Multi-view Architecting and Modeling

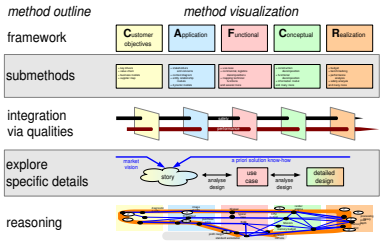
material by *Gerrit Muller*

presented by

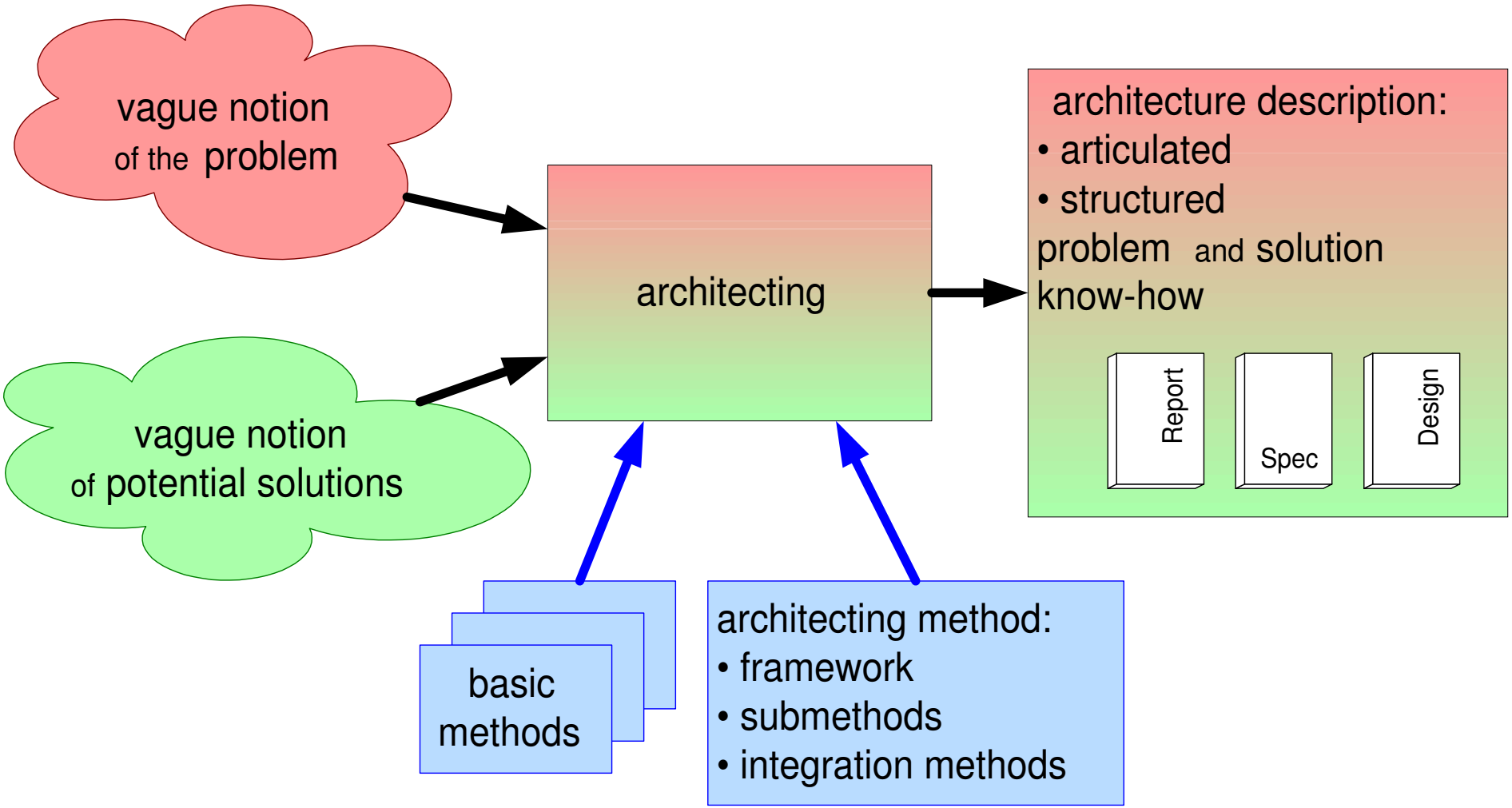
Abstract

Multi-view architecting connects the system design to customer context and life cycle context. We teach an architecting method based on many views and fast iteration of the views. Visual models, functional models, and mathematical models in all views are the means to communicate about the system, to discuss specification and design choices, to reason about consequences, and to make decisions.

