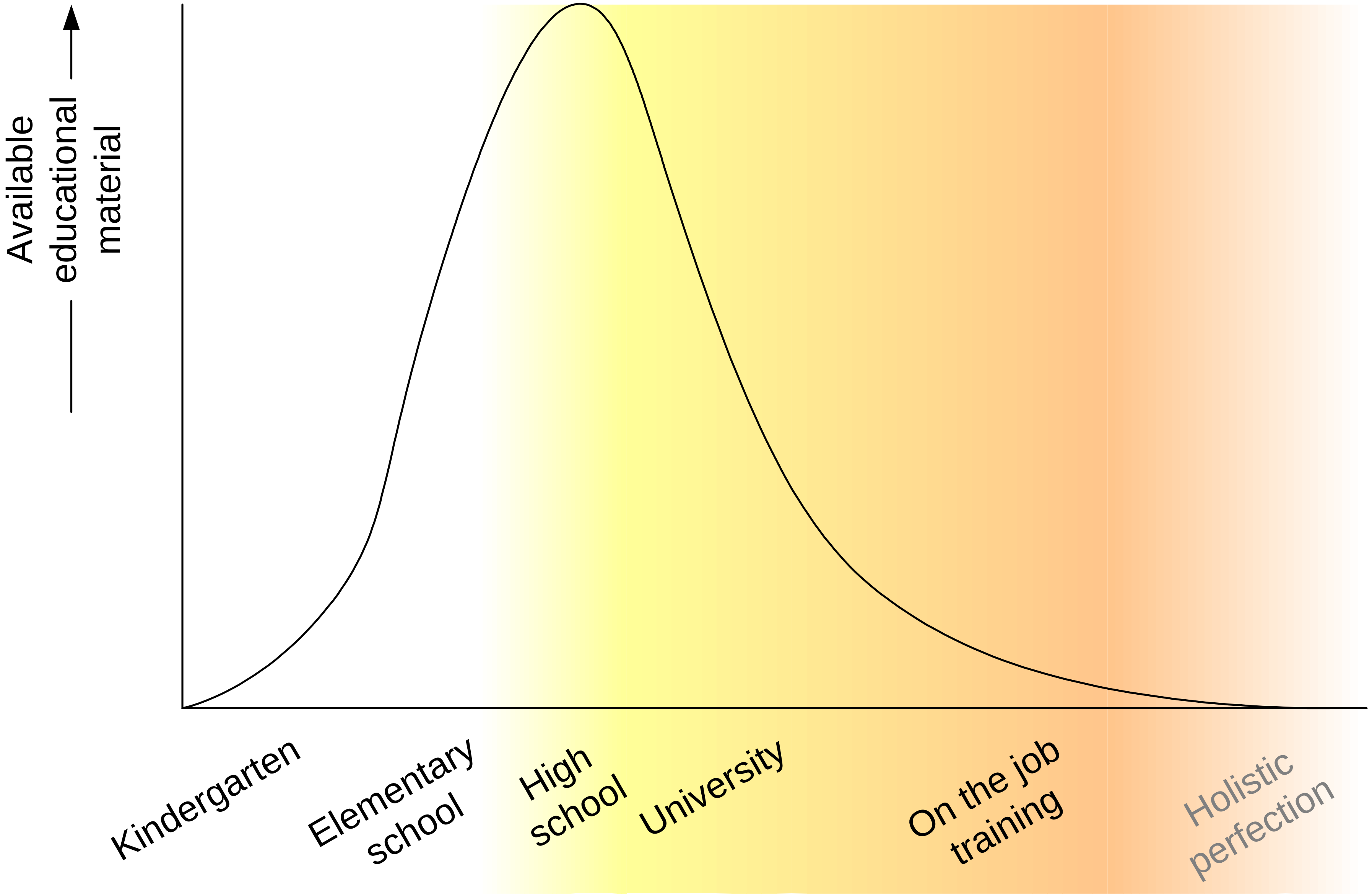
Known:

- Facts
- Notations
- Methods
- Tools
- Patterns

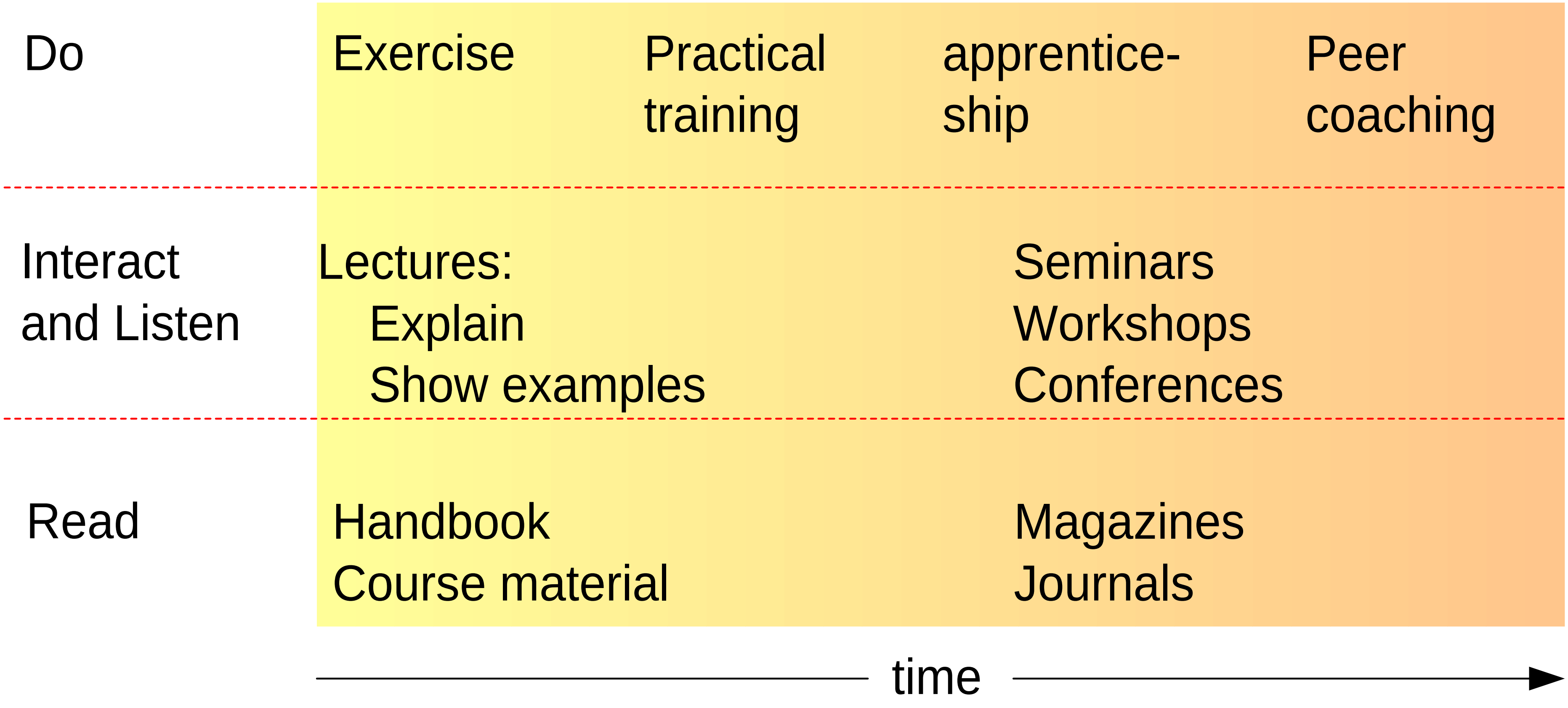
- Intuition
- Observation
- Trial and error
- Lateral thinking
- Collection of references

Education ⟷ Experience

Educational Material per education stage

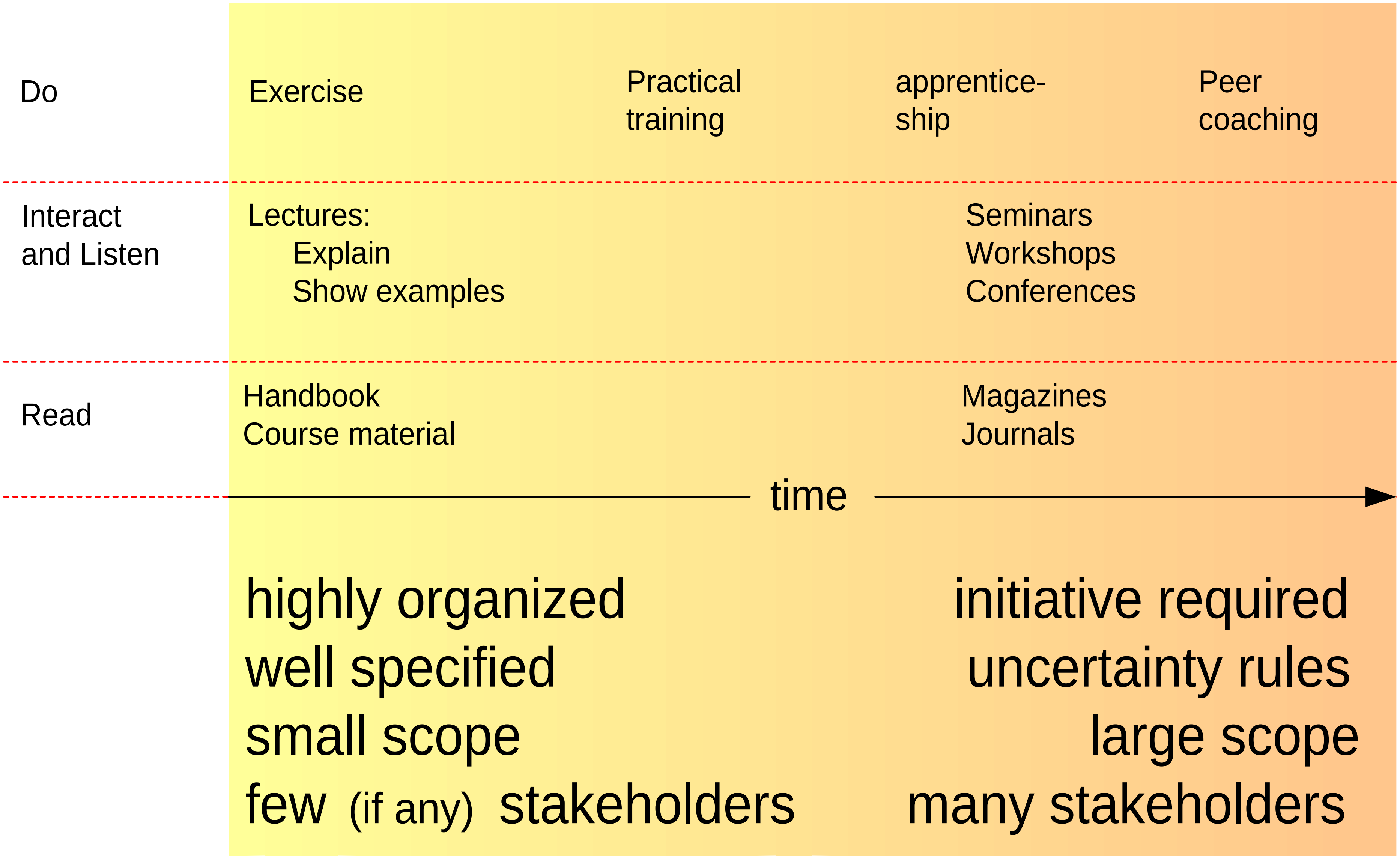


Changing Education model in time



Do	Exercise	Practical training	apprenticeship	Peer coaching
Interact and Listen	Lectures: Explain Show examples		Seminars Workshops Conferences	
Read	Handbook Course material		Magazines Journals	

Increasing Initiative required



Do

Exercise

Practical training

apprenticeship

Peer coaching

Interact and Listen

Lectures:
- Explain
- Show examples

Seminars
Workshops
Conferences

Read

Handbook
Course material

Magazines
Journals

time

highly organized
well specified
small scope
few (if any) stakeholders

initiative required
uncertainty rules
large scope
many stakeholders

Prerequisites for continuous successfull product creation

- Awareness of engineers of human aspects
- Active personal development drive of engineers
- Awareness of managers of education models
- Active motivation by managers

Architecting for Humans

To create an
User Experience

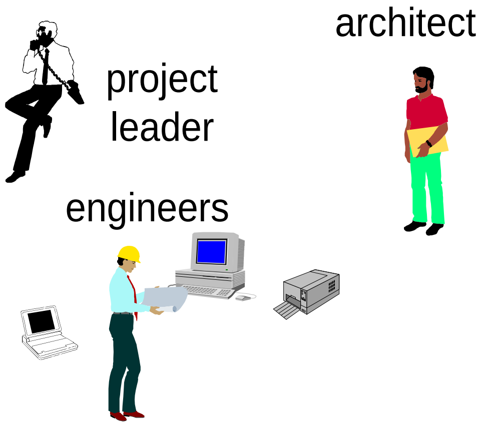
A □

B □

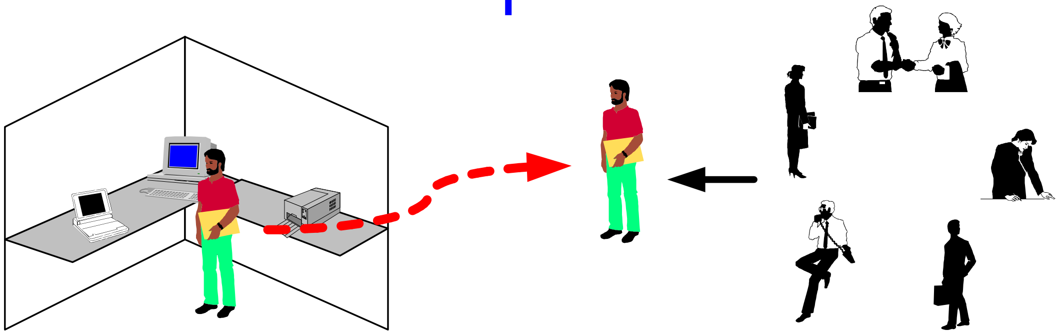
C □

D ☒

Design Experience is needed



Success requires feedback



Experience is not predictable and never garantueed

Experience Transfer

Design experience is not transferable
education is no substitute

Regular education =
Transfer of Engineering methods
+ Training

Transfer is approximated by
personal development

Personal Development =
On the job training
+ feedback
+ continuous personal education

Human Side: Interpersonal Skills

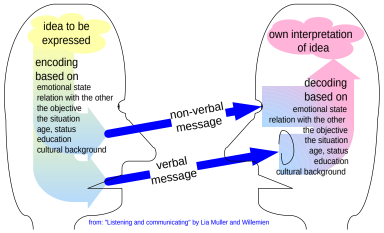
by *Gerrit Muller* Embedded Systems Institute
e-mail: `gerrit.muller@embeddedsystems.nl`
`www.gaudisite.nl`

Abstract

We discuss in this paper a set of skills and techniques to cooperate effectively between two individuals. We show the wonders of communication and then we address techniques such as investigation and acknowledgement, constructive feedback, conflict management, appraisal, good practices in a conversation, searching for ideas.