December 21, 2011
status: planned
version: 0.1



From vague notions to articulate and structured

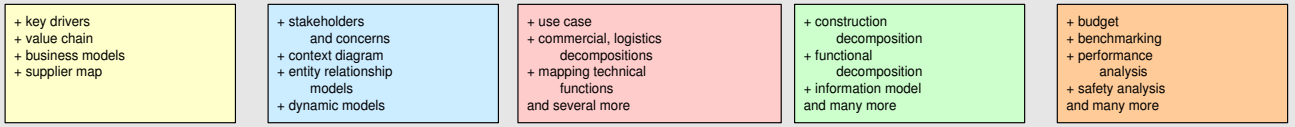


Overview of architecting method

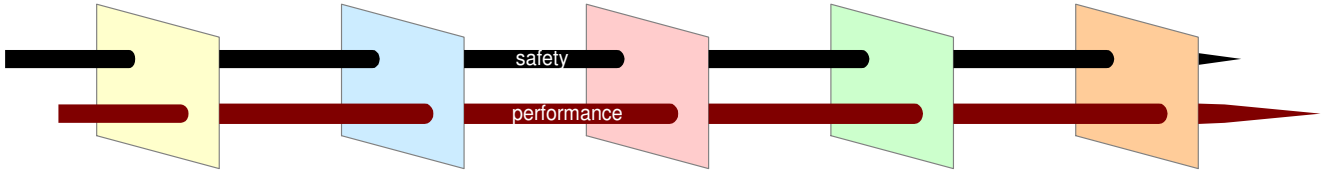
method outline *method visualization*

framework

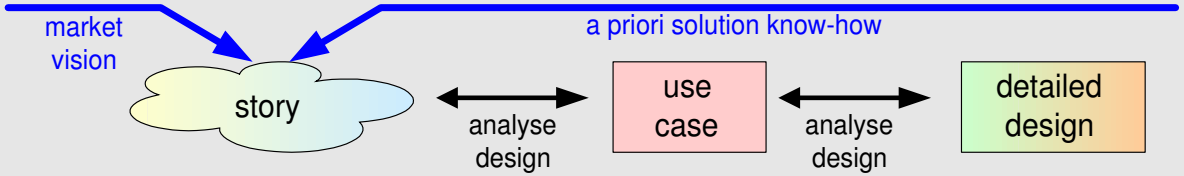
submethods



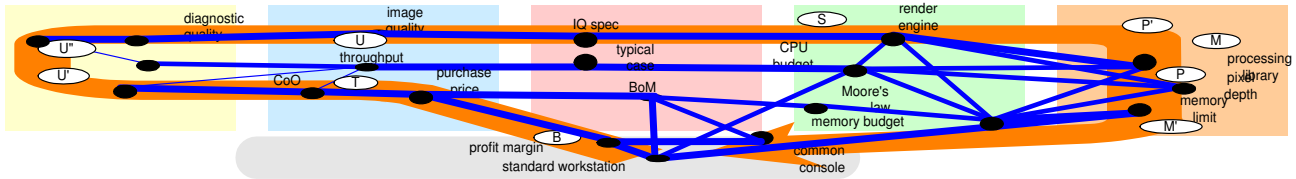
integration via qualities



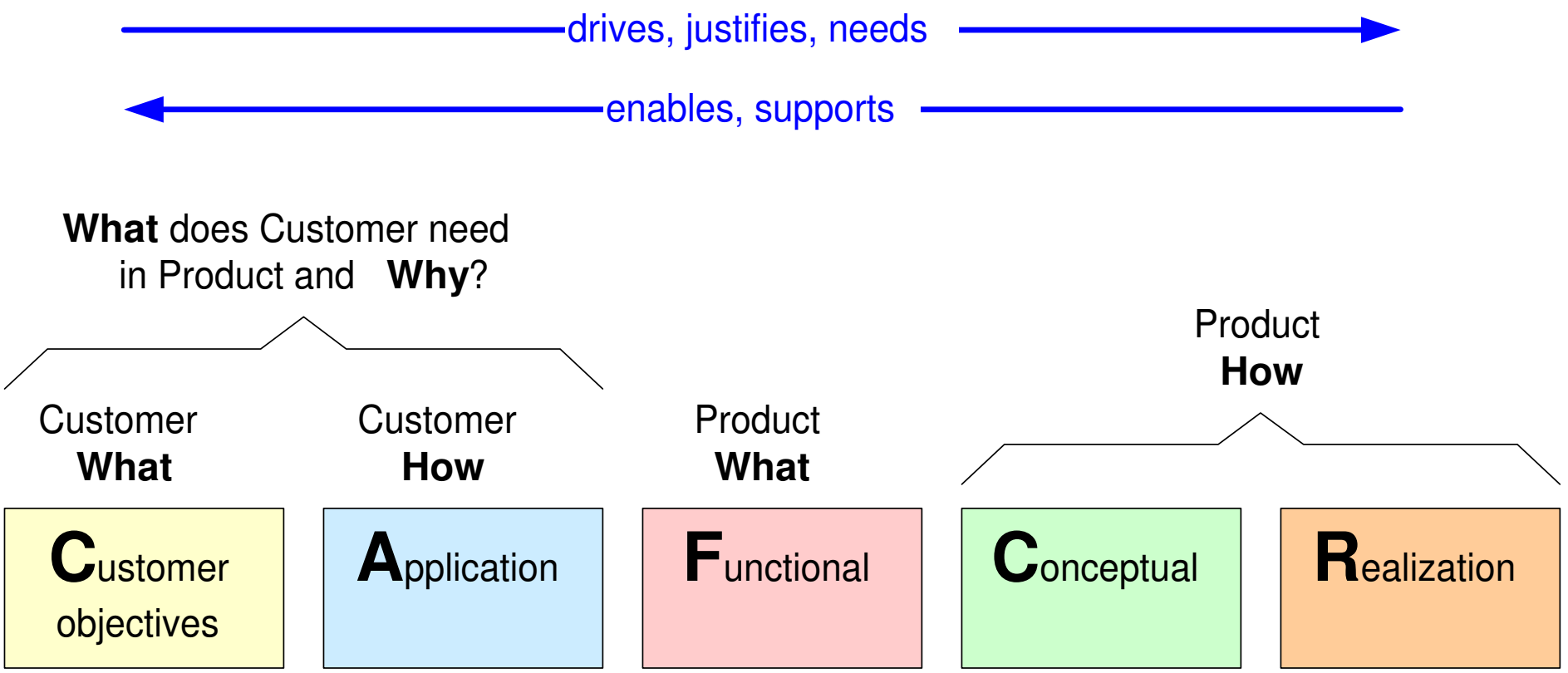
explore specific details



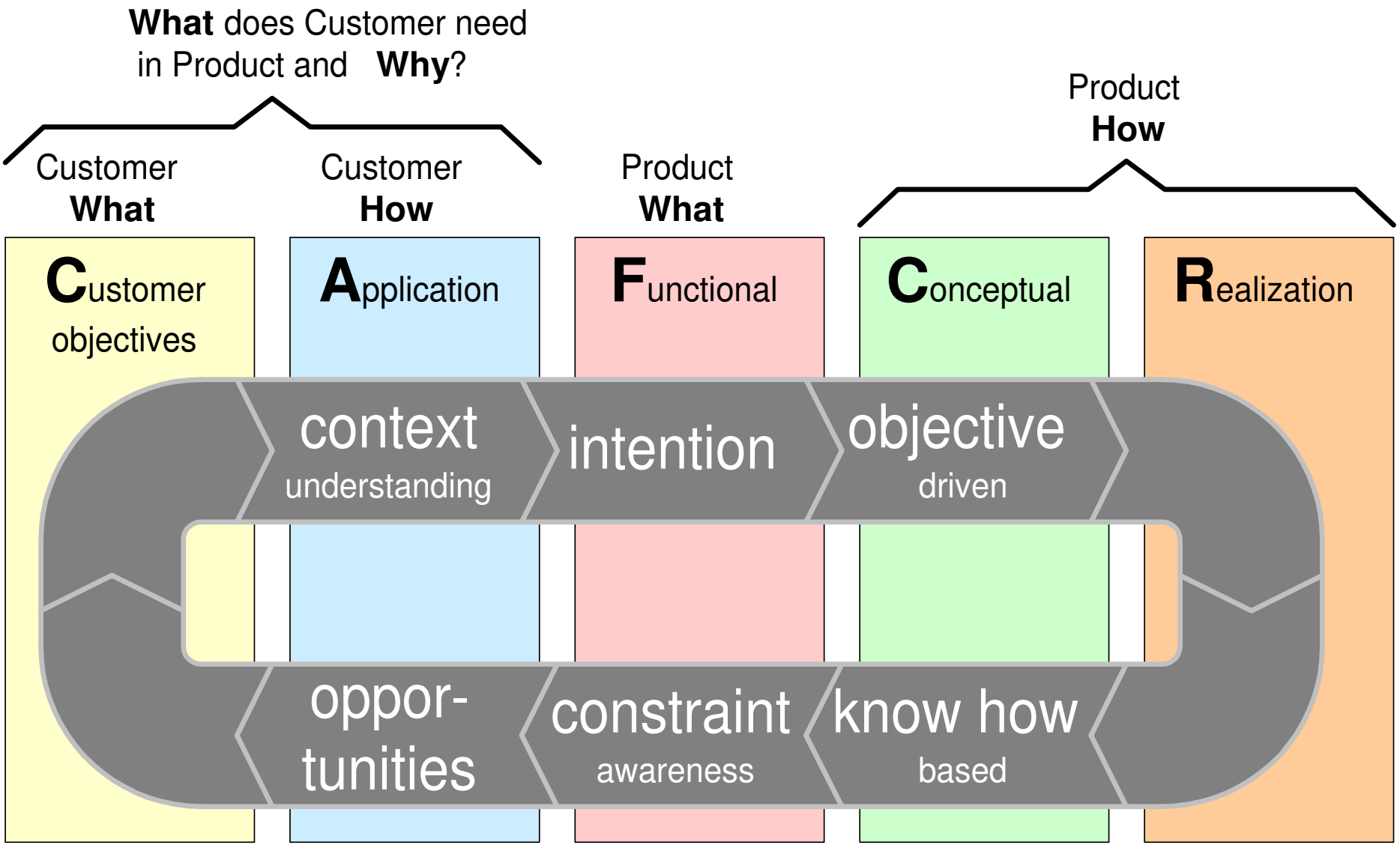
reasoning



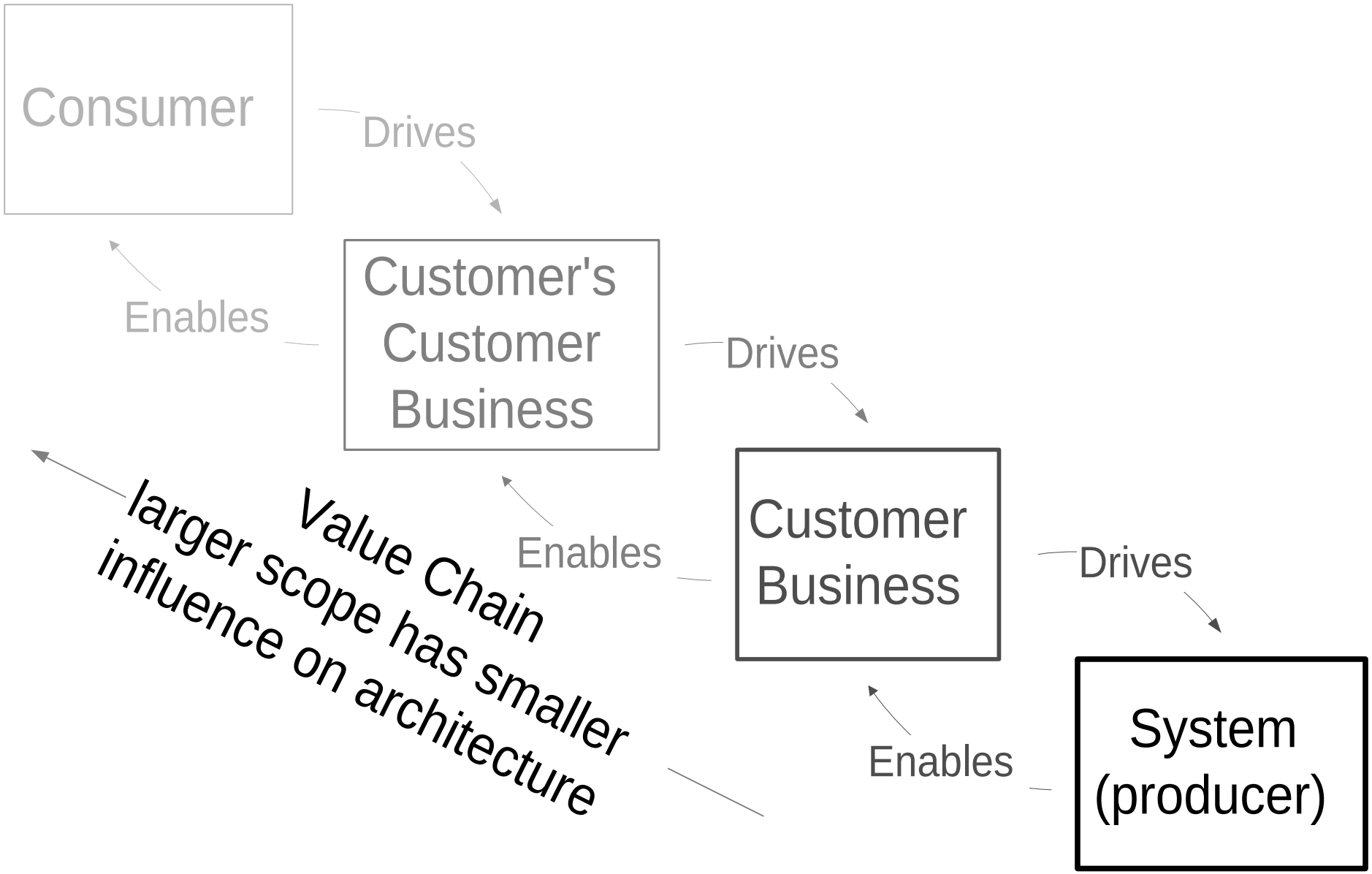
The “CAFCR” model



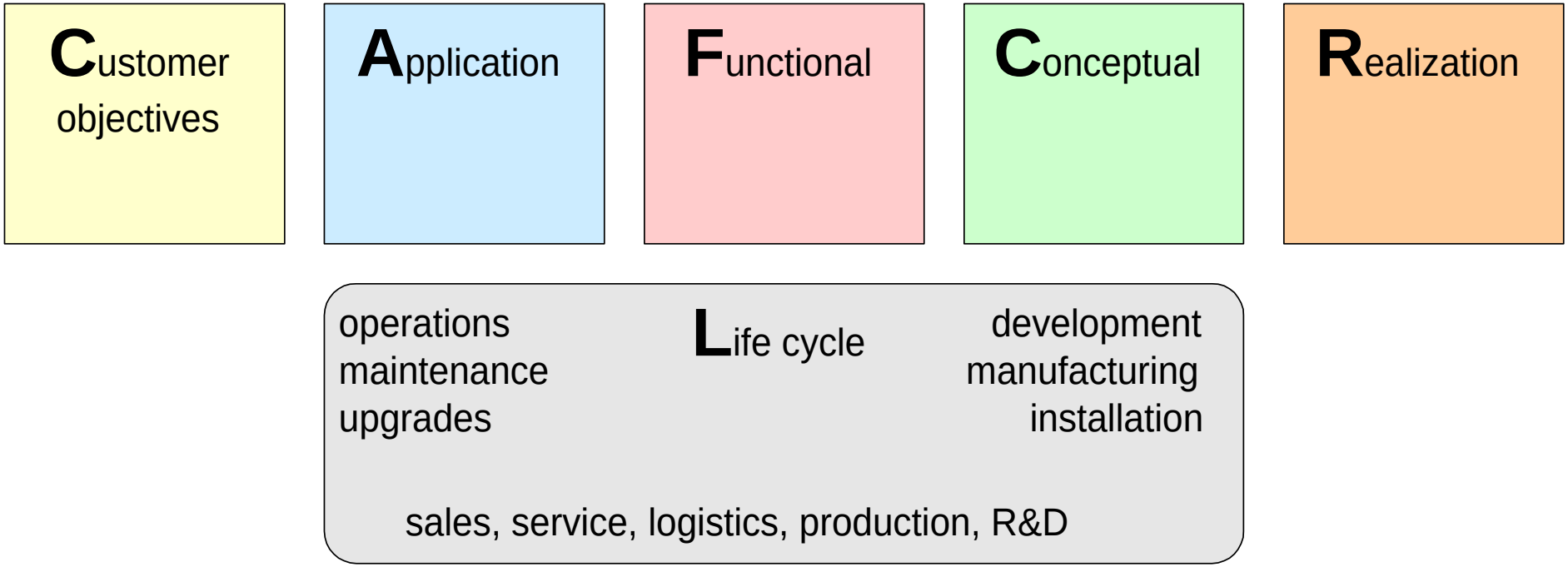
Five viewpoints for an architecture



CAFCR can be applied recursively



CAFCR+ model; Life Cycle View



Exercise Bottom-up Scan CAFCR

make a bottom-up analysis of your product:

1. realization

2. conceptual

3. functional

4. application

5. customer objectives

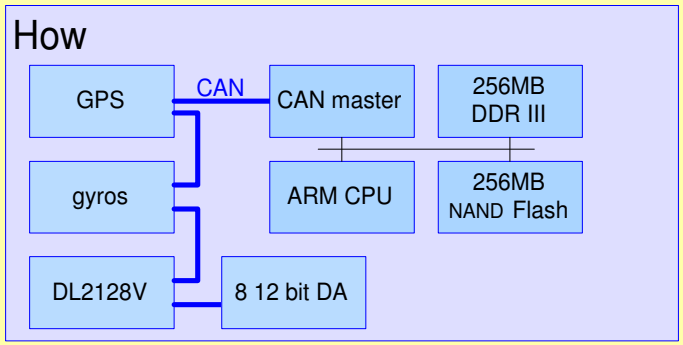
6. qualities

use time boxes of 15 minutes per view

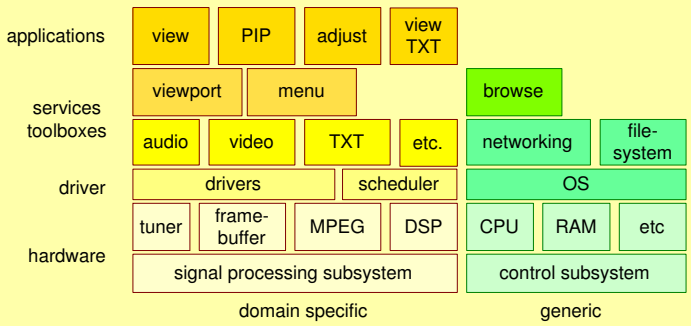
show the most dominant decomposition of that view, as diagram or as a list; some more guidance will be given per step.

Step 1: Realization View

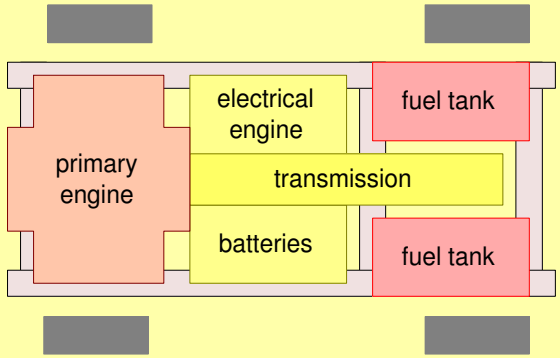
Choose 1 or 2 items from below



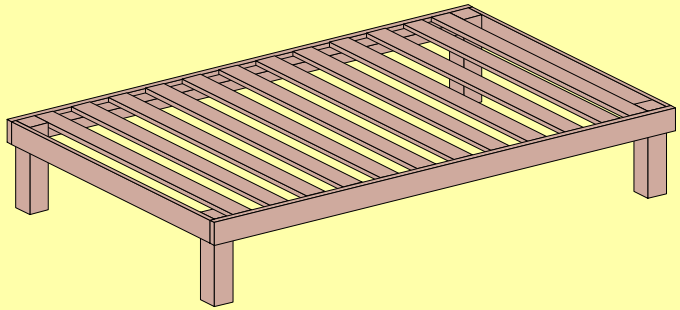
HW block diagram



SW layer diagram



2D layout of system internals

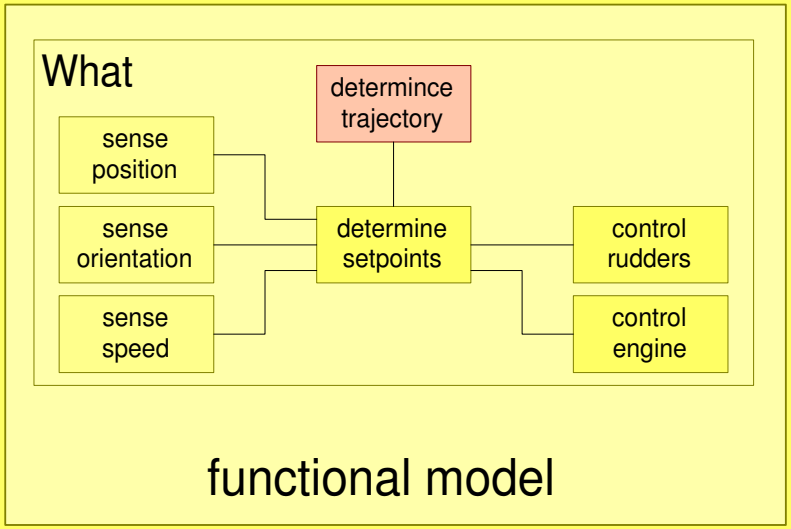


3D sketch of system internals

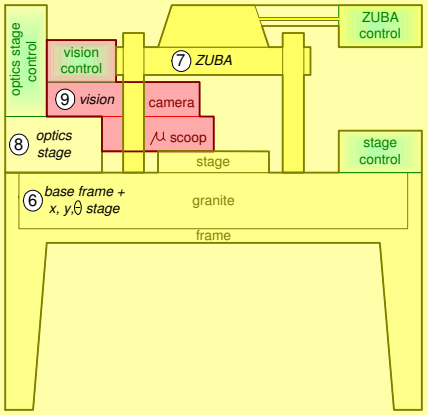
Annotate/mark most critical technologies or characteristics

Step 2: Conceptual View

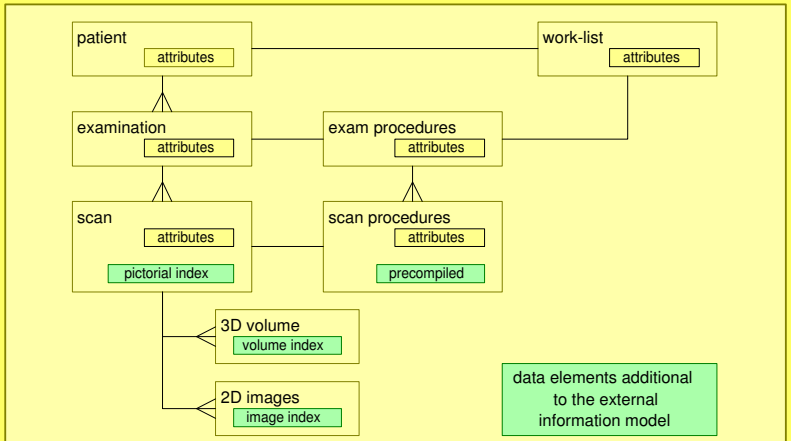
Chose 1 or 2 items from below



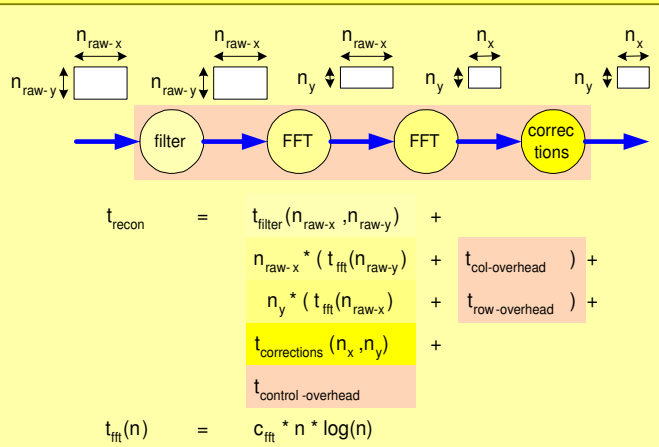
functional model



subsystem decomposition

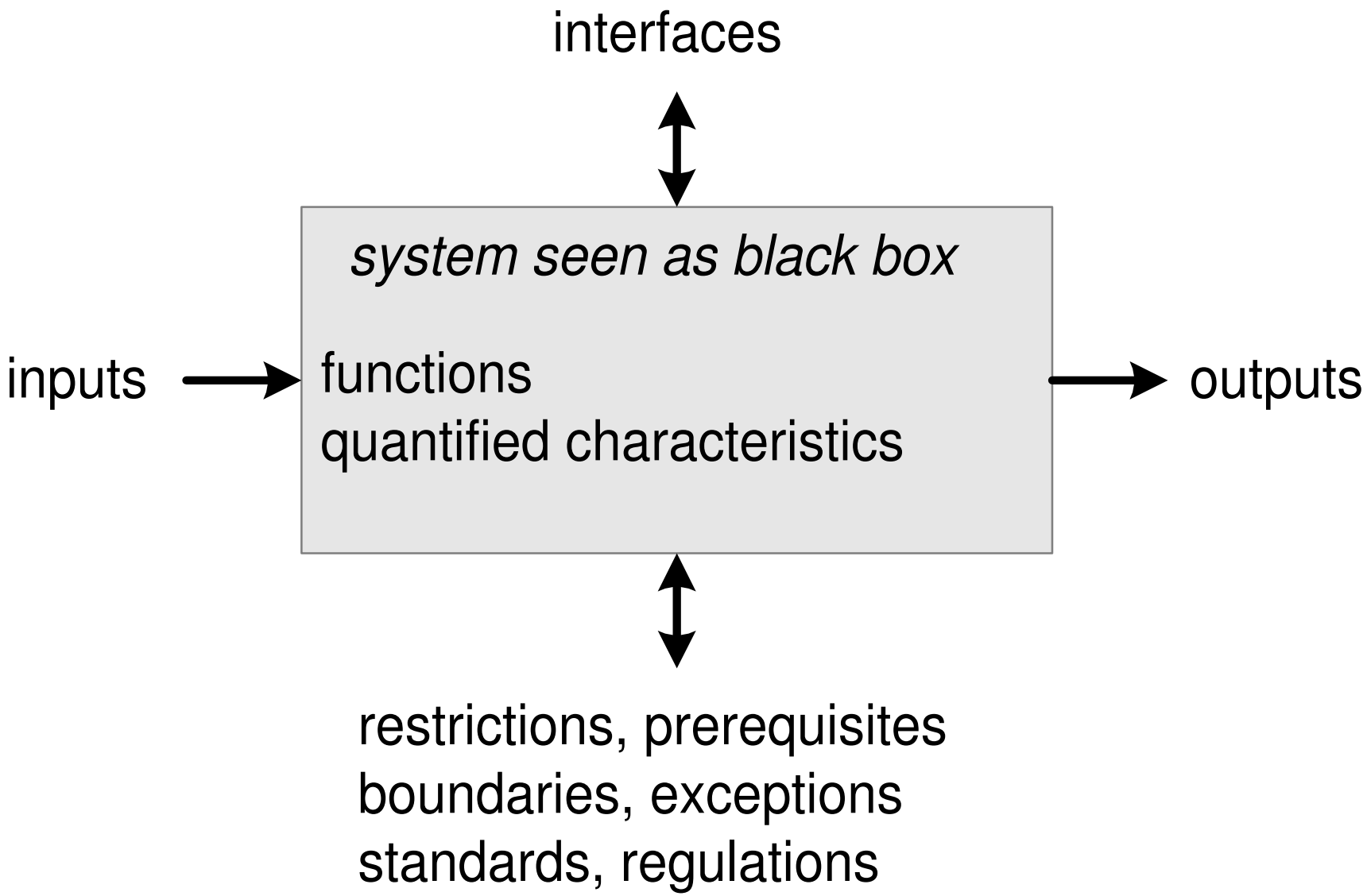


information model



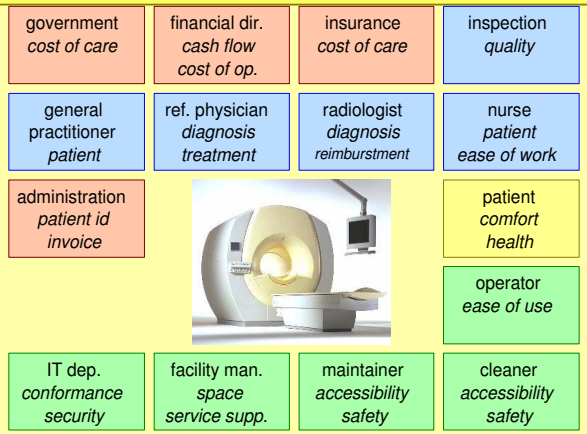
performance model

Step 3: Functional View; Top level Spec

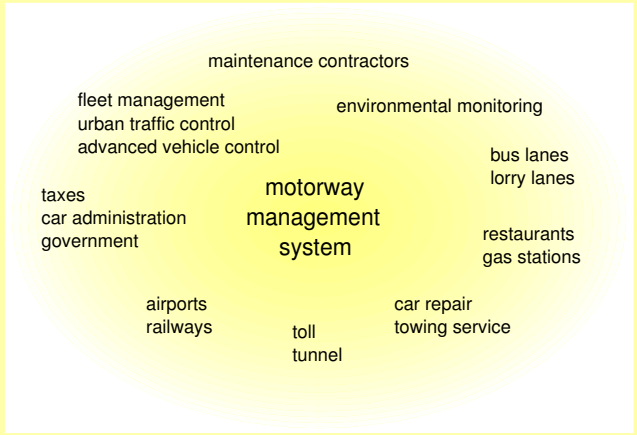


Step 4: Application View

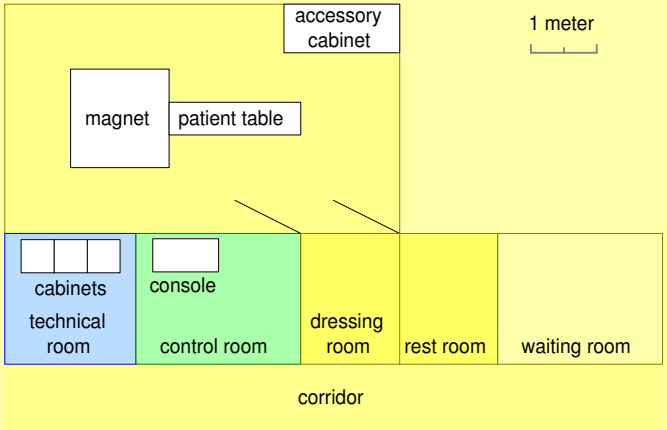
Chose 1 or 2 items from below



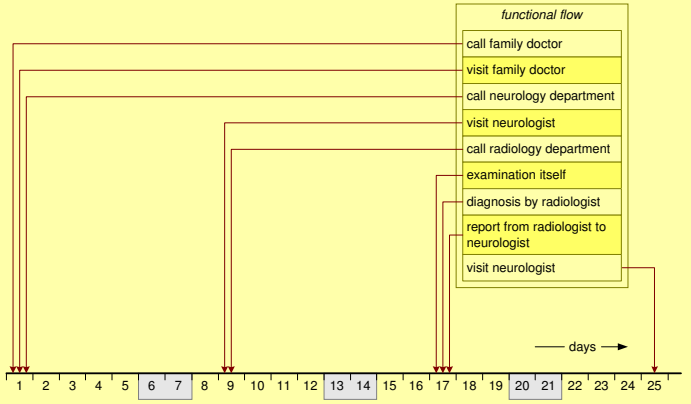
stakeholders and concerns (who)



system context

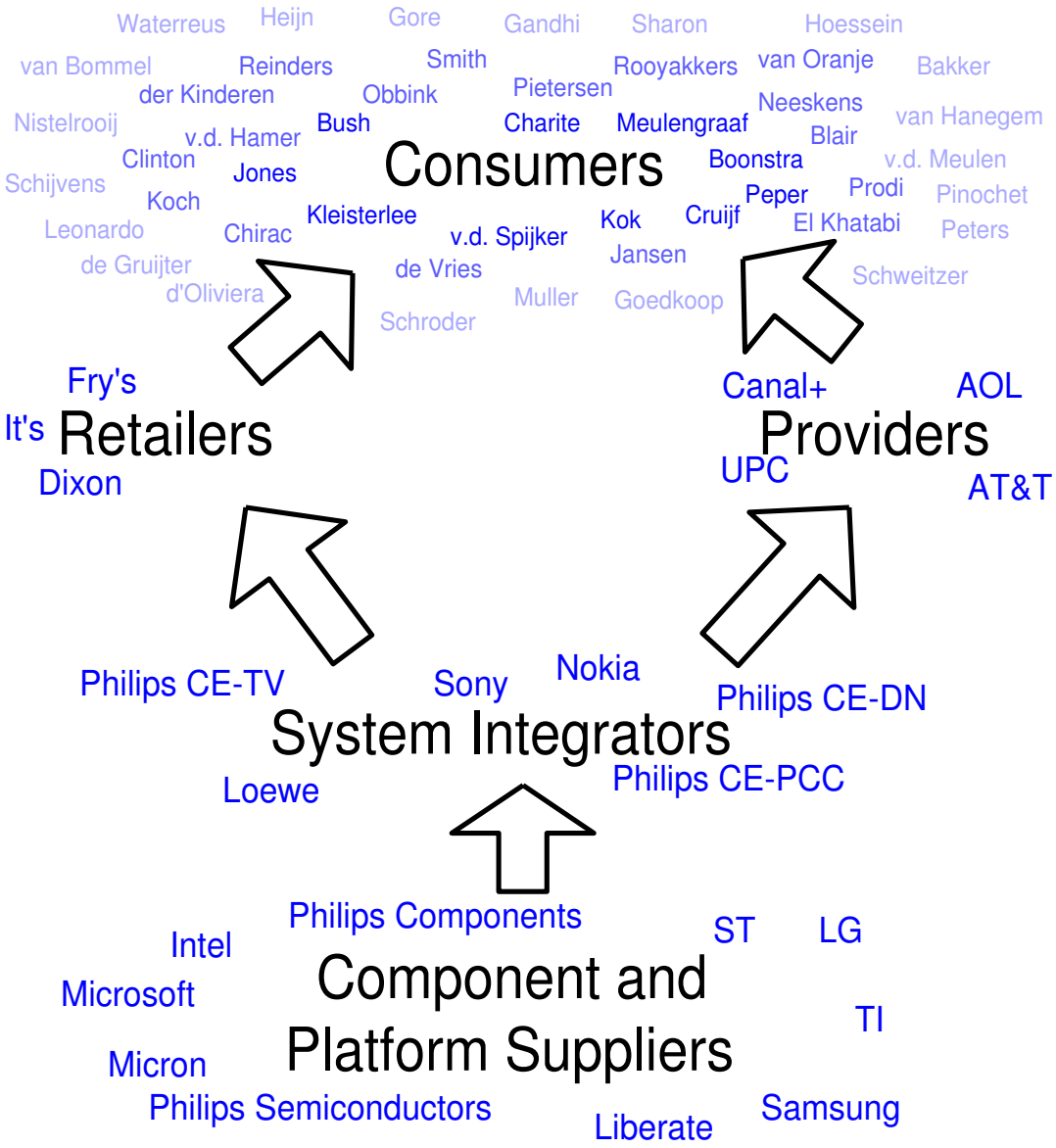


2D map (where)



work flow & time line (what, when)