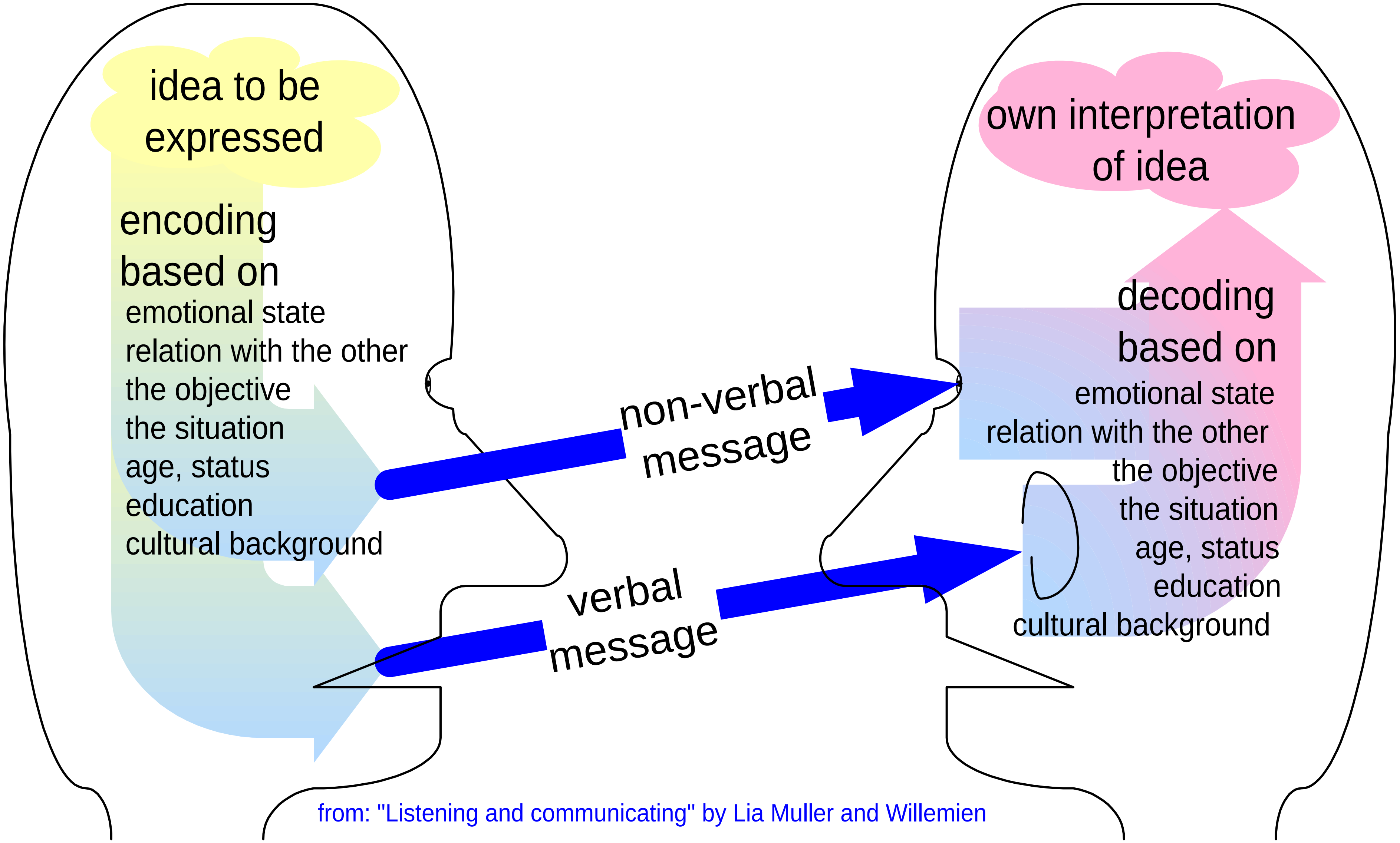
Distribution

6th March 2009
status: preliminary
draft
version: 0



from: "Listening and communicating" by Lia Muller and Willemien

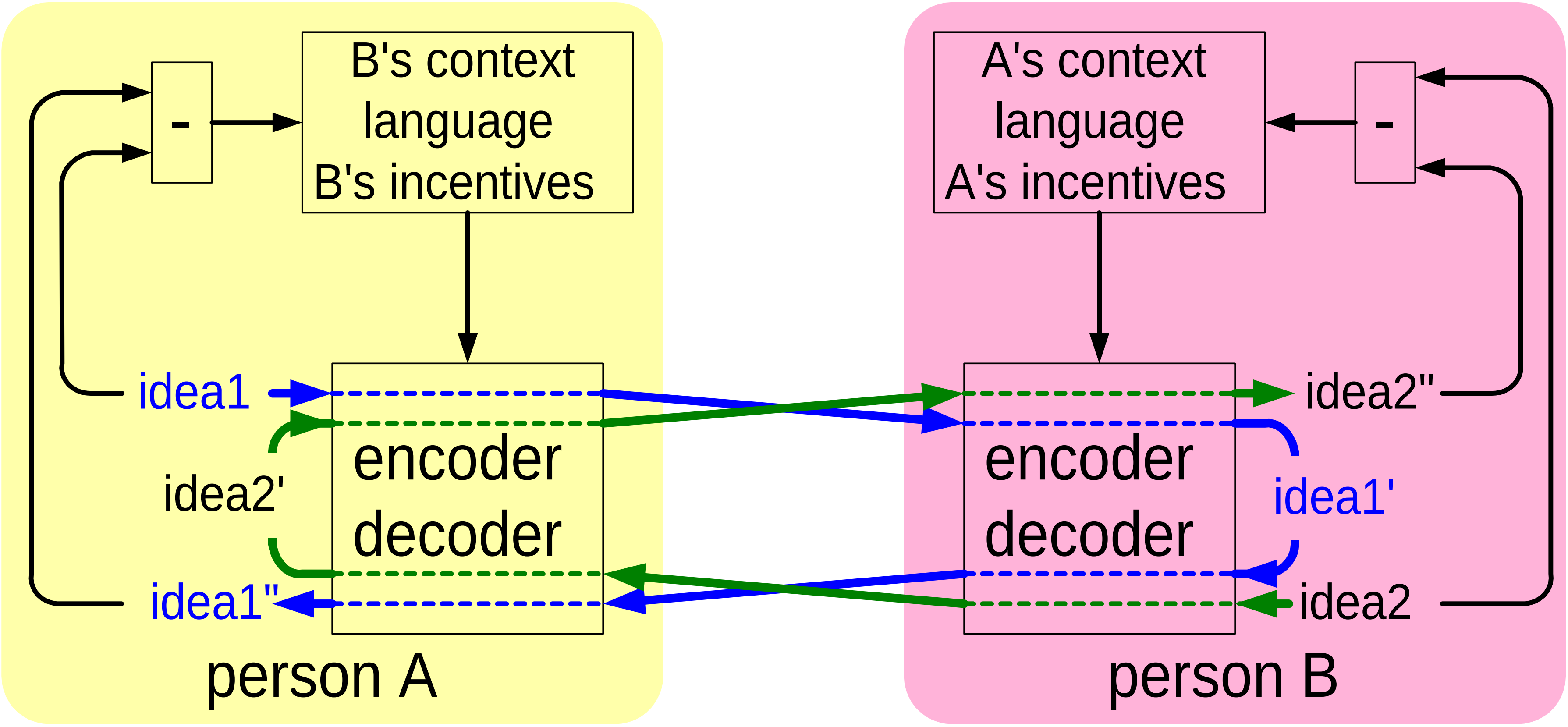
Active listening: the art of the receiver to decode the message



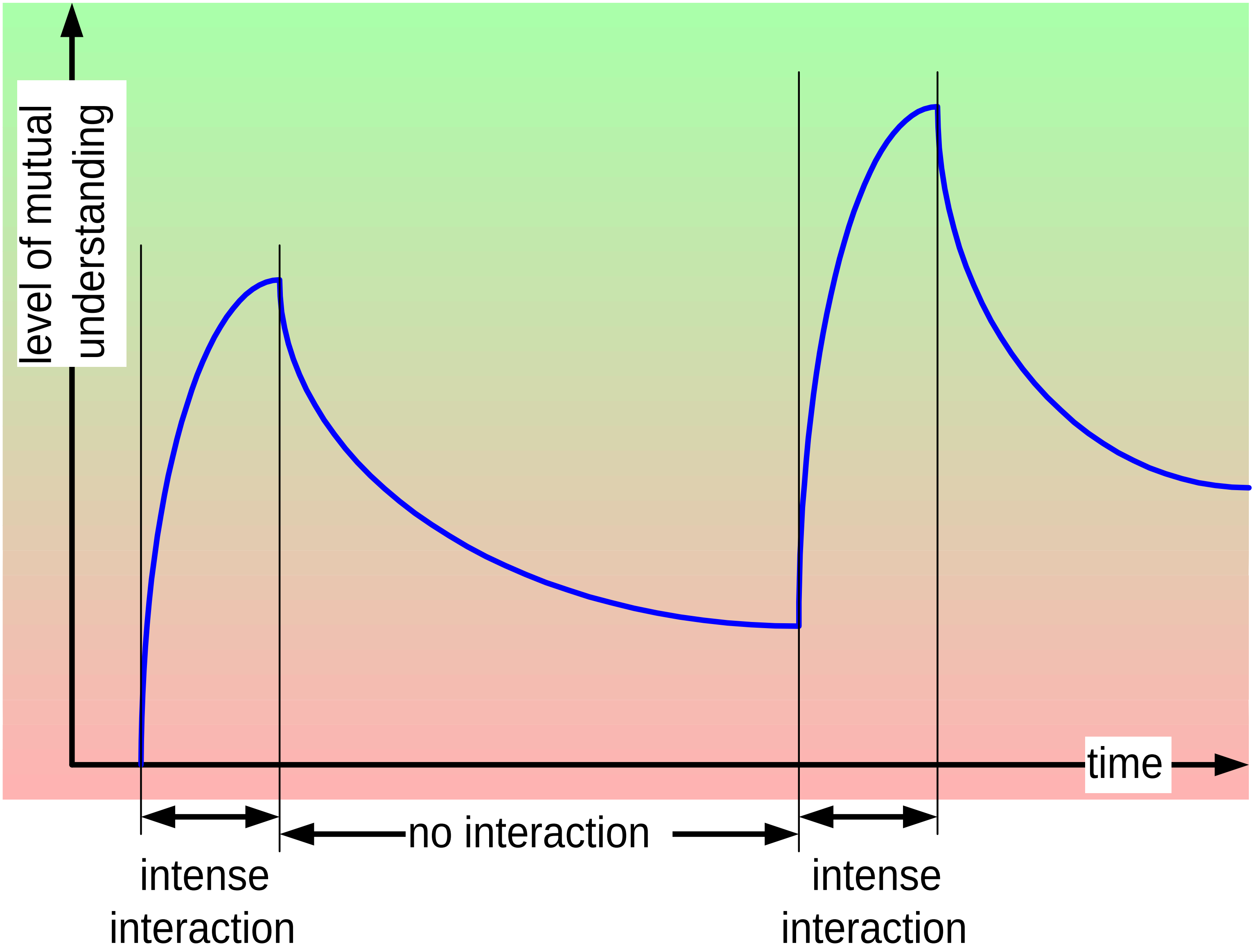
from: "Listening and communicating" by Lia Muller and Willemien

Intense interaction needed for mutual understanding

to calibrate:
repeat many times with different examples, illustrations and explanations



Mutual understanding as function of time



The material for interpersonal skills

is based on a set of techniques

from a course

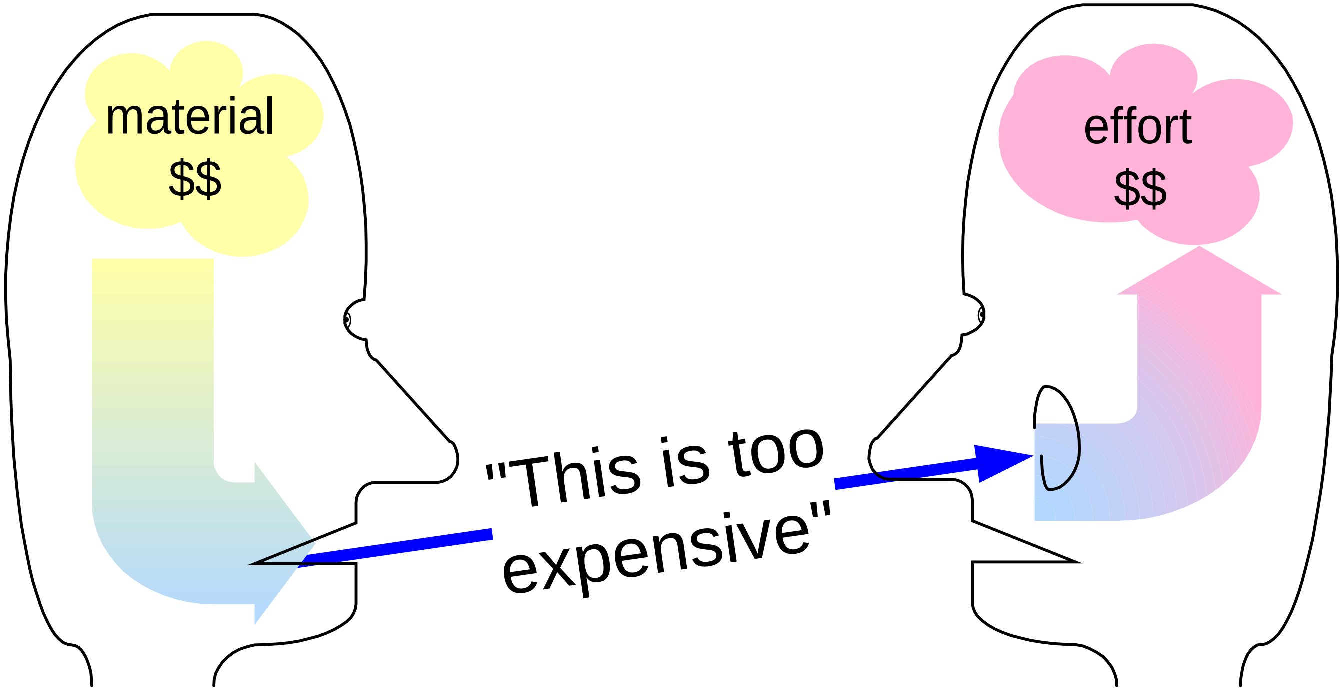
"Interpersonal Management Skills"

by

Hay Management Consultants

in 1998

Investigate and Acknowledge



investigate:

What has been said and why?

acknowledge:

Paraphrase what has been said and why?

i.e. use your own words

When a decision will be taken or an action will be started on the basis of exchanged information, opinions or suggestions
or
when the first reaction is to reject, ignore or contradict what you just heard.