Step 5: Customer Objectives View; Value Chain



Step 6: Qualities

usable
- usability
- attractiveness
- responsiveness
- image quality
- wearability
- storability
- transportability

dependable
- safety
- security
- reliability
- robustness
- integrity
- availability

effective
- throughput or productivity

interoperable
- connectivity
- 3rd party extendible

liable
- liability
- testability
- traceability
- standards compliance

efficient
- resource utilization
- cost of ownership

consistent
- reproducibility
- predictability

serviceable
- serviceability
- configurability
- installability

future proof
- evolvability
- portability
- upgradeability
- extendibility
- maintainability

logistics friendly
- manufacturability
- logistics flexibility
- lead time

ecological
- ecological footprint
- contamination
- noise
- disposability

down to earth attributes
- cost price
- power consumption
- consumption rate (water, air, chemicals, et cetera)
- size, weight
- accuracy

Presentation

Present the results top-down

Use two to three flip charts of the six that have been created.

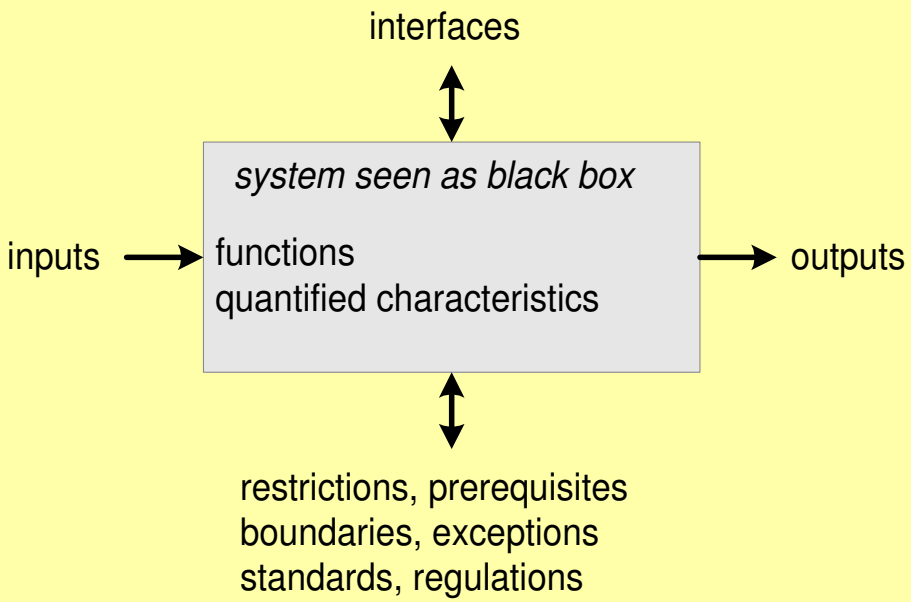
Explain in five minutes the needs of the customer, the system, and the major design choices.

Exercise Black Box

Make specification overview with ~10 Parameters (or functions or interfaces)

SMART Key Performance

determine at least one use case



use case
typical use with relevant context data (quantified!)

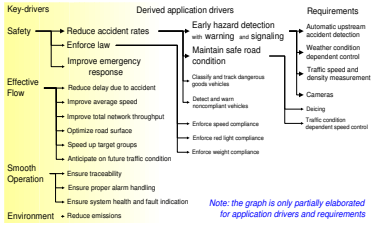
- **S**pecific quantified
- **M**easurable verifiable
- **A**chievable (Attainable, Action oriented, Acceptable, Agreed-upon, Accountable)
- **R**ealistic (Relevant, Result-Oriented)
- **T**ime-bounded (Timely , Tangible, Traceable)

Key Drivers How To

by *Gerrit Muller* Buskerud University College
e-mail: gerrit.muller@embeddedsystems.nl
www.gaudisite.nl

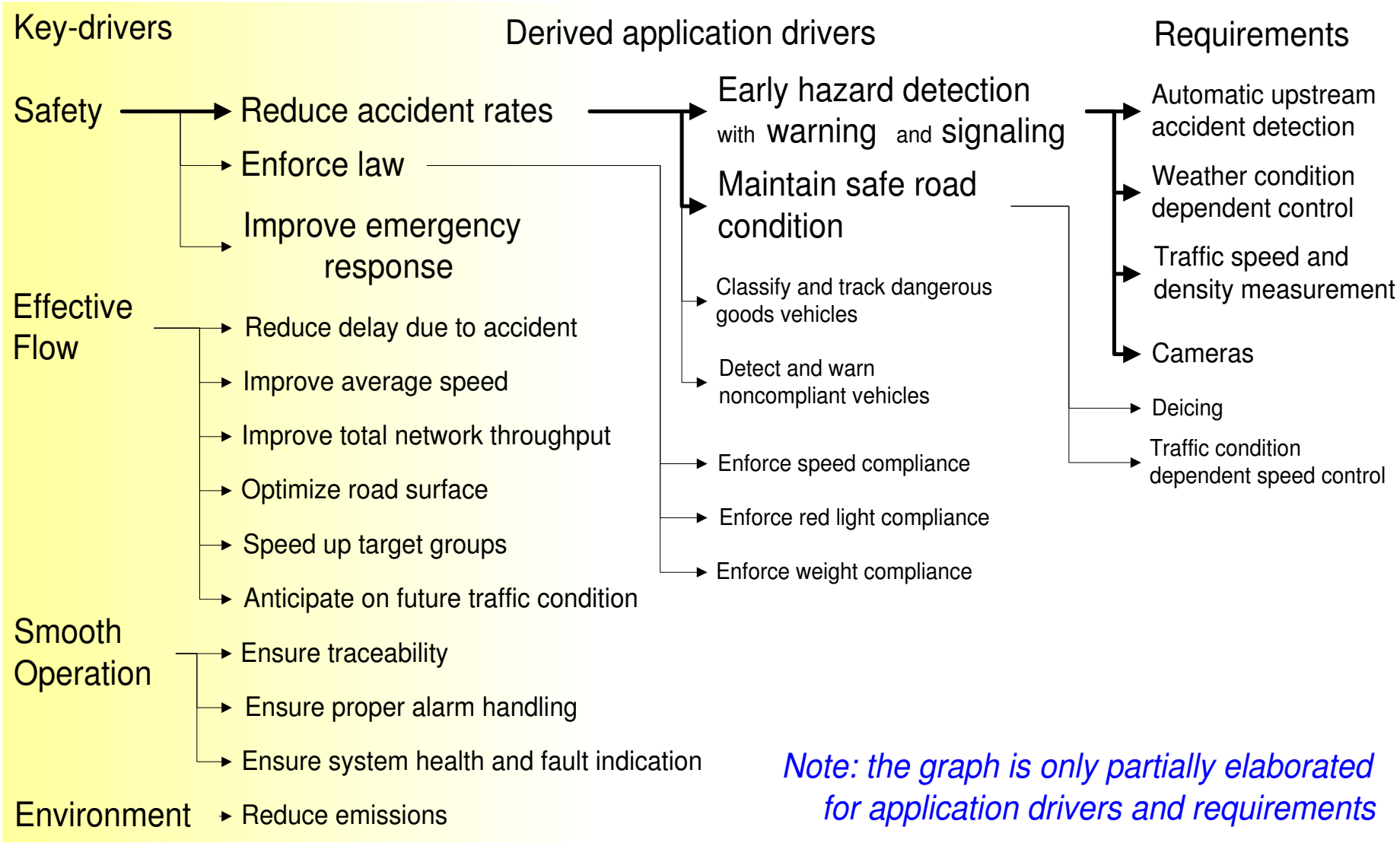
Abstract

The notion of ”business key drivers” is introduced and a method is described to link these key drivers to the product specification.



December 21, 2011
status: draft
version: 0.2

Example Motorway Management Analysis



Note: the graph is only partially elaborated for application drivers and requirements

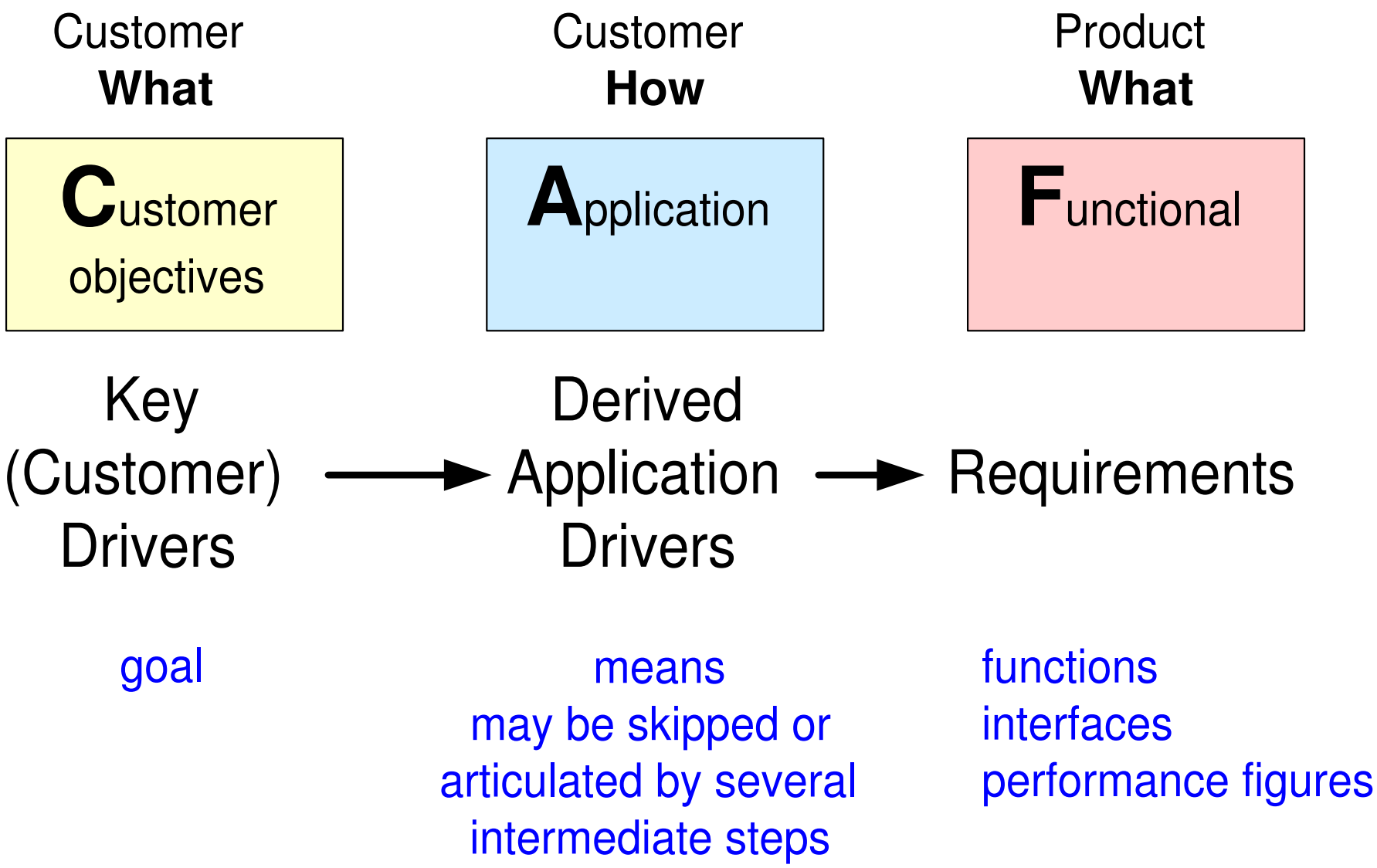
Method to create Key Driver Graph

• Define the scope specific.	in terms of stakeholder or market segments
• Acquire and analyze facts	extract facts from the product specification and ask why questions about the specification of existing products .
• Build a graph of relations between drivers and requirements by means of brainstorming and discussions	where requirements may have multiple drivers
• Obtain feedback	discuss with customers , observe their reactions
• Iterate many times	increased understanding often triggers the move of issues from driver to requirement or vice versa and rephrasing

Recommendation for the Definition of Key Drivers

• Limit the number of key-drivers	minimal 3, maximal 6
• Don't leave out the obvious key-drivers	for instance the well-known main function of the product
• Use short names, recognized by the customer.	
• Use market-/customer- specific names, no generic names	for instance replace " ease of use " by "minimal number of actions for experienced users ", or "efficiency " by " integral cost per patient "
• Do not worry about the exact boundary between Customer Objective and Application	create clear goal means relations

Transformation of Key Drivers into Requirements

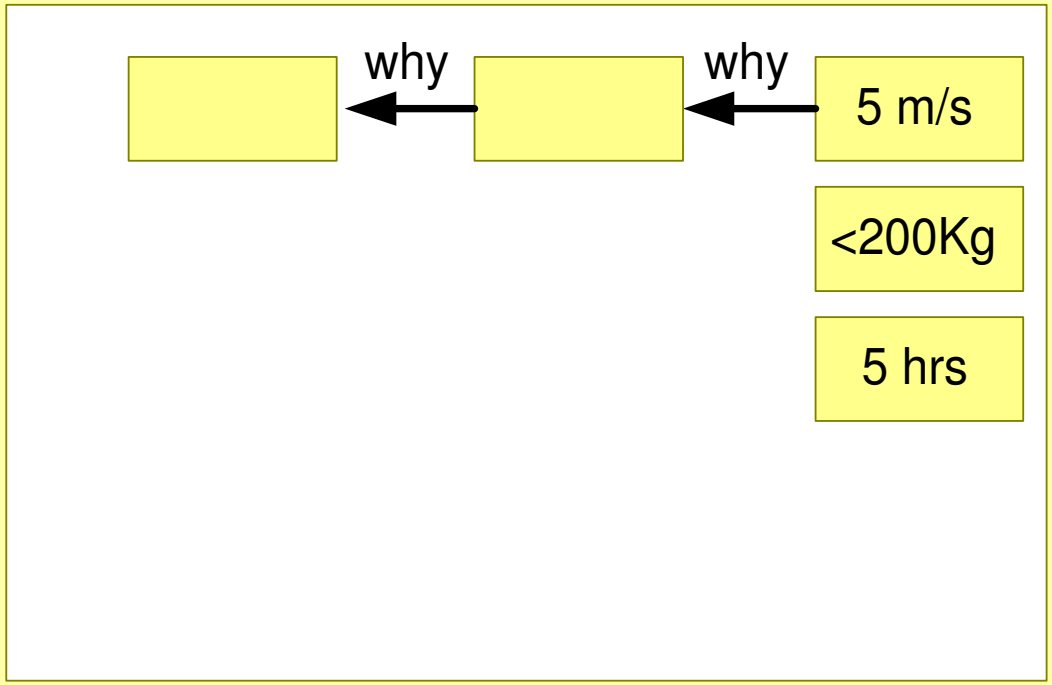


Exercise Customer Side

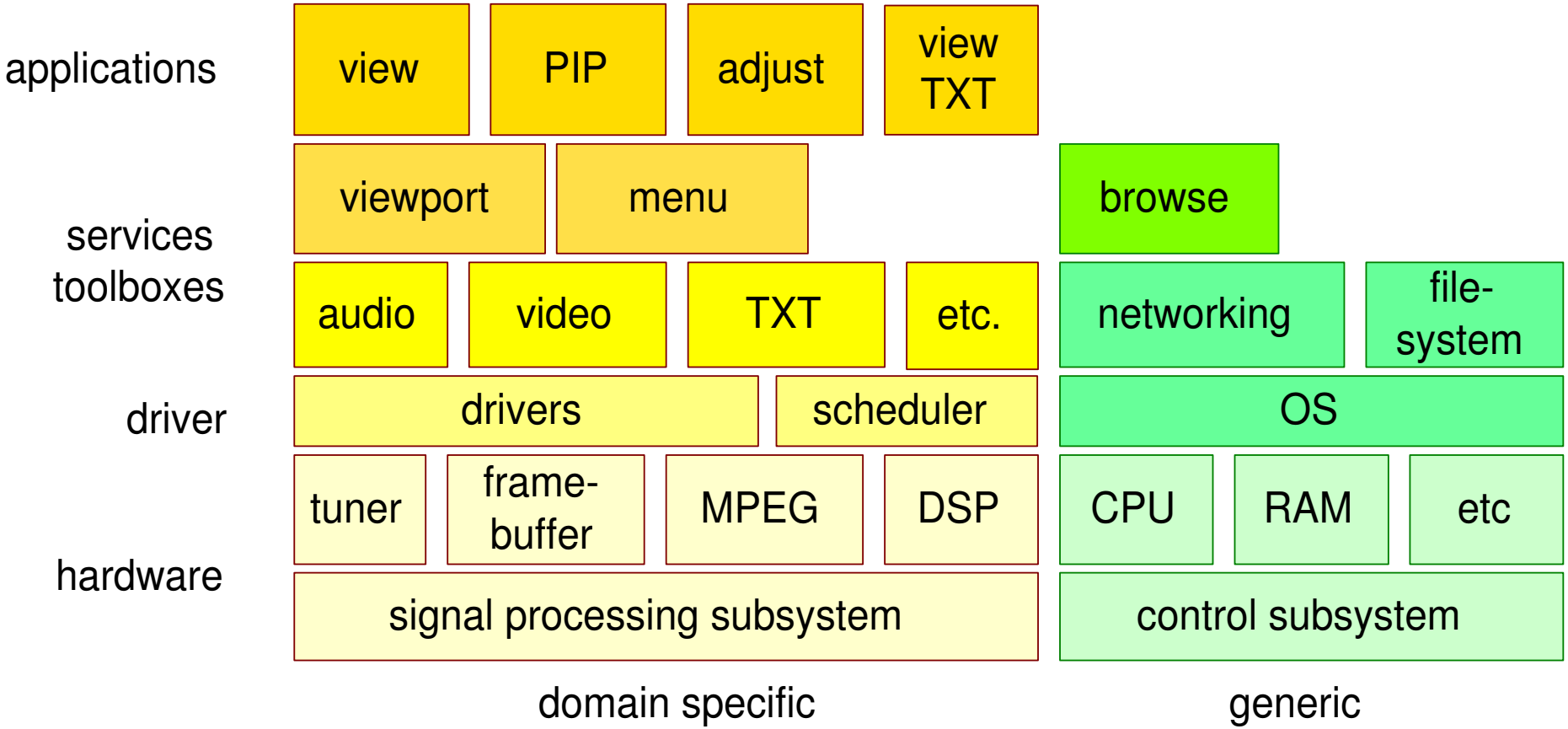
Make a customer key driver graph

Use yellow note stickers

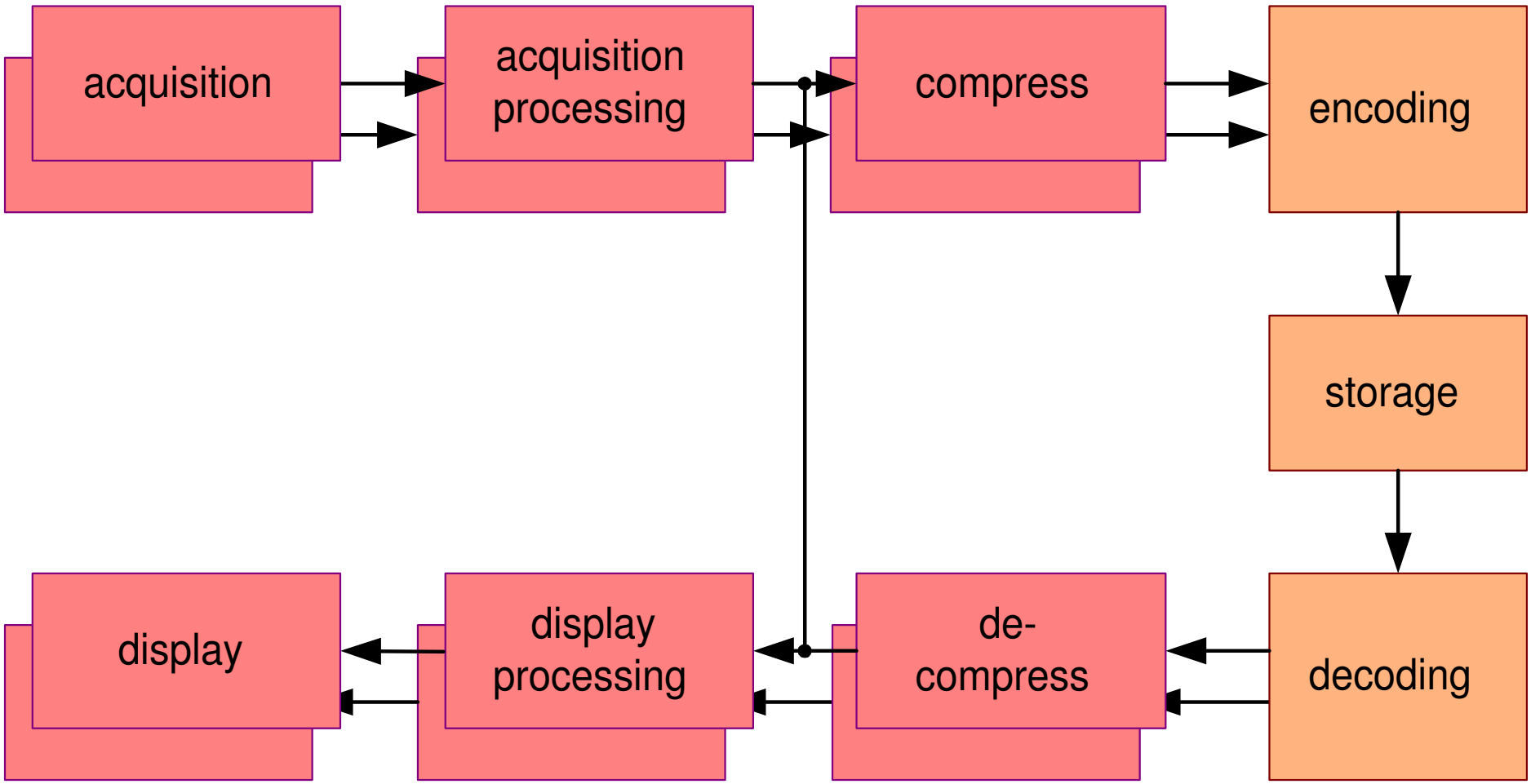
Start at the right hand side



Example construction decomposition simple TV

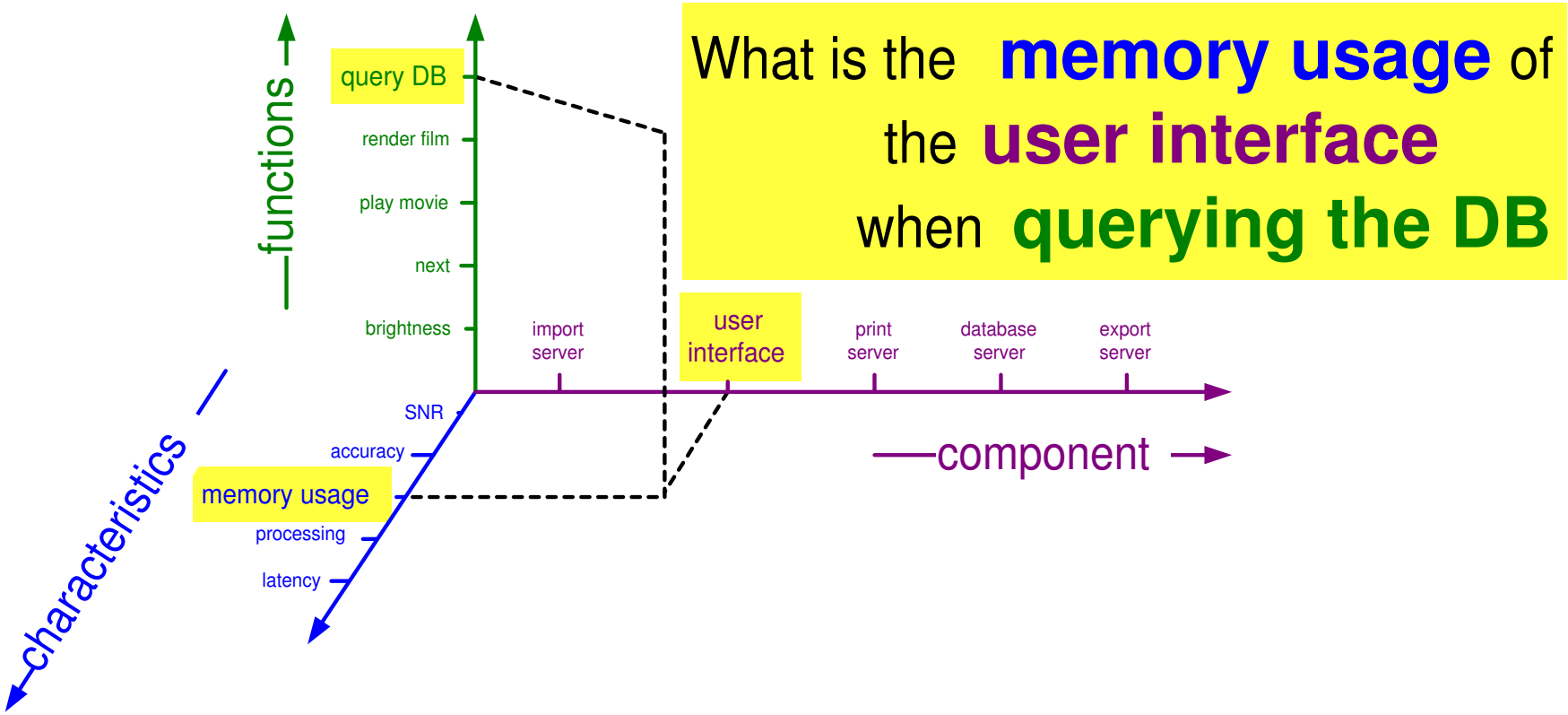


Example functional decomposition camera type device



Question generator for multiple decompositions

How about the **<characteristic>**
of the **<component>**
when performing **<function>**?