Constructive Feedback

How

+ Indicate the strong points to be kept

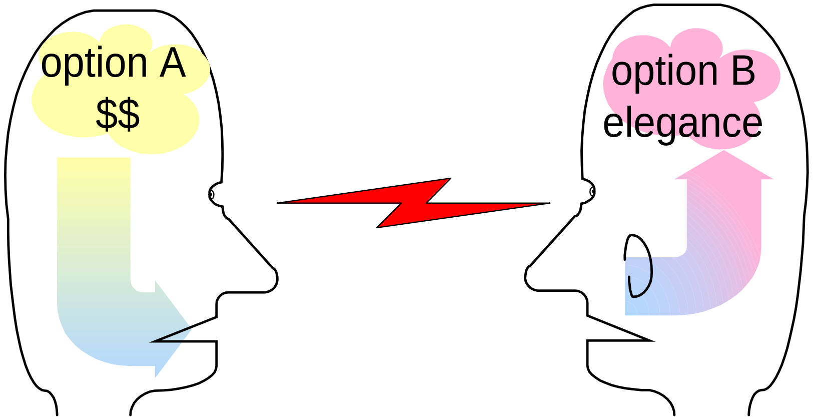
+ Indicate the points to be improved

+ Search for solutions which build upon the strong points and improve the weak points

When

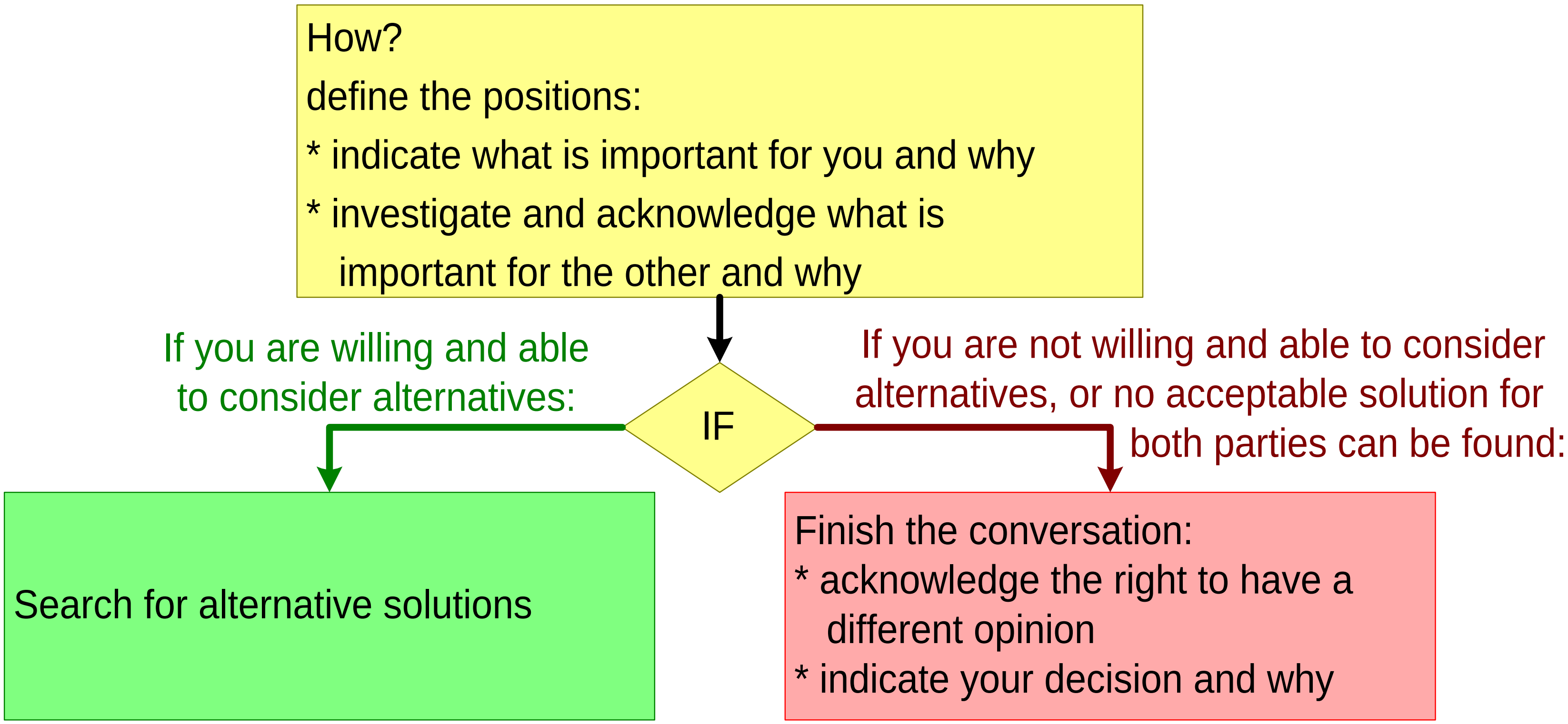
You want to facilitate someone to improve his/her performance

Conflict Management



When

in case of conflict



How?

define the positions:

* indicate what is important for you and why

* investigate and acknowledge what is important for the other and why

IF

If you are willing and able to consider alternatives:

Search for alternative solutions

If you are not willing and able to consider alternatives, or no acceptable solution for both parties can be found:

Finish the conversation:
* acknowledge the right to have a different opinion
* indicate your decision and why

Appraisal

When

Someone’s performance is important for you

* exceeding the expectations
* meets expectations continuously
* meets expectations, which exceed the normal
  performance level of this person

Appraise only when authentic!

How

+ Mention the performance very specific.
+ Mention the personal qualities which lead to this
  performance.
+ Describe which advantages arise for you,
  the department or the organization.

Conversation Good Practices

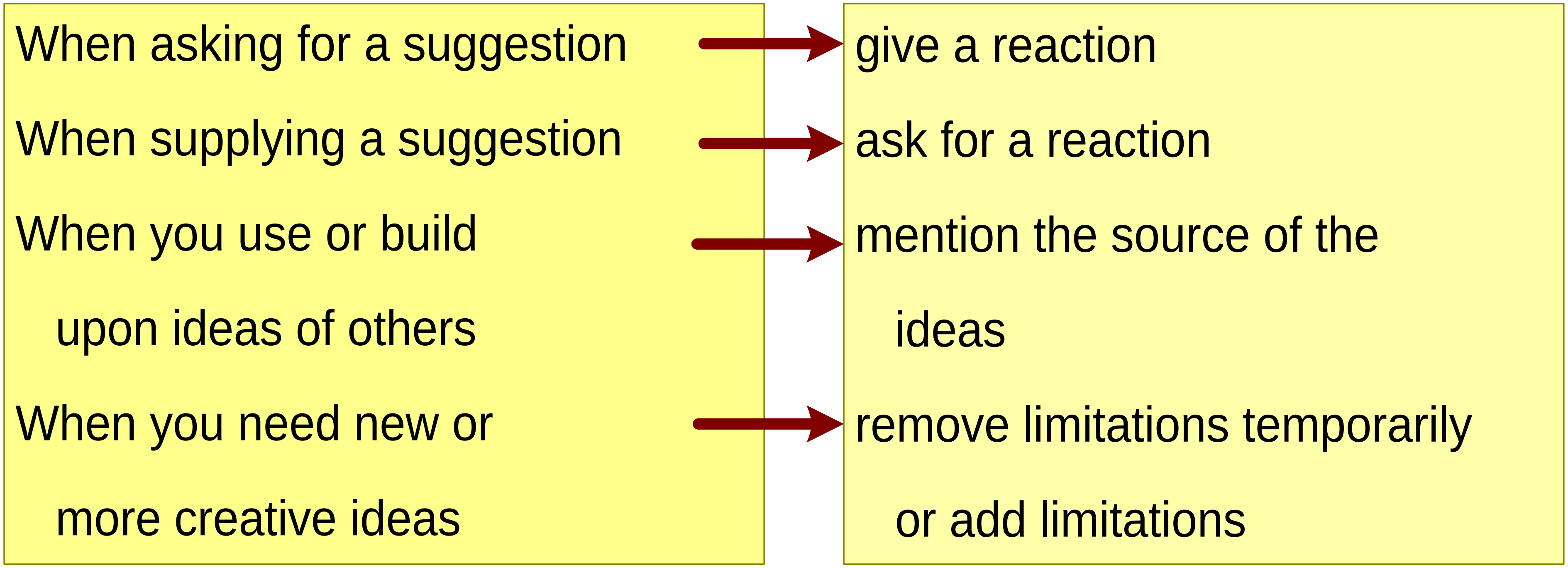
When you open a conversation

formulate the purpose

When you finish the conversation

summarize the agreements and the actionplan

Searching for Ideas



When asking for a suggestion	give a reaction
When supplying a suggestion	ask for a reaction
When you use or build upon ideas of others	mention the source of the ideas
When you need new or more creative ideas	remove limitations temporarily or add limitations

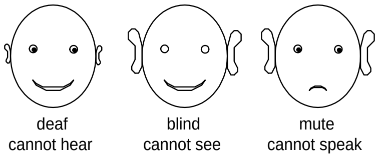
Human Side: Team Work

by *Gerrit Muller* Embedded Systems Institute
e-mail: gerrit.muller@embeddedsystems.nl
www.gaudisite.nl

Abstract

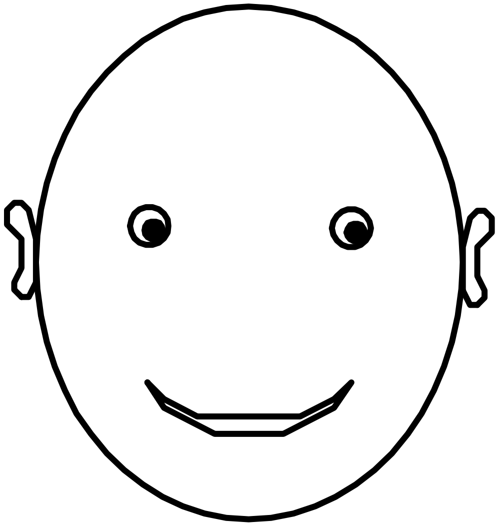
The creation of products requires many different people to cooperate. The work is often organized in teams. The team members have complimentary skills and knowledge. In many management courses the need to design teams is emphasized. Unfortunately, often these recommendations are ignored. We re-iterate in this paper the rationale for teams and the recommendations for designing the team itself.