What is the **memory usage** of
the **user interface**
when **querying the DB**

Selection factors to improve the question generator

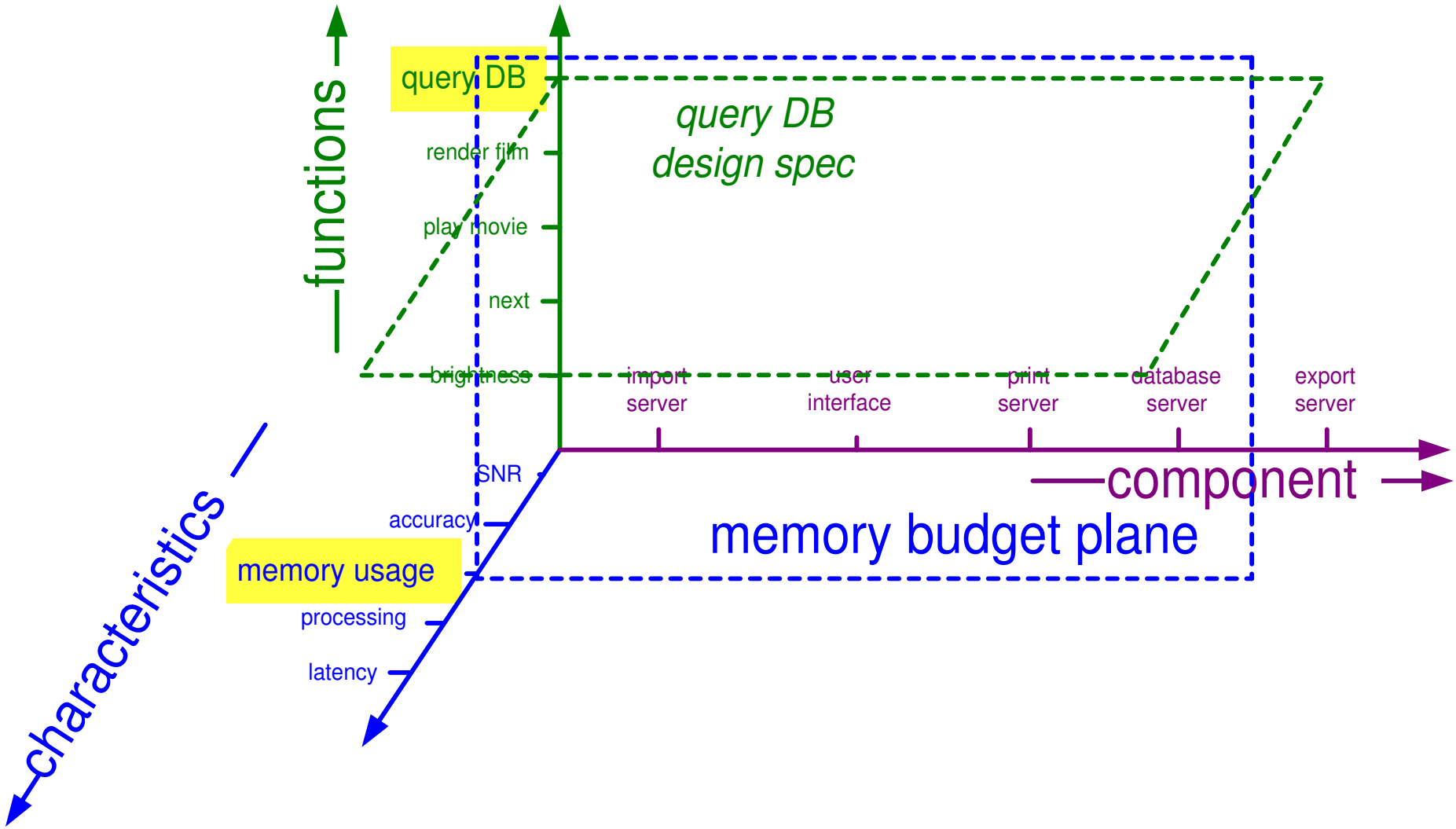
Critical for system performance

Risk planning wise

Least robust part of the design

Suspect part of the design
- experience based
- person based

Addressing planes or lines



Exercise Design Side

Make/improve functional model and construction decomposition

Make a design for one of the critical design aspects

Security as example through all views

Customer objectives	**A**pplication	**F**unctional	**C**onceptual	**R**ealization
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
	desired characteristics, specifications & mechanisms			
 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
			threats	

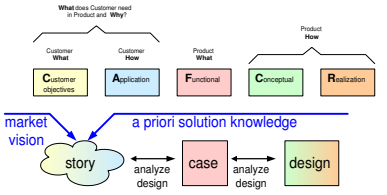
Story How To

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

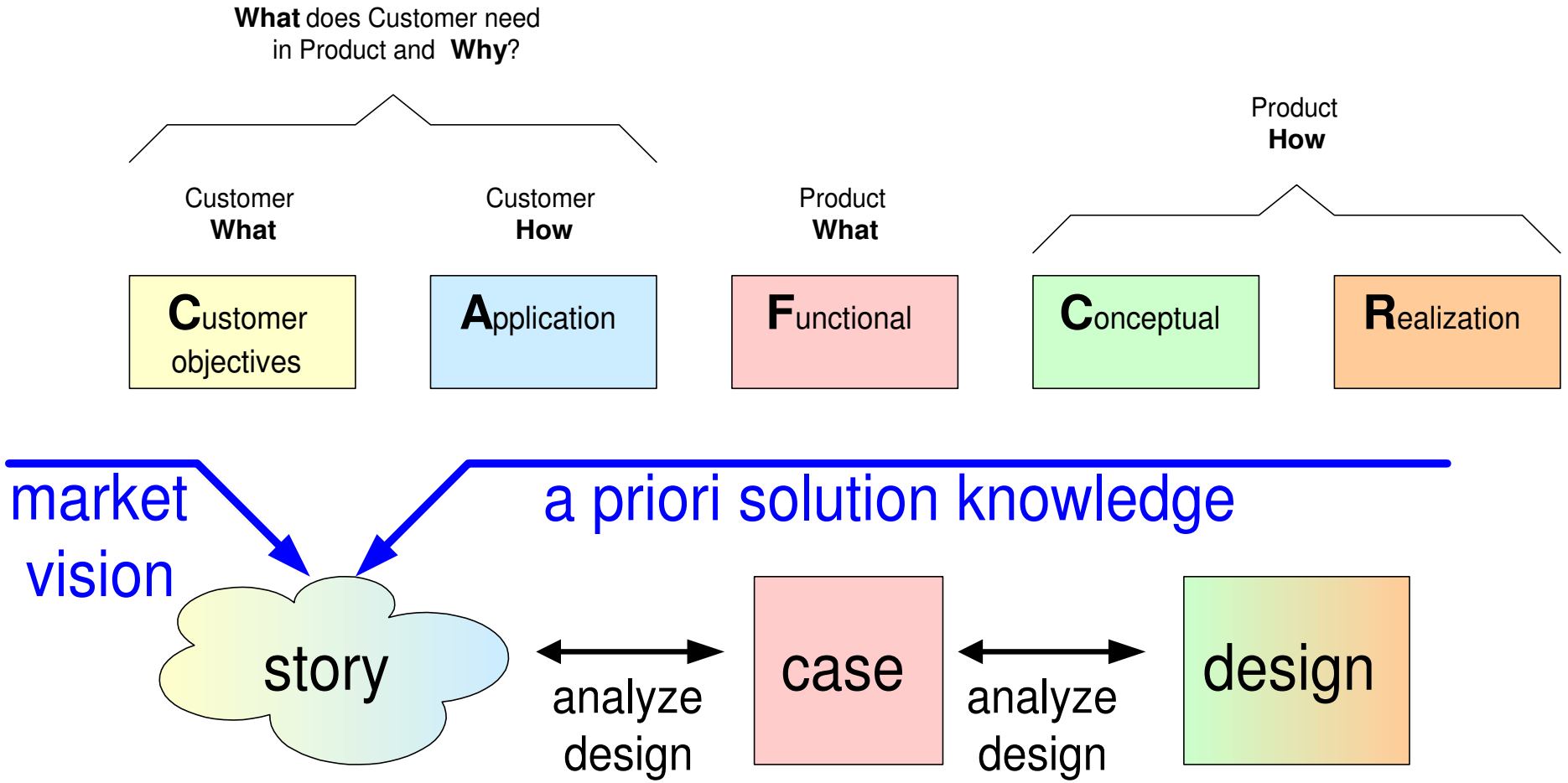
Abstract

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.

December 21, 2011
status: concept
version: 1.1

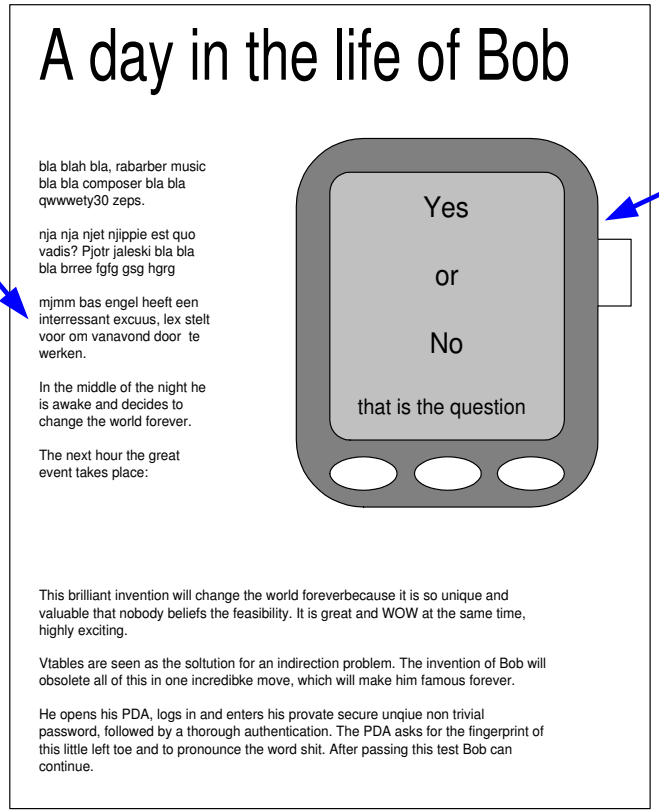


From story to design



Example story layout

ca. half a page of plain English text



draft or sketch of some essential appliance

Points of attention

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

Criteria 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 customers queuing up 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 significantly to the bottom line?"

Application
Functional

- specific

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

Example of a story

Betty is a 70-year-old woman who lives in Eindhoven. Three years ago her husband passed away and since then she lives in a home for the elderly. Her 2 children, Angela and Robert, come and visit her every weekend, often with Betty's grandchildren Ashley and Christopher. As so many women of her age, Betty is reluctant to touch anything that has a technical appearance. She knows how to operate her television, but a VCR or even a DVD player is way to complex.

When Betty turned 60, she stopped working in a sewing studio. Her work in this noisy environment made her hard-of-hearing with a hearing-loss of 70dB around 2kHz. The rest of the frequency spectrum shows a loss of about 45dB. This is why she had problems understanding her grandchildren and why her children urged her to apply for hearing aids two years ago. Her technophobia (and her first hints or arthritis) inhibit her to change her hearing aids' batteries. Fortunately her children can do this every weekend.

This Wednesday Betty visits the weekly Bingo afternoon in the meetingplace of the old-folk's home. It's summer now and the tables are outside. With all those people there it's a lot of chatter and babble. Two years ago Betty would never go to the bingo: "I cannot hear a thing when everyone babbles and clatters with the coffee cups. How can I hear the winning numbers?!". Now that she has her new digital hearing instruments, even in the bingo cacophony, she can understand everyone she looks at. Her social life has improved a lot and she even won the bingo a few times.

That same night, together with her friend Janet, she attends Mozart's opera The Magic Flute. Two years earlier this would have been one big low rumbly mess, but now she even hears the sparkling high piccolos. Her other friend Carol never joins their visits to the theaters. Carol also has hearing aids, however hers only "work well" in normal conversations. "When I hear music it's as if a butcher's knife cuts through my head. It's way too sharp!". So Carol prefers to take her hearing aids out, missing most of the fun. Betty is so happy that her hearing instruments simply know where they are and adapt to their environment.

source: Roland Mathijssen
Embedded Systems Institute
Eindhoven

Value and Challenges in this story

Customer objectives

Application

Value proposition in this story:

quality of life:

active participation in different social settings

usability for nontechnical elderly people:

"intelligent" system is simple to use

loading of batteries

Conceptual

Realization

Challenges in this story:

Intelligent hearing instrument

Battery life —at least 1 week

No buttons or other fancy user interface on the hearing instrument, other than a robust On/Off method

The user does not want a technical device but a solution for a problem

Instrument can be adapted to the hearing loss of the user

Directional sensitivity (to prevent the so-called cocktail party effect)

Recognition of sound environments and automatic adaptation (adaptive filtering)

source: Roland Mathijssen, Embedded Systems Institute, Eindhoven

Threads of Reasoning

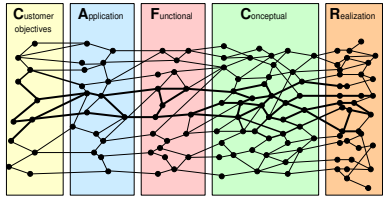
by *Gerrit Muller* Embedded Systems Institute
e-mail: gerrit.muller@embeddedsystems.nl
www.gaudisite.nl

Abstract

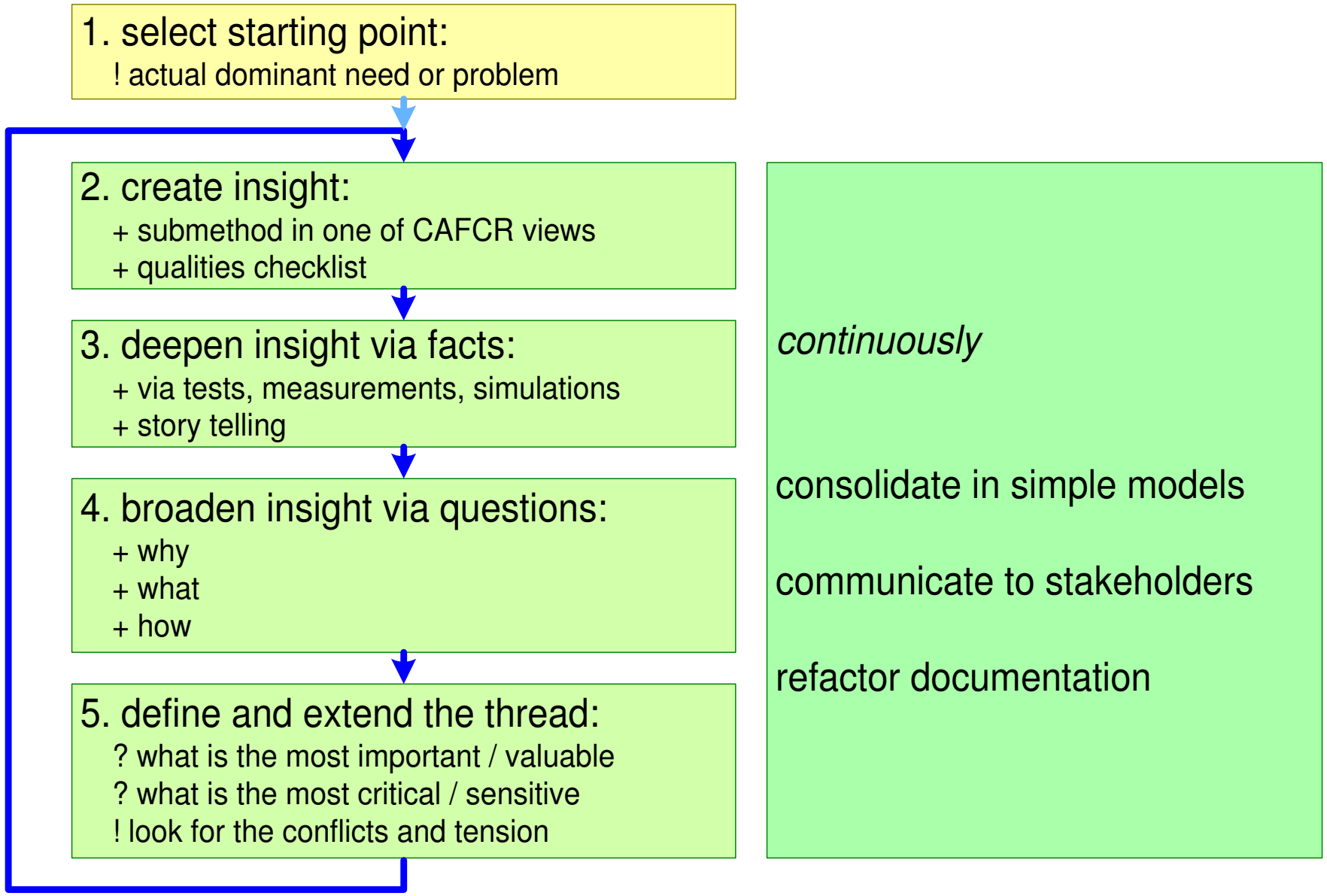
A method of reasoning is described, which addresses cross-cutting issues. The basis is fast iteration in the problem and solution space.

A thread of reasoning is a set of highly relevant related issues, which are addressed by articulating the problem in terms of tension and analyzing it in the CAFCR framework.

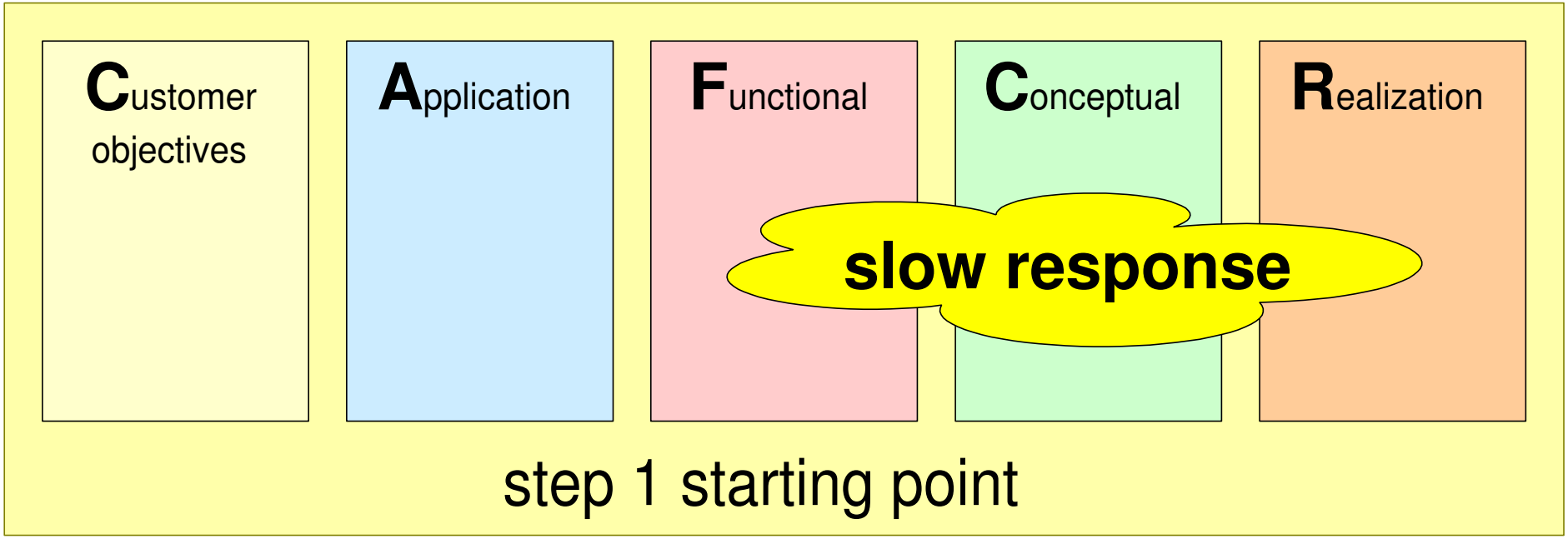
December 21, 2011
status: finished
version: 2.4