6th March 2009
status: preliminary draft
version: 0.1

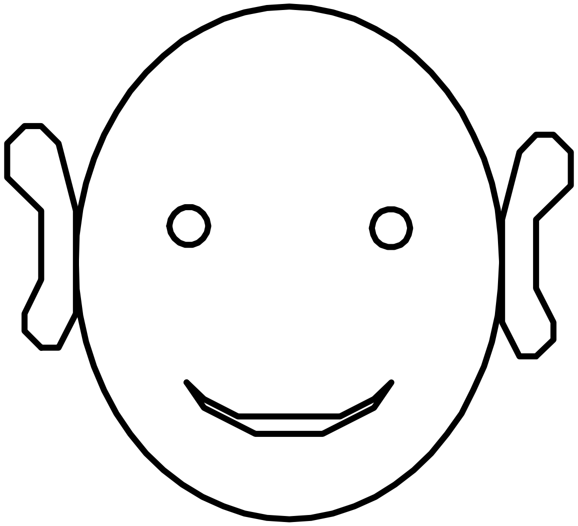


but in the team two can hear, two can see, and two can speak

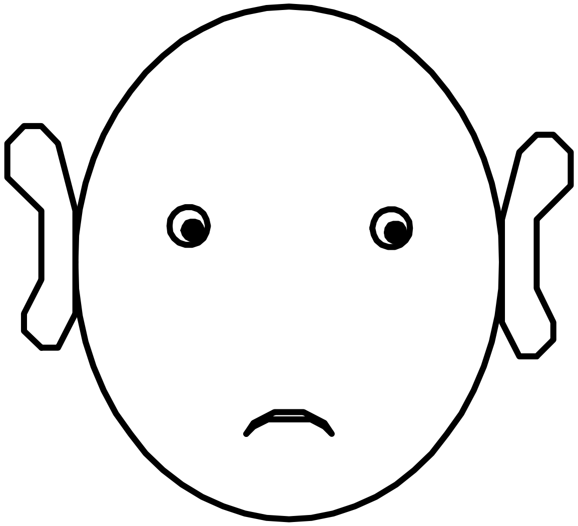
Teams consist of complementary people



deaf
cannot hear



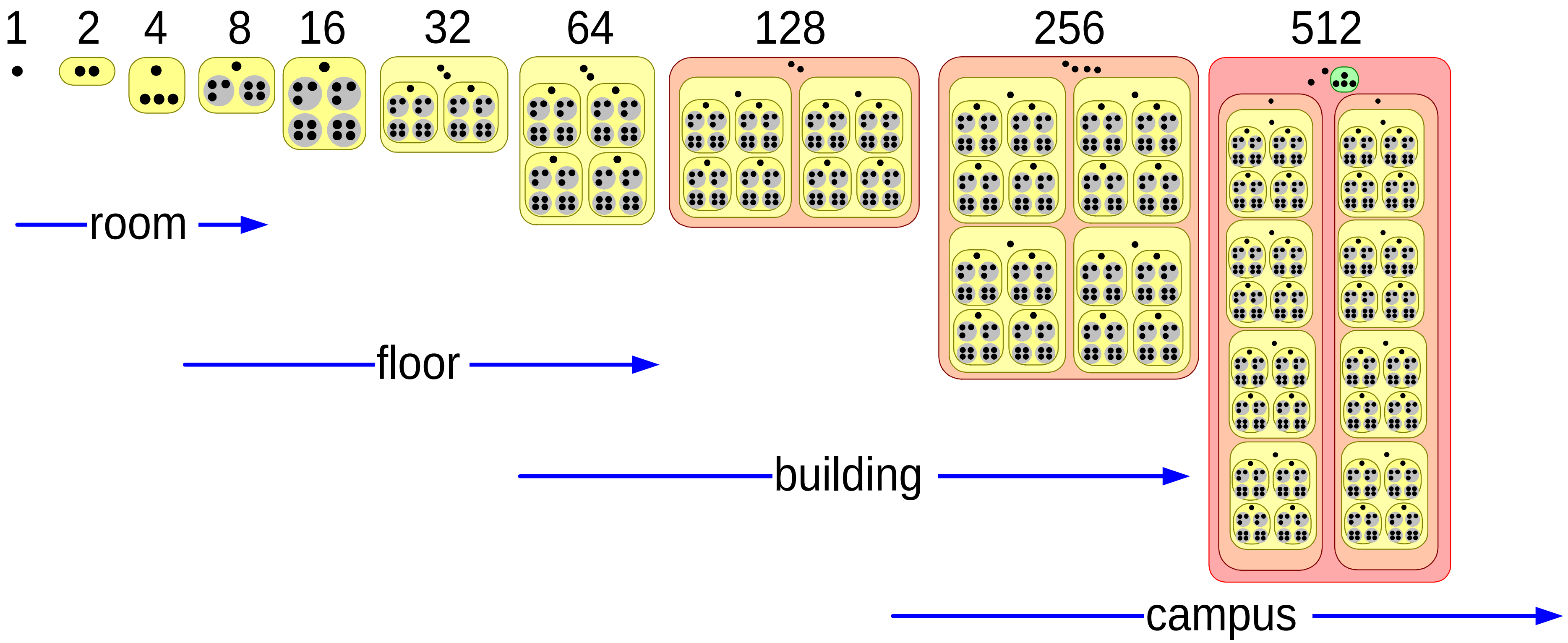
blind
cannot see



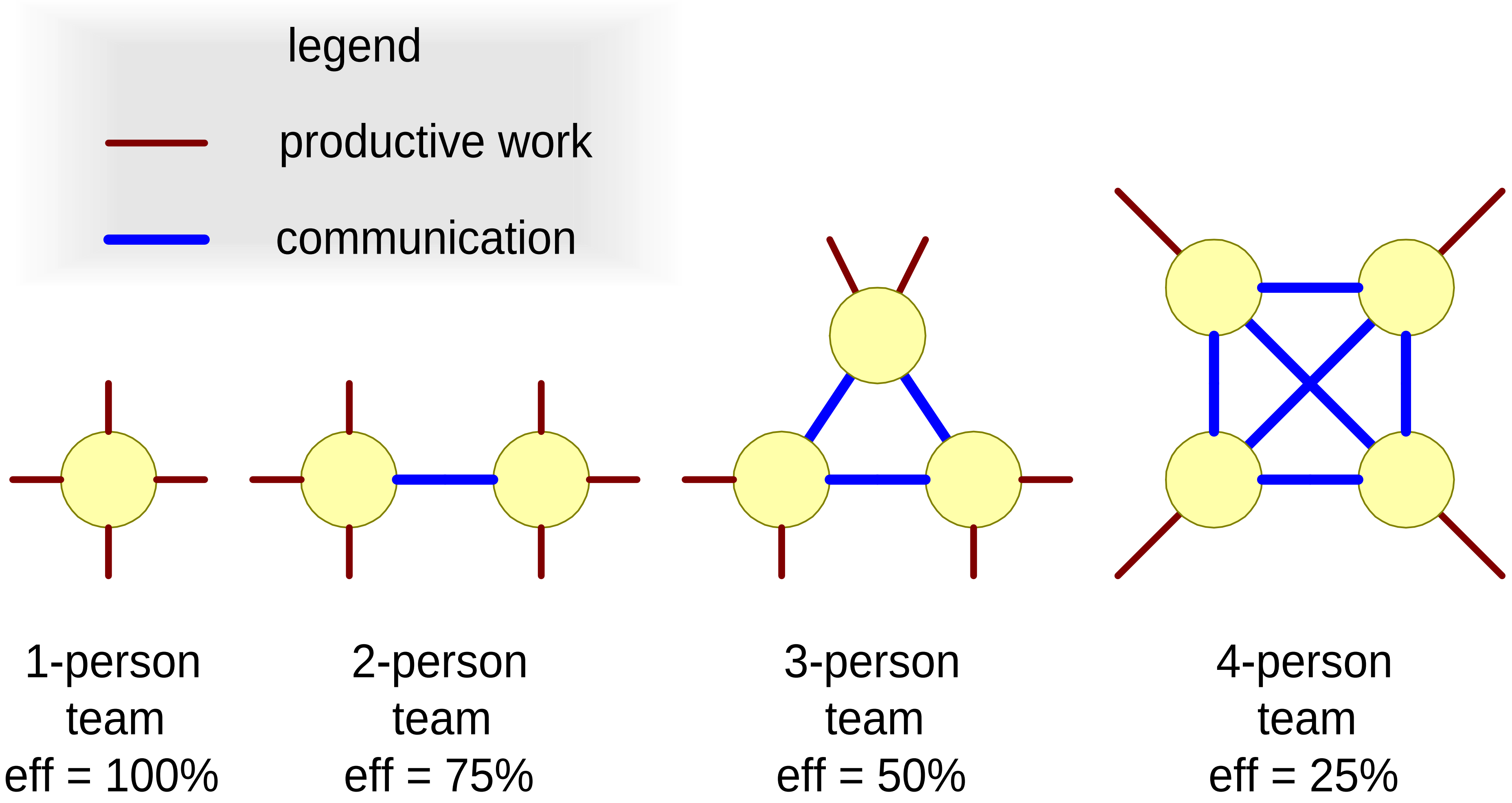
mute
cannot speak

but in the team two can hear,
two can see, and two can speak

Organization size and teams



Very simplistic team model



legend

productive work

communication

1-person team eff = 100%

2-person team eff = 75%

3-person team eff = 50%

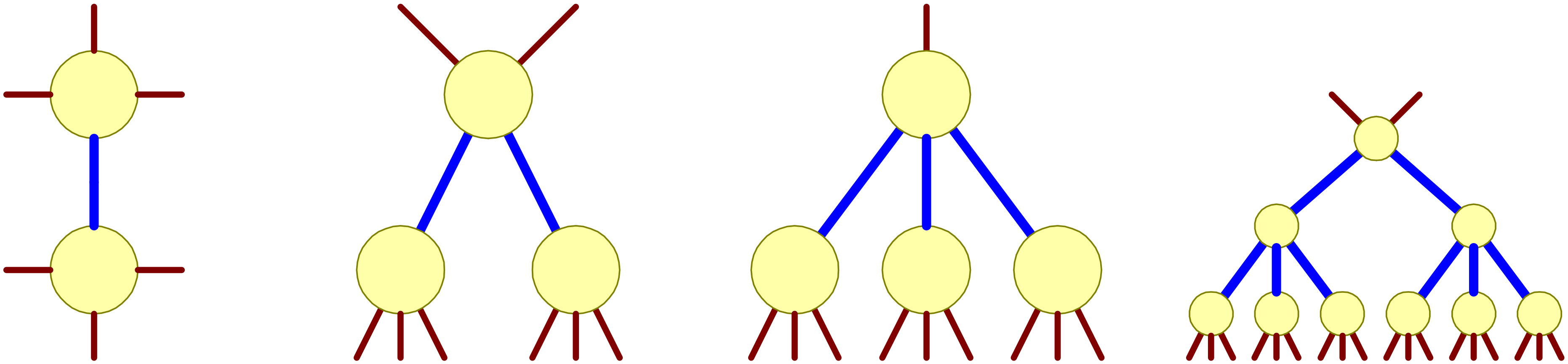
4-person team eff = 25%

Hierarchical simplistic team model

legend

— productive work

▬ communication



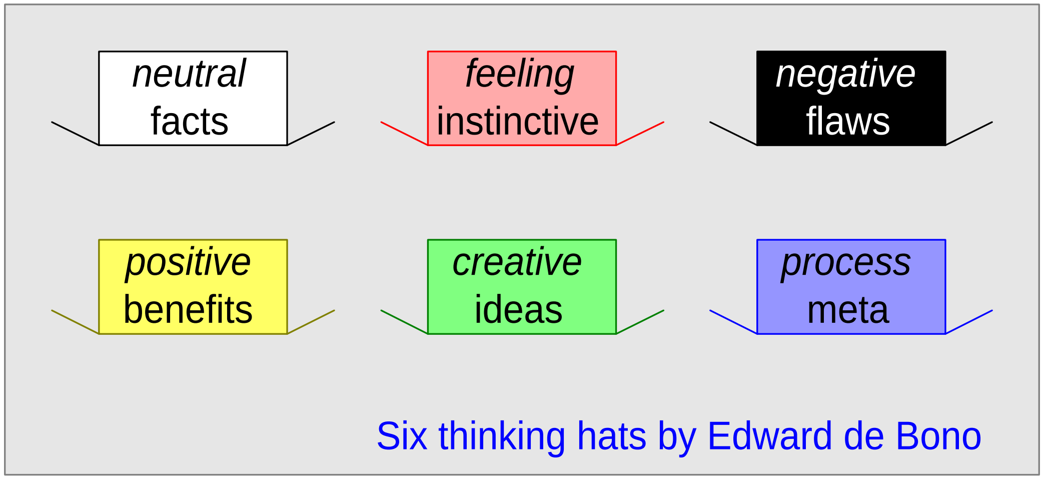
2-person
team
eff = 75%

3-person
team
eff = 66%

4-person
team
eff = 62.5%

9-person
team
eff ~= 56%

Many personality and role models are available



Six thinking hats by Edward de Bono

E Extraversion	⟷	Introversion **I**
S Sensing	⟷	iNtuition **N**
T Thinking	⟷	Feeling **F**
J Judging	⟷	Perceiving **P**

Myers-Briggs Type Indicators

plant
creative

team worker
co-operative, averts friction

implementer
disciplined, conservative, do-er

resource investigator
enthusiatic communicator

shaper
driver, dynamic

completer finisher
conscientious, painstaking

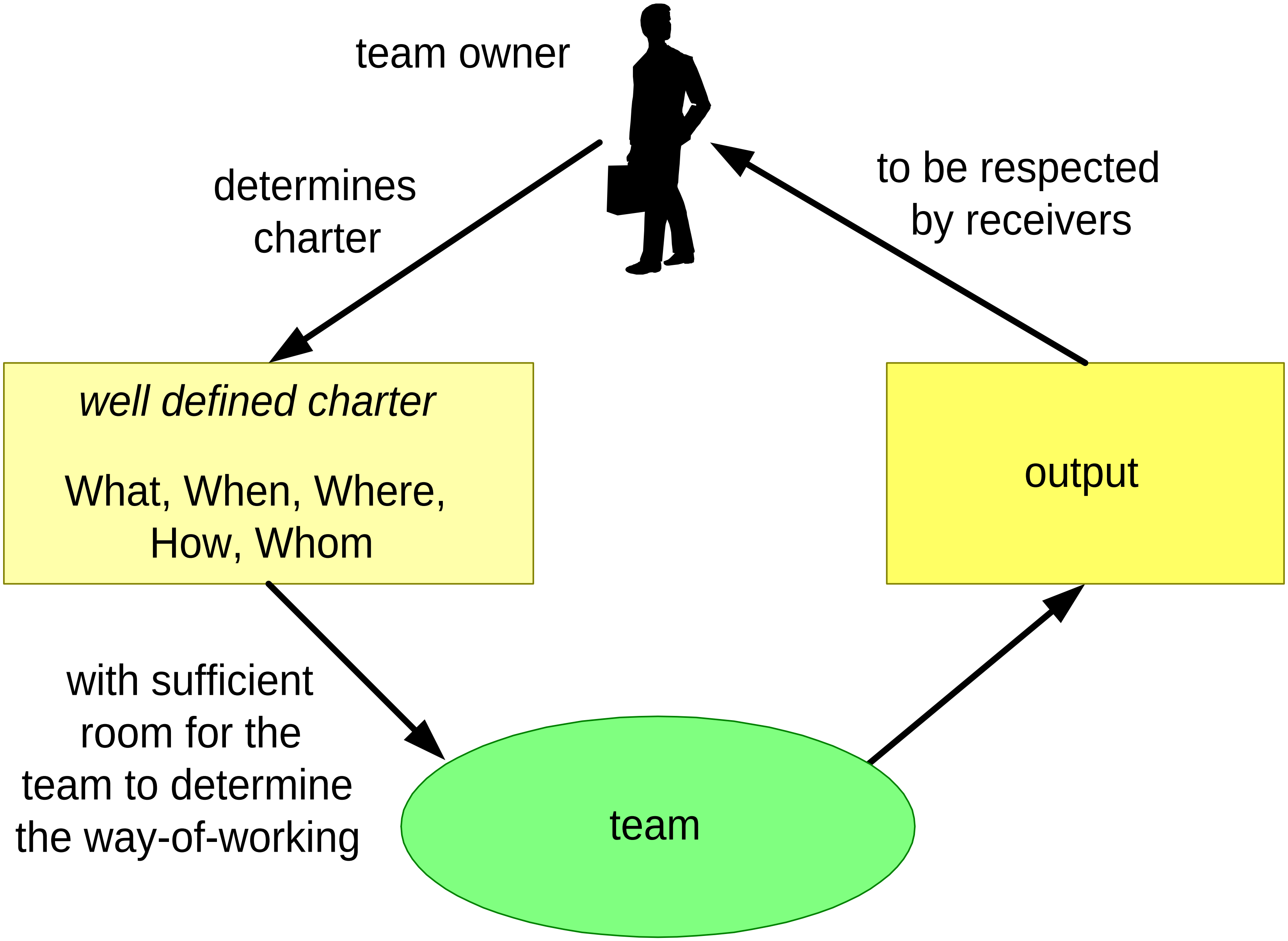
co-ordinator
mature, chairman

monitor evaluator
sober, analytical

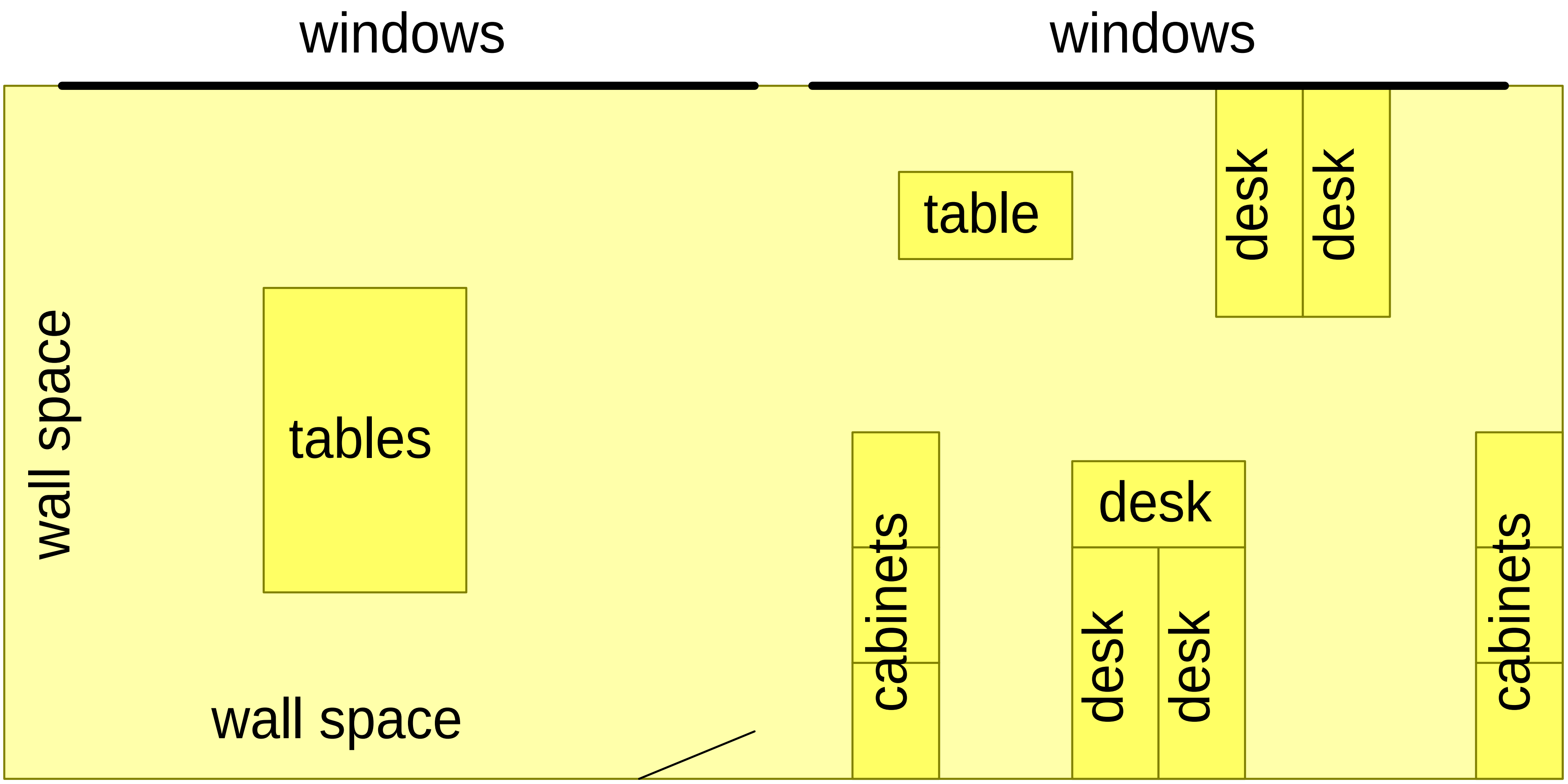
specialist
single-minded, rare skills

Belbin's team roles

Process of creating and using a team

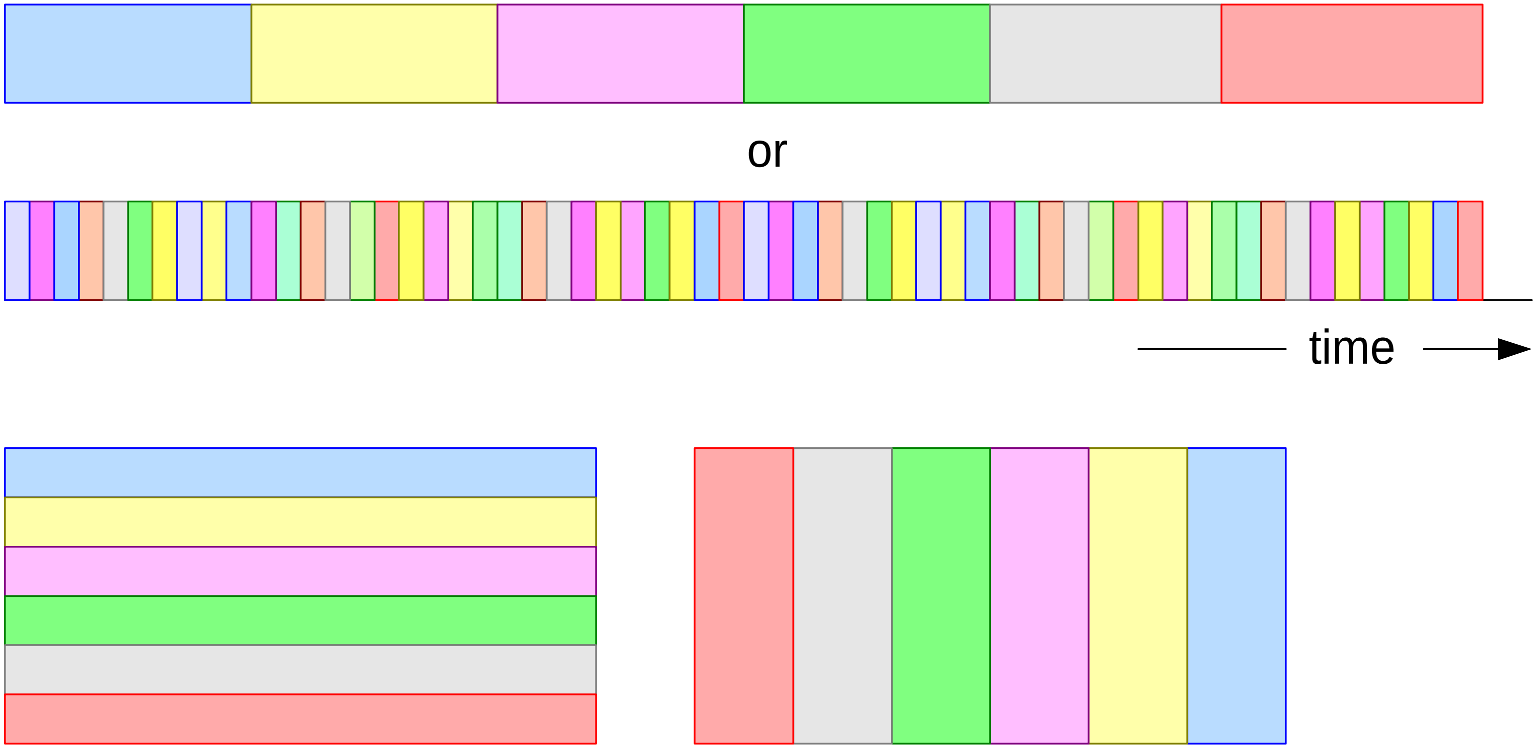

team owner
determines charter
to be respected by receivers
well defined charter
What, When, Where, How, Whom
output
with sufficient room for the team to determine the way-of-working
team

“War Room” is very effective



Concurrency and Fragmentation lower efficiency

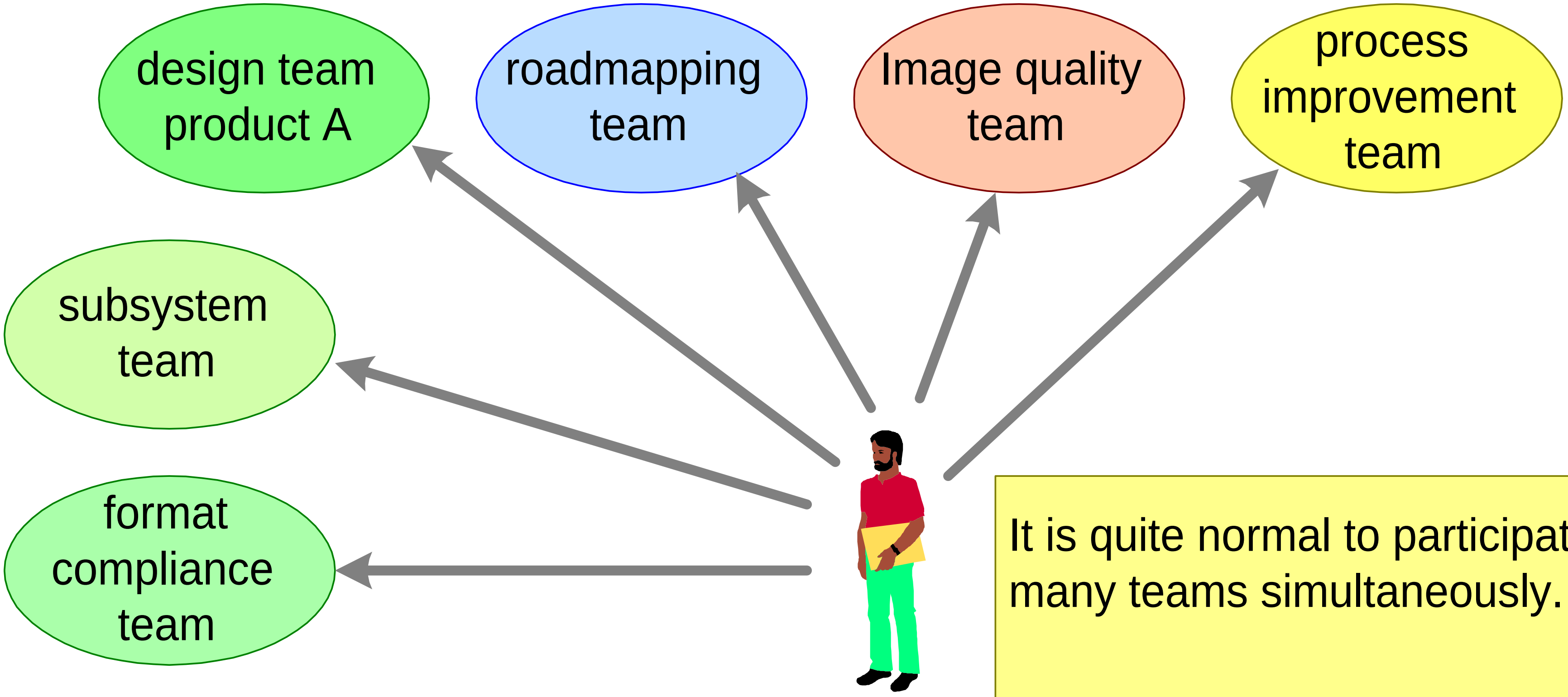
How many (semi-)concurrent tasks can a person handle?
Working in burst-mode (concentrating on one task for one day, week or month) can increase efficiency.



six tasks in parallel:
all results are late

six tasks sequential
first result in 1/6 of time!

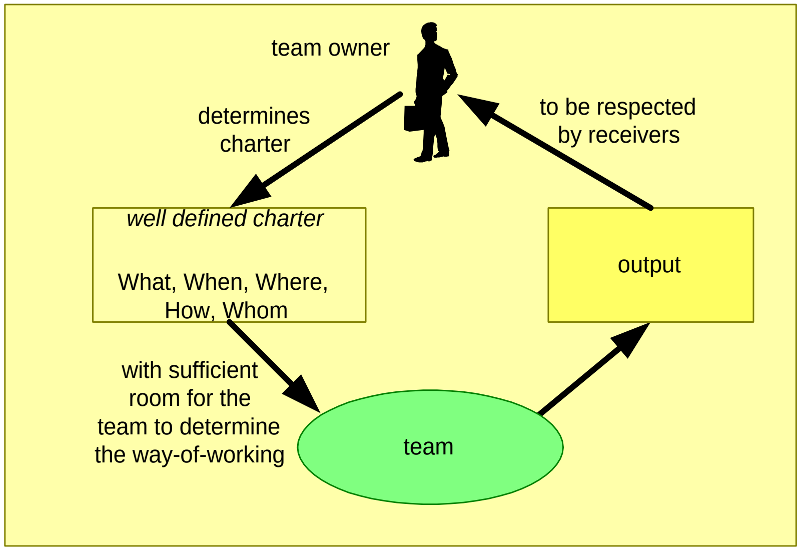
One person will be member of multiple teams



It is quite normal to participate in many teams simultaneously.

However, a team can only function if the members are sufficiently available!

Critical Success Factors for teams

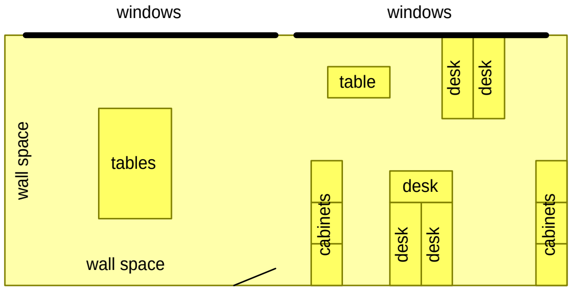

team owner
determines charter
to be respected by receivers
well defined charter
What, When, Where, How, Whom
output
with sufficient room for the team to determine the way-of-working
team

well defined charter

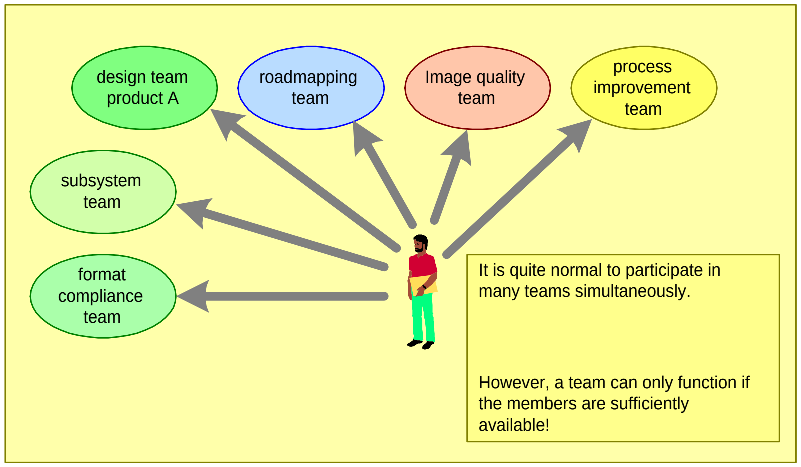
clear owner of the result

respect for the output of the team

freedom of way-of-working



housing and location



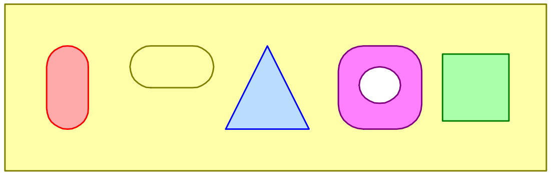
It is quite normal to participate in many teams simultaneously.

However, a team can only function if the members are sufficiently available!

availability of the team members

plant	*team worker*	*implementer*
resource investigator	*shaper*	*completer finisher*
co-ordinator	*monitor evaluator*	*specialist*

complementary roles



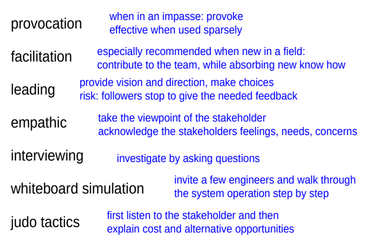
diversity, pluriformity

Architecting Interaction Styles

by *Gerrit Muller*   Embedded Systems Institute
e-mail: `gerrit.muller@embeddedsystems.nl`
`www.gaudisite.nl`

Abstract

A system architects needs skills to apply different interactions styles, depending on the circumstances. This document discusses the following interaction styles: provocation, facilitation, leading, empathic, interviewing, white board simulation, and judo tactics.



provocation	when in an impasse: provoke effective when used sparsely
facilitation	especially recommended when new in a field: contribute to the team, while absorbing new know how
leading	provide vision and direction, make choices risk: followers stop to give the needed feedback
empathic	take the viewpoint of the stakeholder acknowledge the stakeholders feelings, needs, concerns
interviewing	investigate by asking questions
whiteboard simulation	invite a few engineers and walk through the system operation step by step
judo tactics	first listen to the stakeholder and then explain cost and alternative opportunities

Distribution

6th March 2009
status: preliminary draft
version: 0.1

Architecting Styles

provocation

when in an impasse: provoke
effective when used sparsely

facilitation

especially recommended when new in a field:
contribute to the team, while absorbing new know how

leading

provide vision and direction, make choices
risk: followers stop to give the needed feedback

empathic

take the viewpoint of the stakeholder
acknowledge the stakeholders feelings, needs, concerns

interviewing

investigate by asking questions

whiteboard simulation

invite a few engineers and walk through
the system operation step by step

judo tactics

first listen to the stakeholder and then
explain cost and alternative opportunities

Function Profiles; The Sheep with Seven Legs

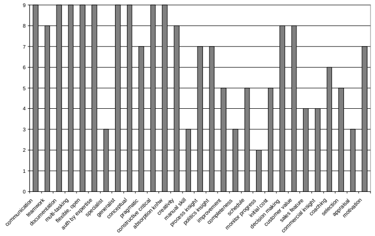
by *Gerrit Muller* Embedded Systems Institute
e-mail: `gerrit.muller@embeddedsystems.nl`
`www.gaudisite.nl`

Abstract

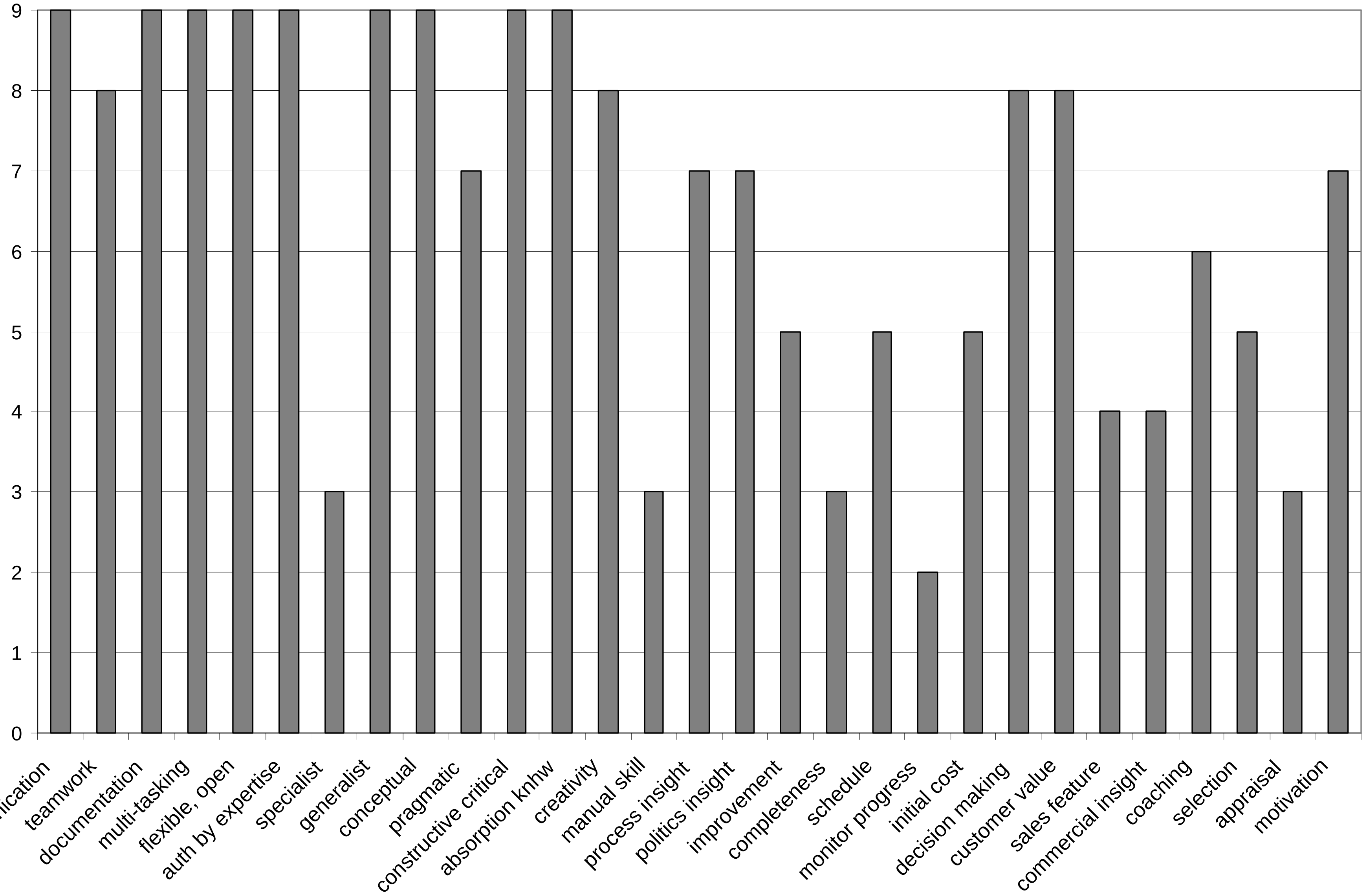
The profile of a system architect is quantified for a large list of system architect related characteristics. For comparison the function profiles of related functions are given as well. This profile is based on personal observations and experience.

Distribution

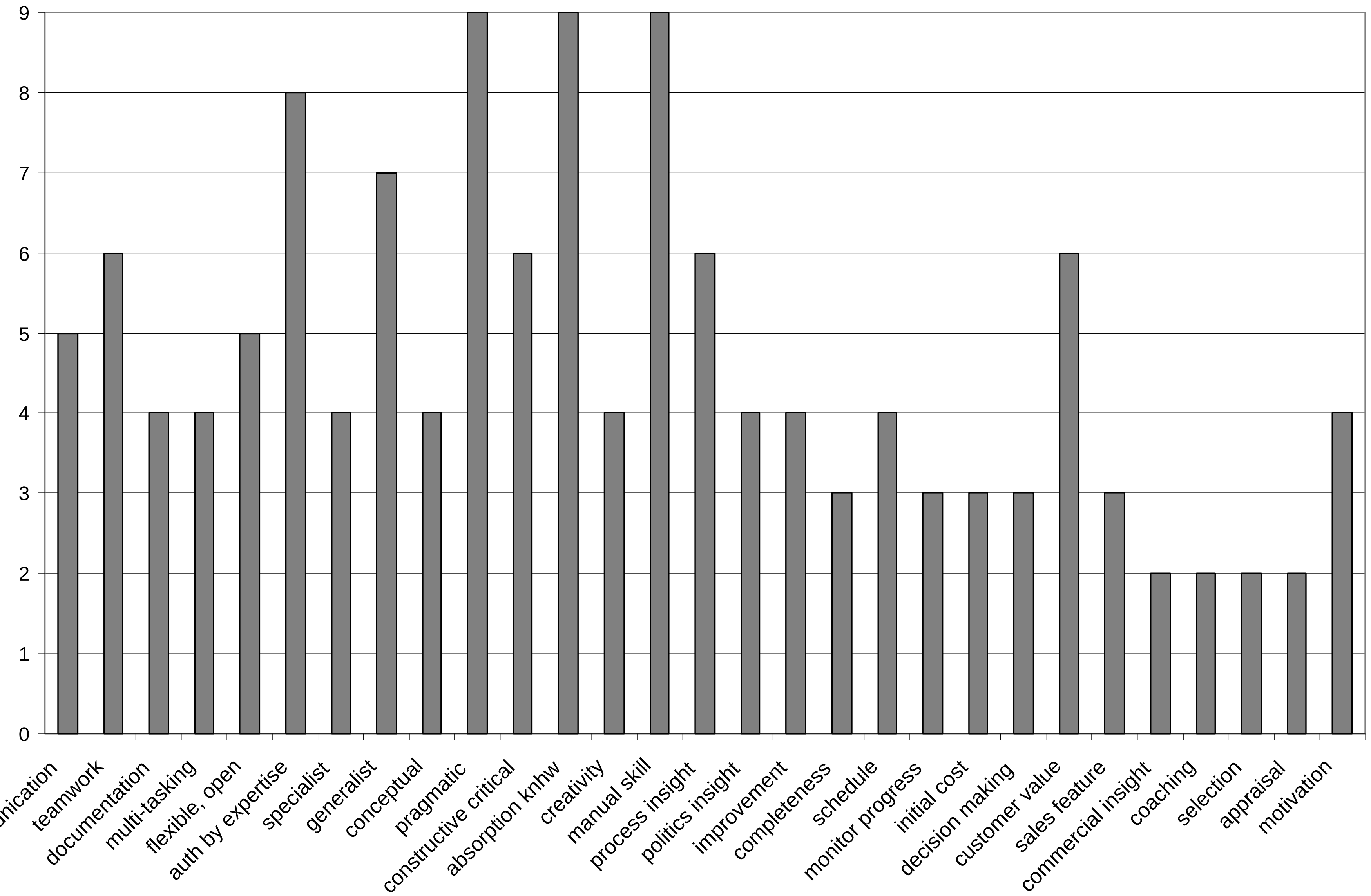
6th March 2009
status: preliminary draft
version: 0.5



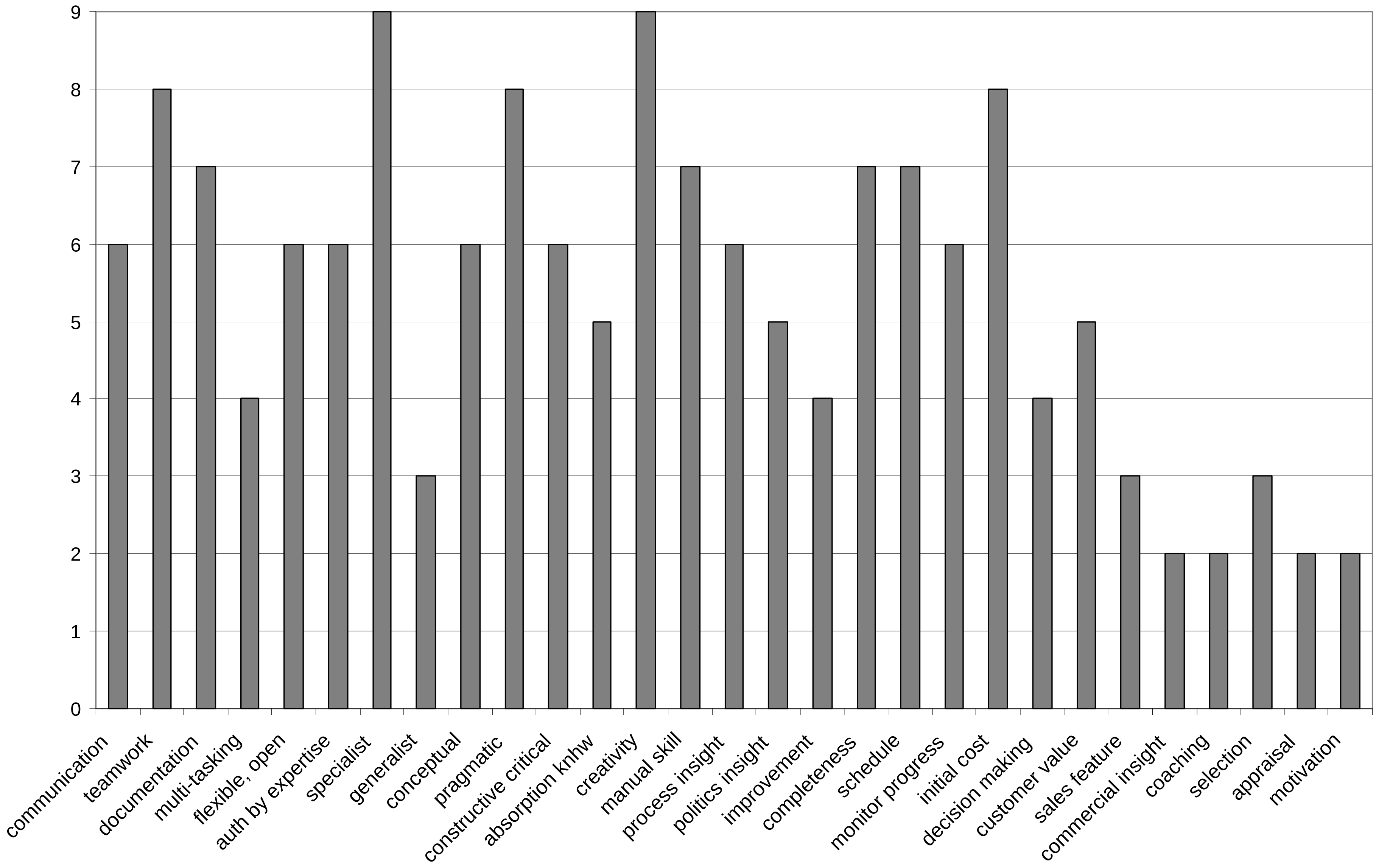
System Architect



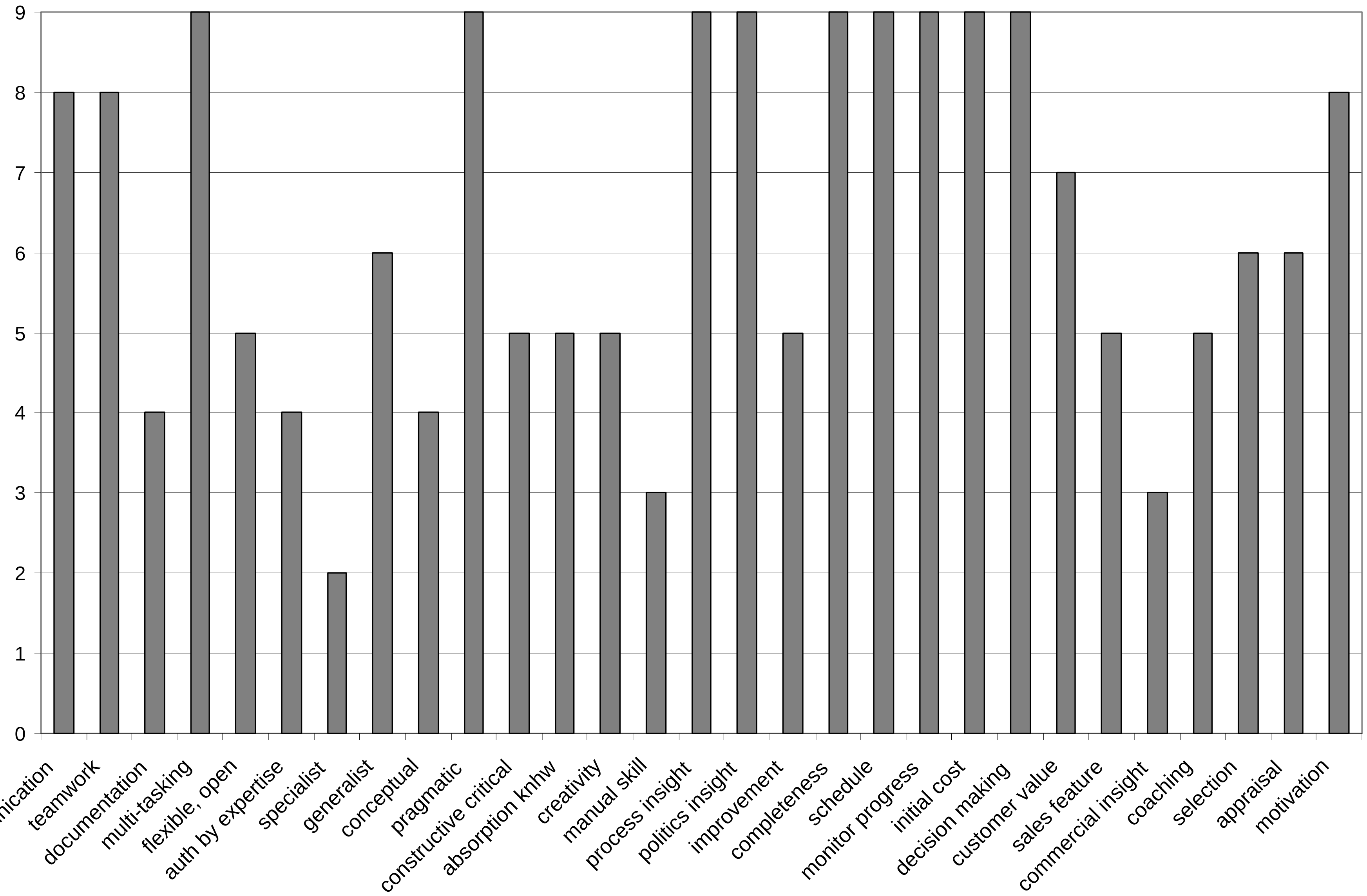
Test Engineer



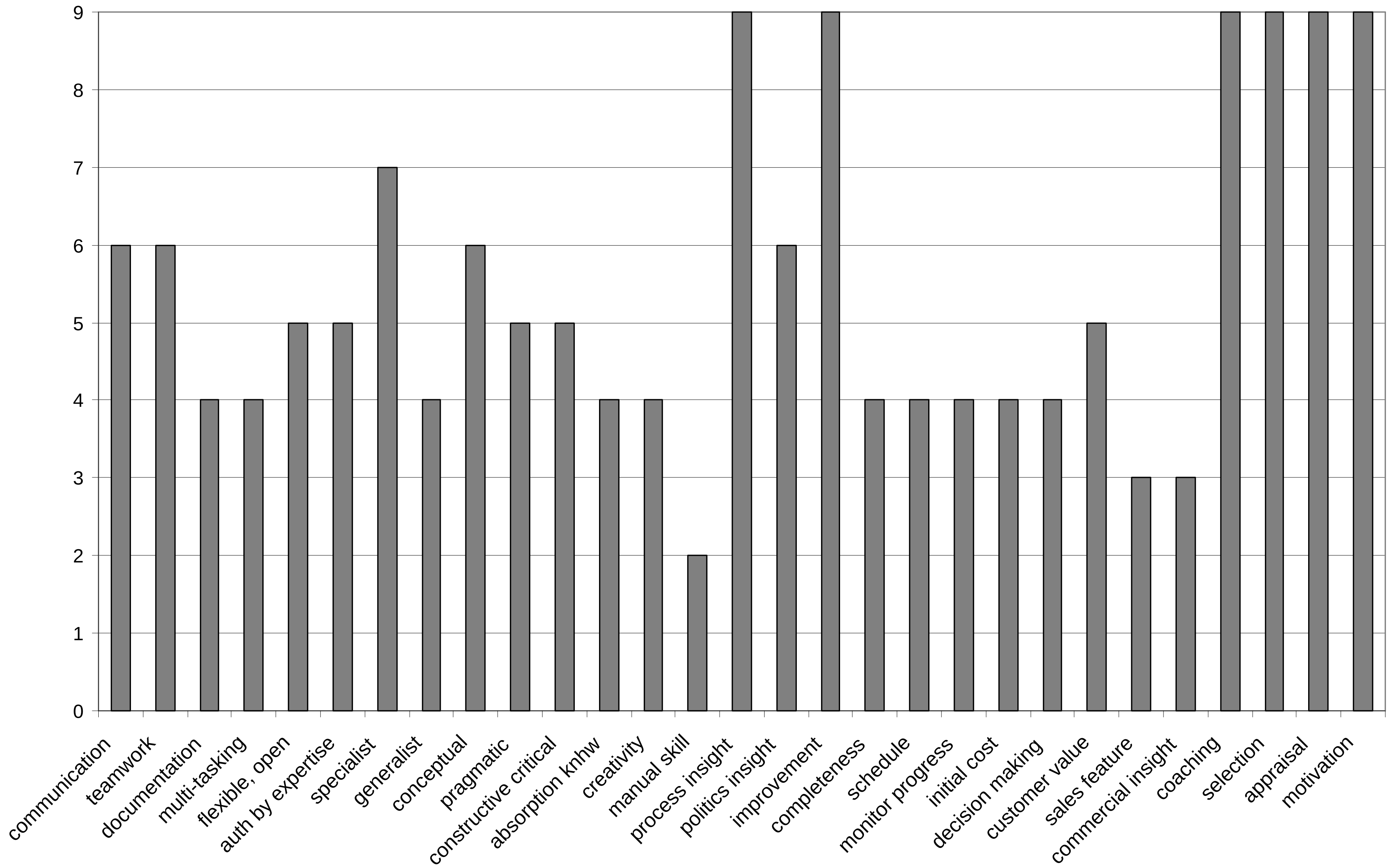
Developer


9
8
7
6
5
4
3
2
1
0
communication
teamwork
documentation
multi-tasking
flexible, open
auth by expertise
specialist
generalist
conceptual
pragmatic
constructive critical
absorption knhw
creativity
manual skill
process insight
politics insight
improvement
completeness
schedule
monitor progress
initial cost
decision making
customer value
sales feature
commercial insight
coaching
selection
appraisal
motivation

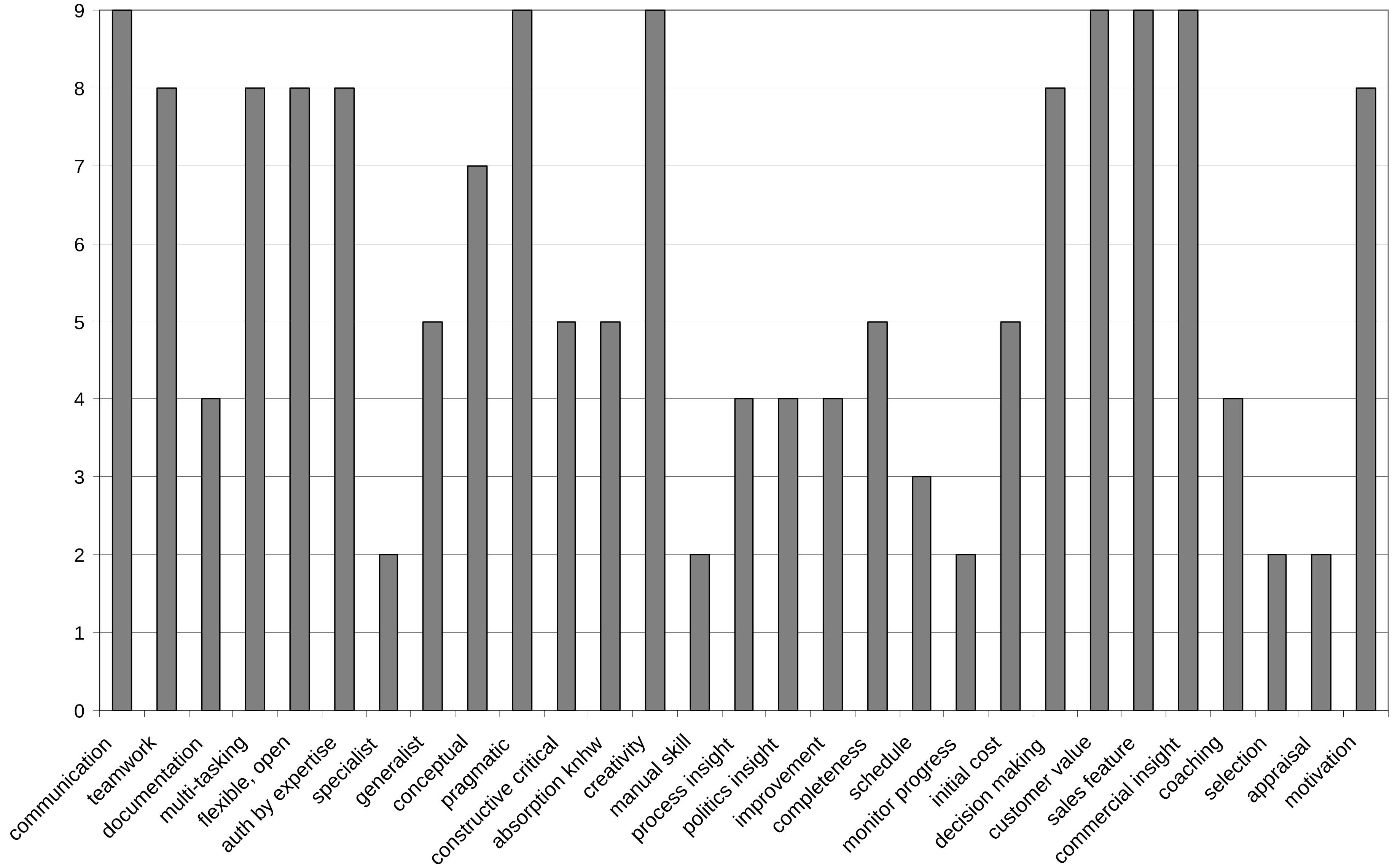
Operational Leader



Line Manager



Commercial Manager



The numbers behind the bars

	interpersonal skills								reasoning power						process			project mngmt					commercial			HRM			
	communication	teamwork	documentation	multi-tasking	flexible, open	auth by expertise	specialist	generalist	conceptual	pragmatic	constructive critical	absorption knhw	creativity	manual skill	process insight	politics insight	improvement	completeness	schedule	monitor progress	initial cost	decision making	customer value	sales feature	commercial insight	coaching	selection	appraisal	motivation
system architect	9	8	9	9	9	9	3	9	9	7	9	9	8	3	7	7	5	3	5	2	5	8	8	4	4	6	5	3	7
test engineer	5	6	4	4	5	8	4	7	4	9	6	9	4	9	6	4	4	3	4	3	3	3	6	3	2	2	2	2	4
developer	6	8	7	4	6	6	9	3	6	8	6	5	9	7	6	5	4	7	7	6	8	4	5	3	2	2	3	2	2
operational leader	8	8	4	9	5	4	2	6	4	9	5	5	5	3	9	9	5	9	9	9	9	9	7	5	3	5	6	6	8
line manager	6	6	4	4	5	5	7	4	6	5	5	4	4	2	9	6	9	4	4	4	4	4	5	3	3	9	9	9	9
commercial manager	9	8	4	8	8	8	2	5	7	9	5	5	9	2	4	4	4	5	3	2	5	8	9	9	9	4	2	2	8

Short introduction to basic “CAFCR” model

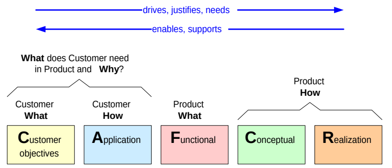
by *Gerrit Muller* Embedded Systems Institute
e-mail: `gerrit.muller@embeddedsystems.nl`
`www.gaudisite.nl`

Abstract

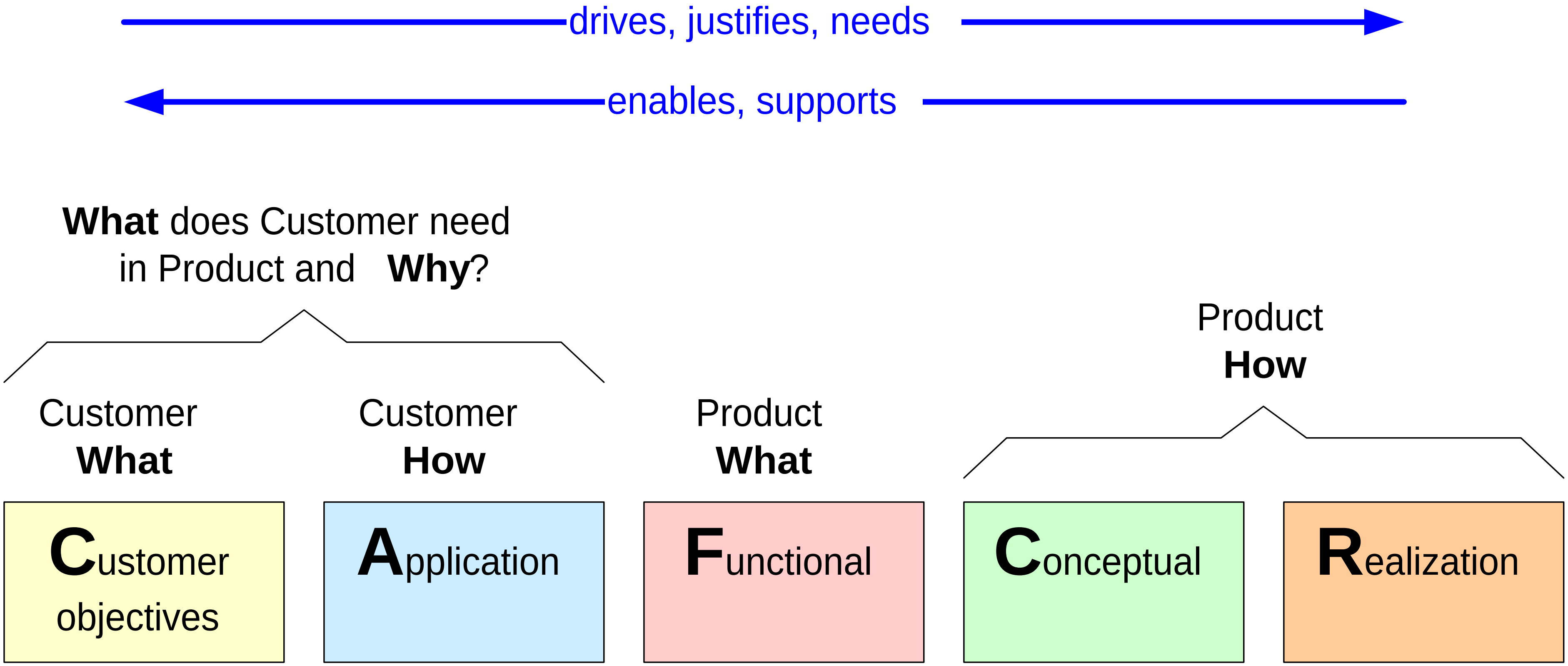
The basic “CAFCR” reference model is described, which is used to describe a system in relation to its context. The main stakeholder in the context is the customer. The question “Who is the customer?” is addressed.

Distribution

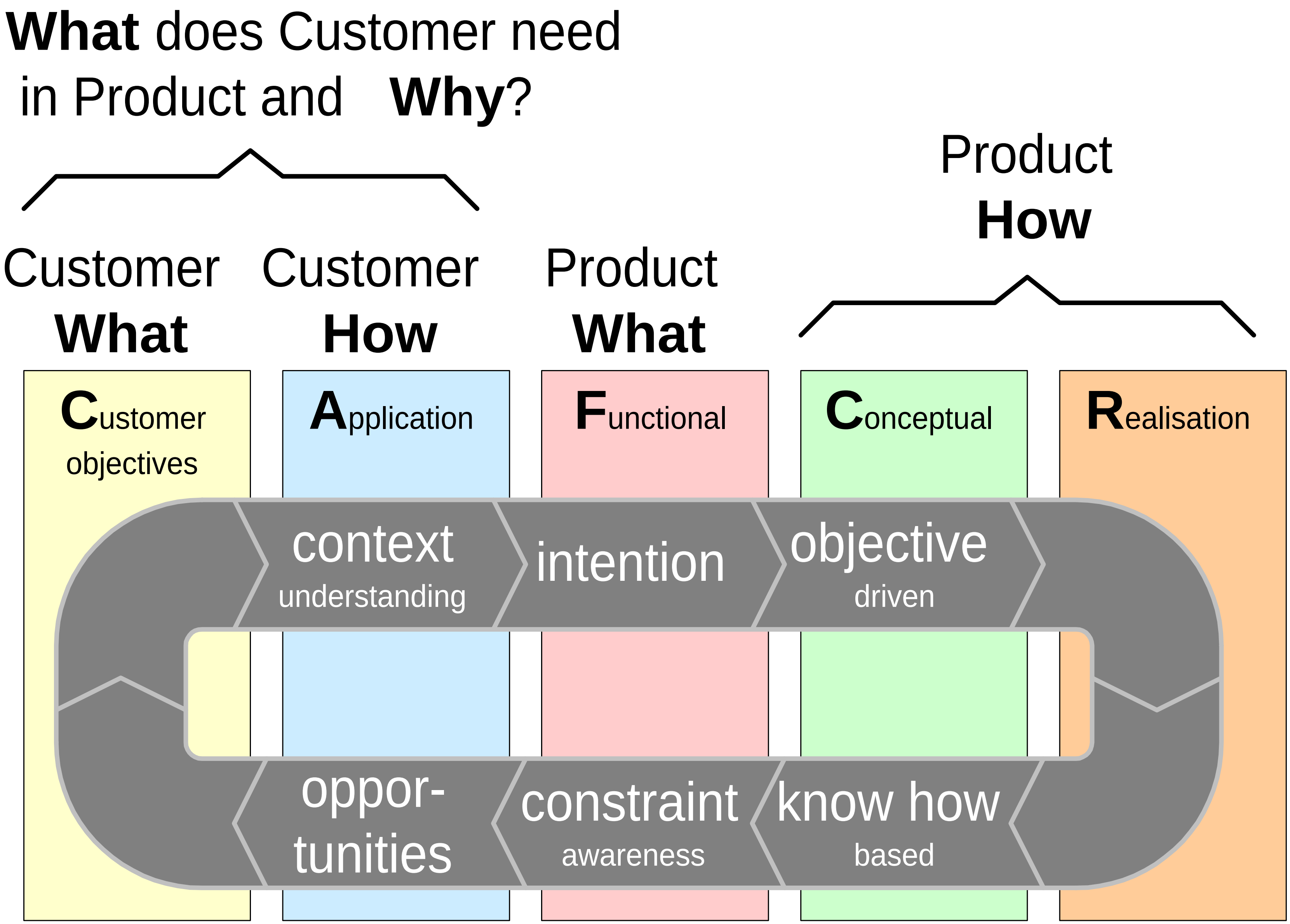
6th March 2009
status: preliminary draft
version: 0.3



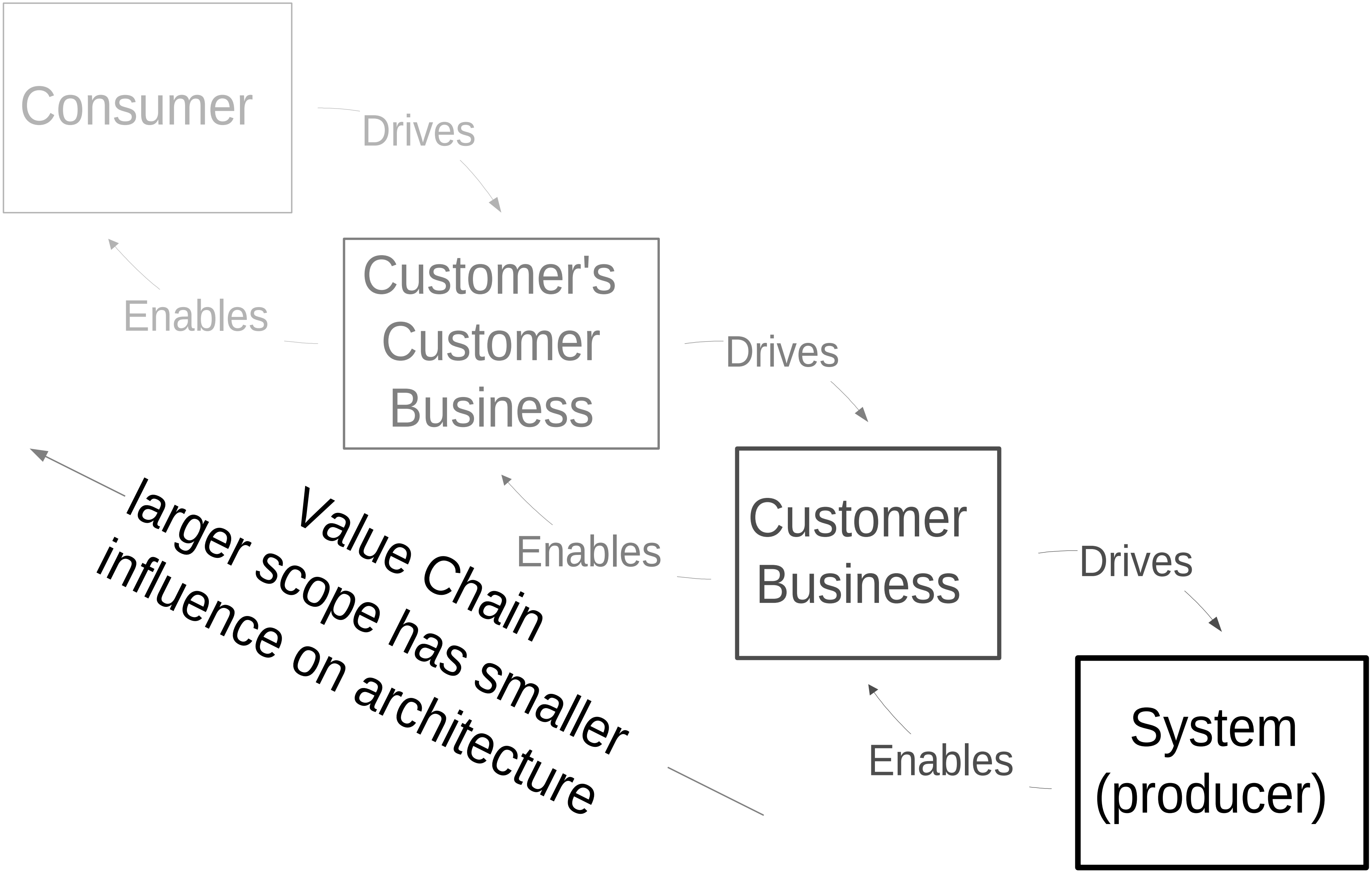
The “CAFCR” model



Integrating CAFCR



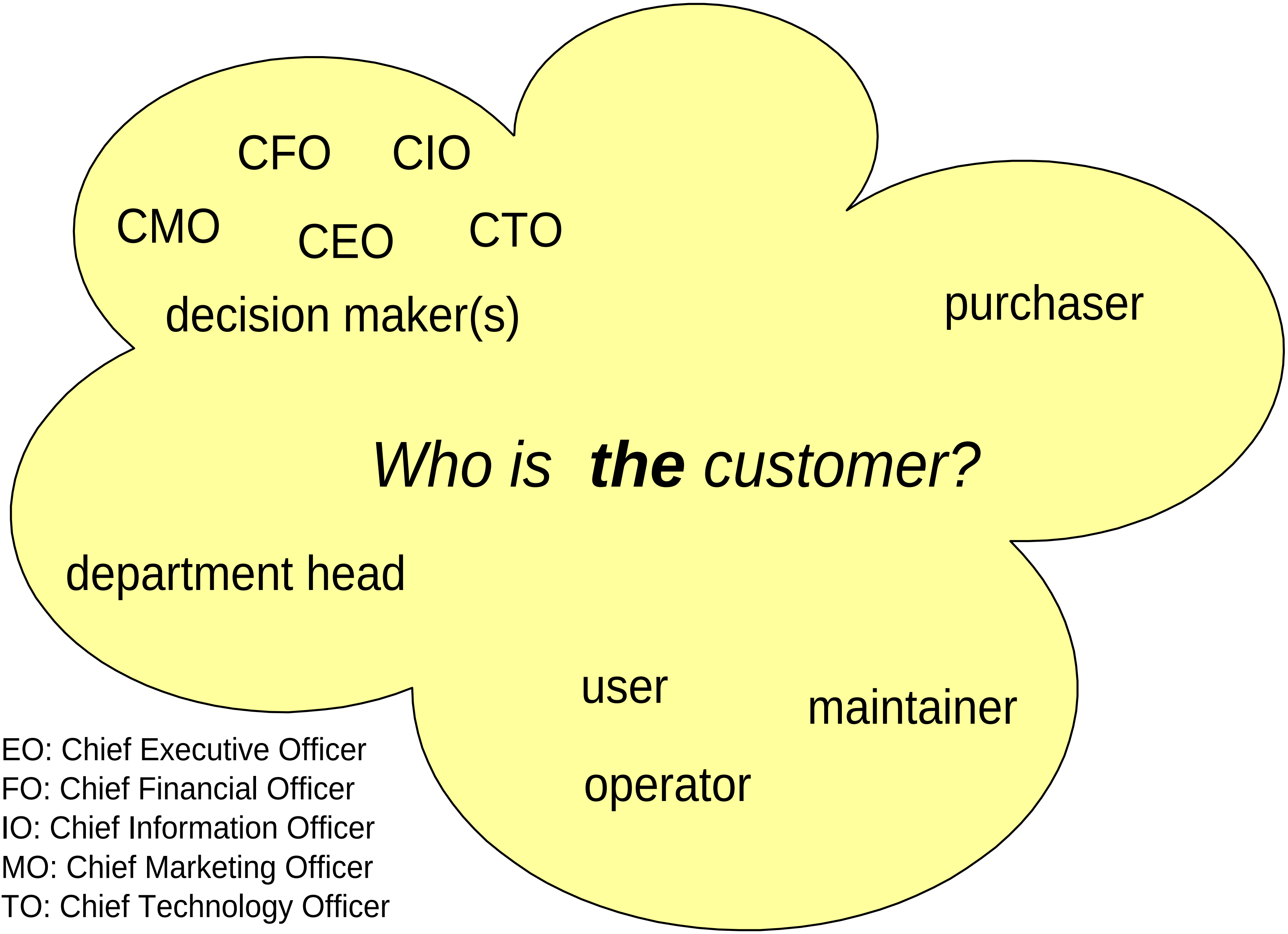
CAFCR can be applied recursively



Market segmentation

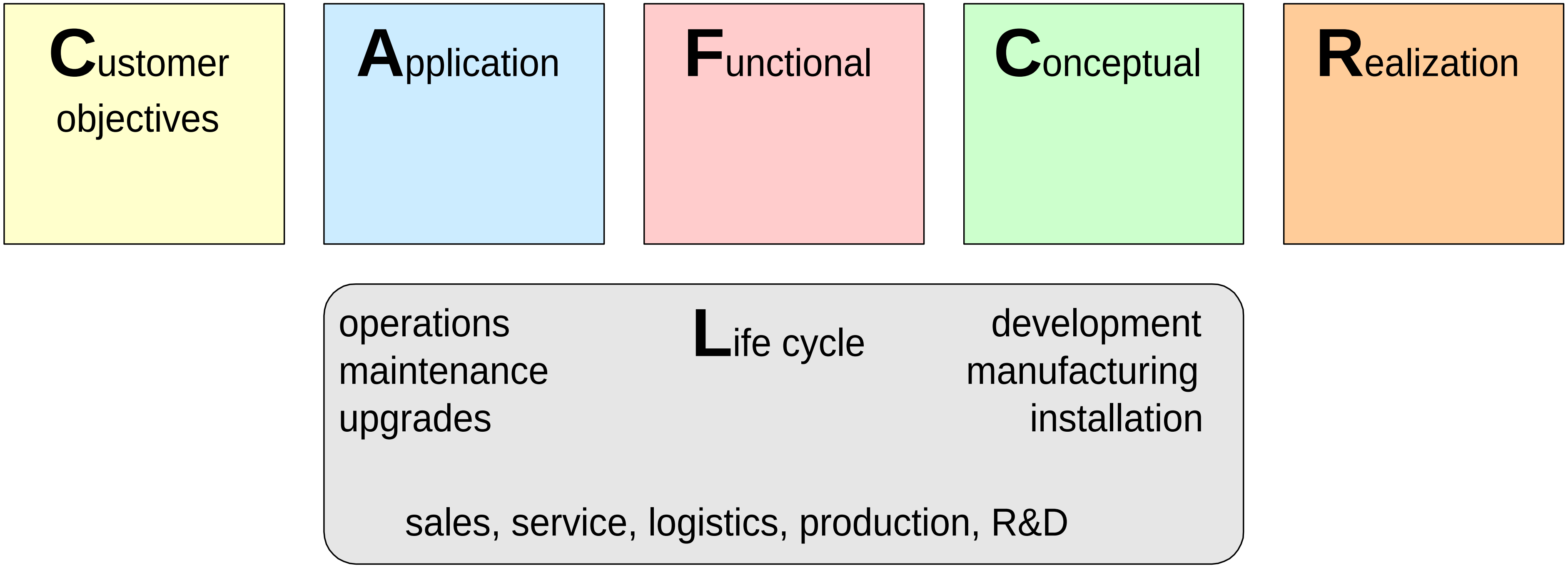
segmentation axis	examples
geographical	USA, UK, Germany, Japan, China
business model	profit, non profit
economics	high end versus cost constrained
consumers	youth, elderly
outlet	retailer, provider, OEM, consumer direct

Example of a small buying organization



CEO: Chief Executive Officer
CFO: Chief Financial Officer
CIO: Chief Information Officer
CMO: Chief Marketing Officer
CTO: Chief Technology Officer

CAFCR+ model; Life Cycle View



Security as example through all views

	Customer objectives	**A**pplication	**F**unctional	**C**onceptual	**R**ealization
desired characteristics, specifications & mechanisms	sensitive information trusted	selection classification people information authentication badges passwords locks / walls guards administrators	functions for administration authentication intrusion detection logging quantification	cryptography firewall security zones authentication registry logging	***specific*** algorithms interfaces libraries servers storage protocols
threats	 not trusted	social contacts open passwords blackmail burglary fraud unworkable procedures	missing functionality wrong quantification	holes between concepts	bugs buffer overflow non encrypted storage poor exception handling

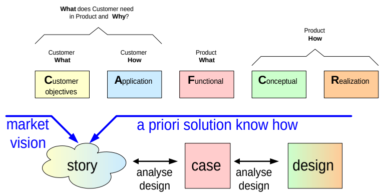
Story How To

by *Gerrit Muller*   Embedded Systems Institute
e-mail: `gerrit.muller@embeddedsystems.nl`
`www.gaudisite.nl`

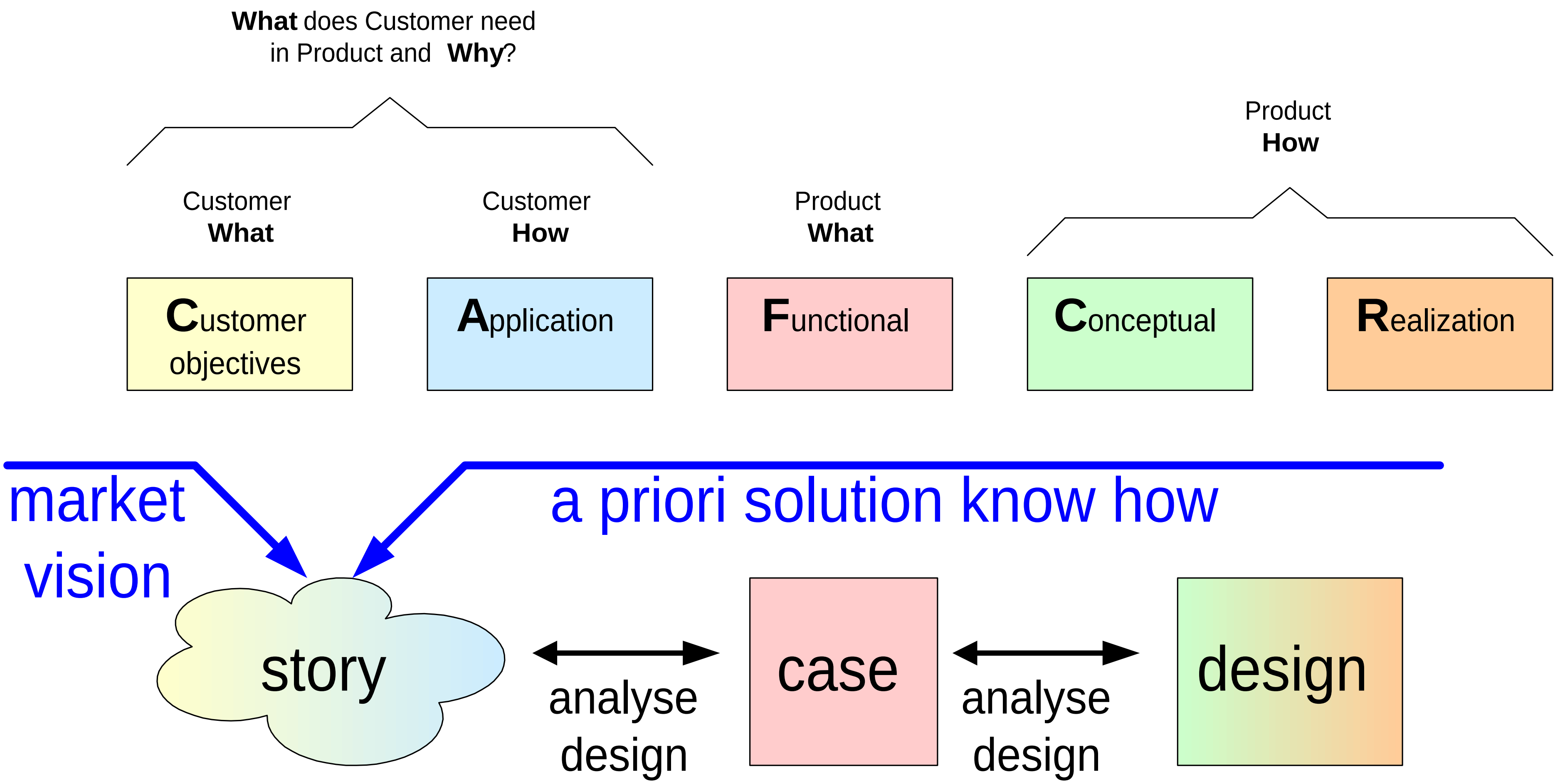
Abstract

A story is an easily accessible story or narrative to make an application live. A good story is highly specific and articulated entirely in the problem domain: the native world of the users. An important function of a story is to enable specific (*quantified, relevant, explicit*) discussions.

6th March 2009
status: draft
version: 0.4



From story to design



What does Customer need
in Product and **Why**?

Product
How

Customer
What

Customer
How

Product
What

Customer objectives

Application

Functional

Conceptual

Realization

market vision

a priori solution know how

story

analyse design

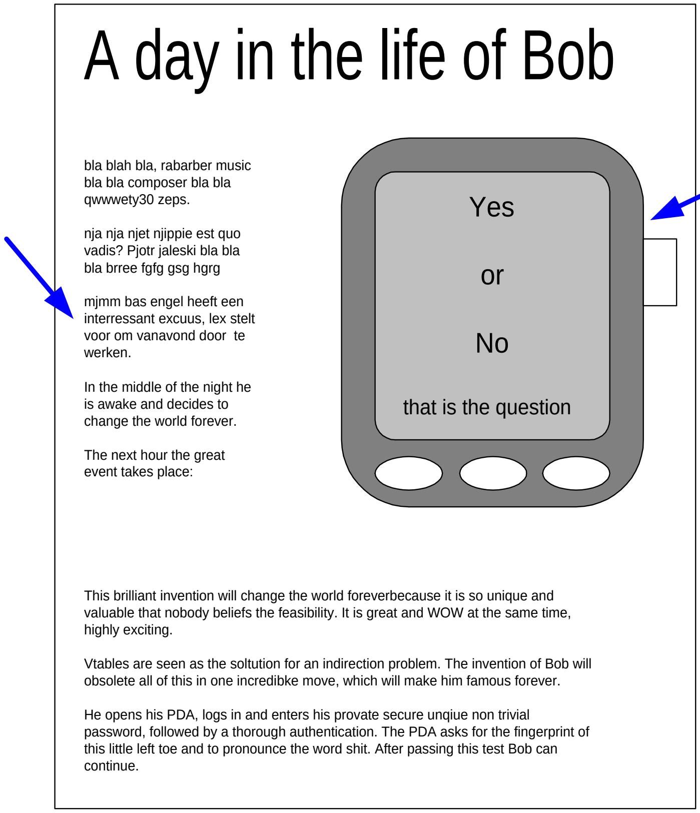
case

analyse design

design

Example story layout

ca. half a page of plain English text



A day in the life of Bob

bla blah bla, rabarber music bla bla composer bla bla qwwwety30 zeps.

nja nja njet njippie est quo vadis? Pjotr jaleski bla bla bla brree fgfg gsg hgrg

mjmm bas engel heeft een interressant excuus, lex stelt voor om vanavond door te werken.

In the middle of the night he is awake and decides to change the world forever.

The next hour the great event takes place:

This brilliant invention will change the world foreverbecause it is so unique and valuable that nobody beliefs the feasibility. It is great and WOW at the same time, highly exciting.

Vtables are seen as the soltution for an indirection problem. The invention of Bob will obsolete all of this in one incredibke move, which will make him famous forever.

He opens his PDA, logs in and enters his provate secure unqiue non trivial password, followed by a thorough authentication. The PDA asks for the fingerprint of this little left toe and to pronounce the word shit. After passing this test Bob can continue.

draft or sketch of some essential appliance

Points of attention

- purpose
- scope
- viewpoint, stakeholders
- visualization
- size (max 1 A4)
- recursive decomposition, refinement

Criterions for a good story

Customer objectives
Application

- accessible, understandable

"Do you see it in front of you?"

Customer objectives
Application

- valuable, appealing

attractive, important
"Are the customers queuing for this?"

Conceptual
Realization

- critical, challenging

"What is difficult in the realization?"
"What do you learn w.r.t. the design?"

Application

- frequent, no exceptional niche

"Does it add significant to the bottomline?"

Application
Functional

- specific

names, ages, amounts, durations, titles, ...