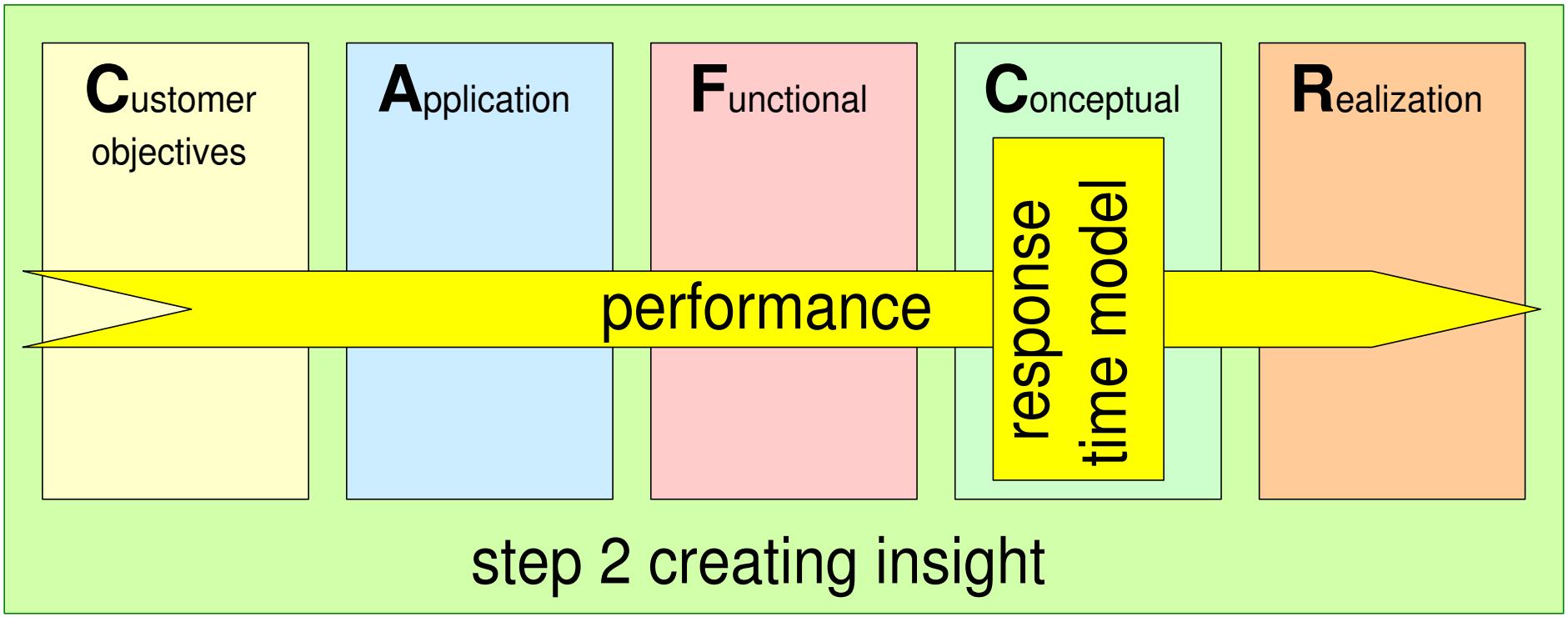
Overview of the reasoning approach



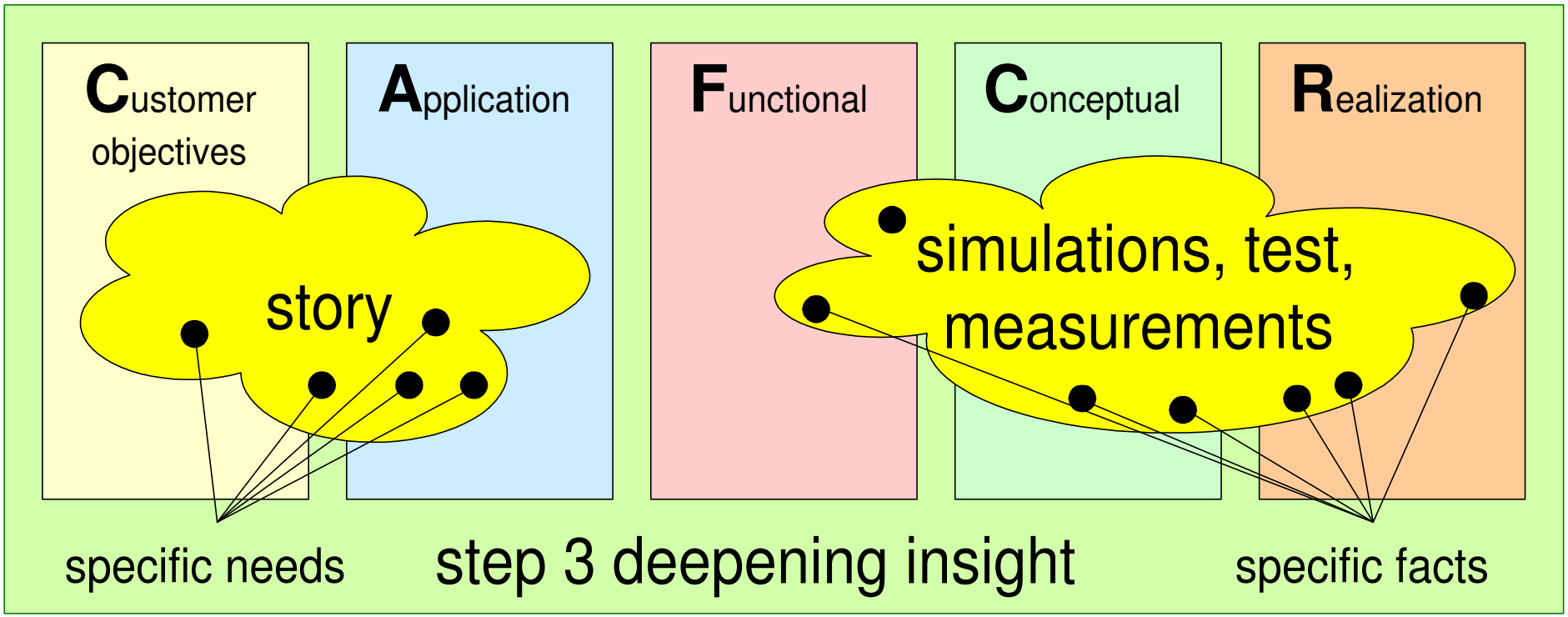
From starting point to insight



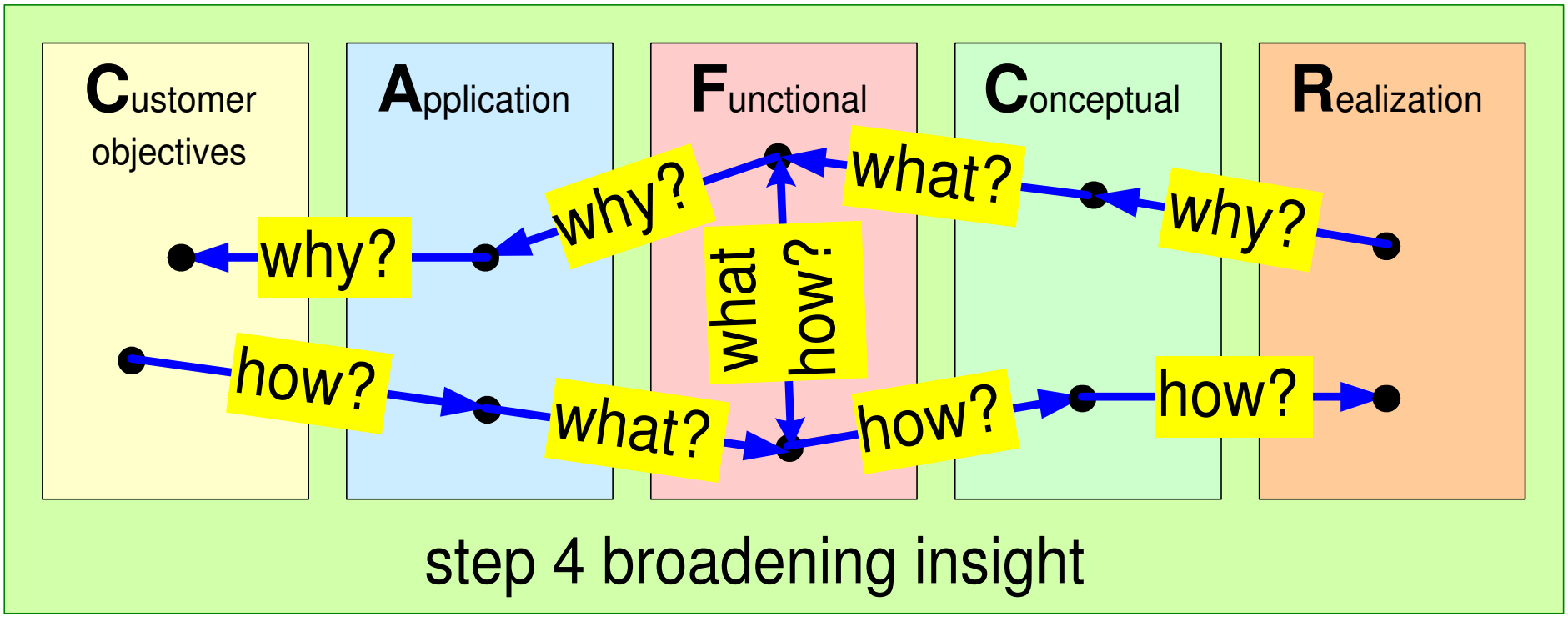
Creating Insight



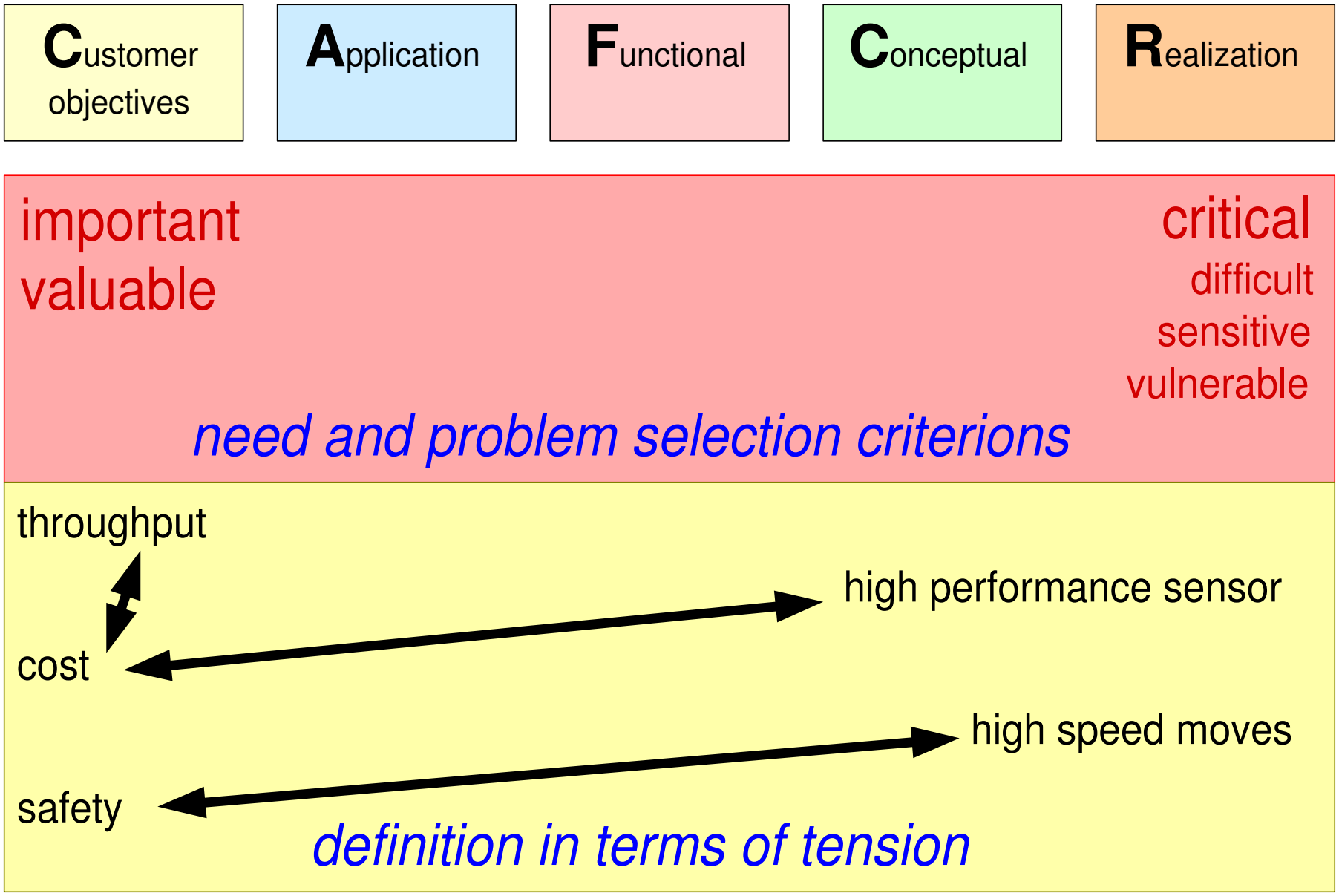
Deepening Insight



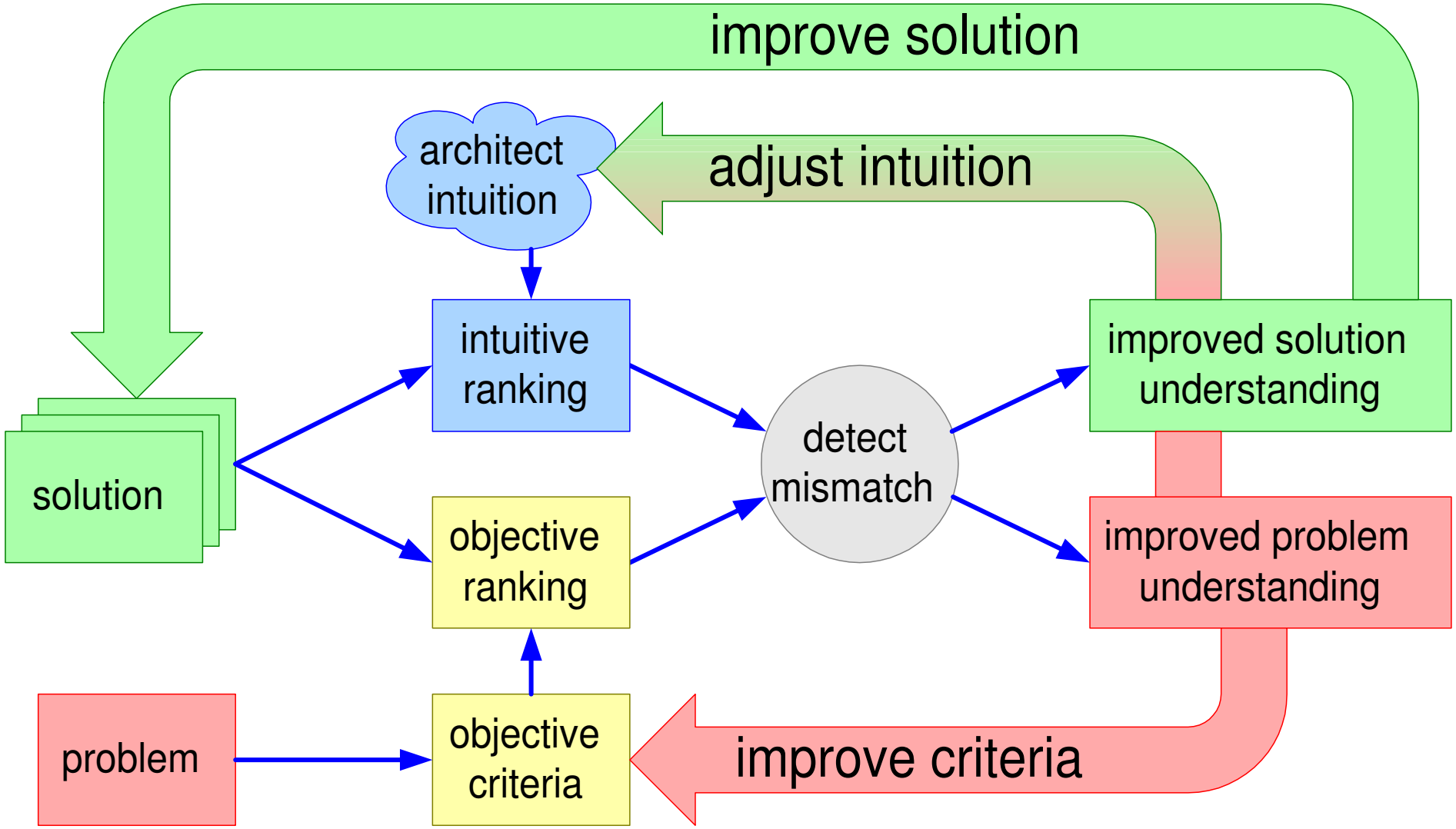
Broadening Insight



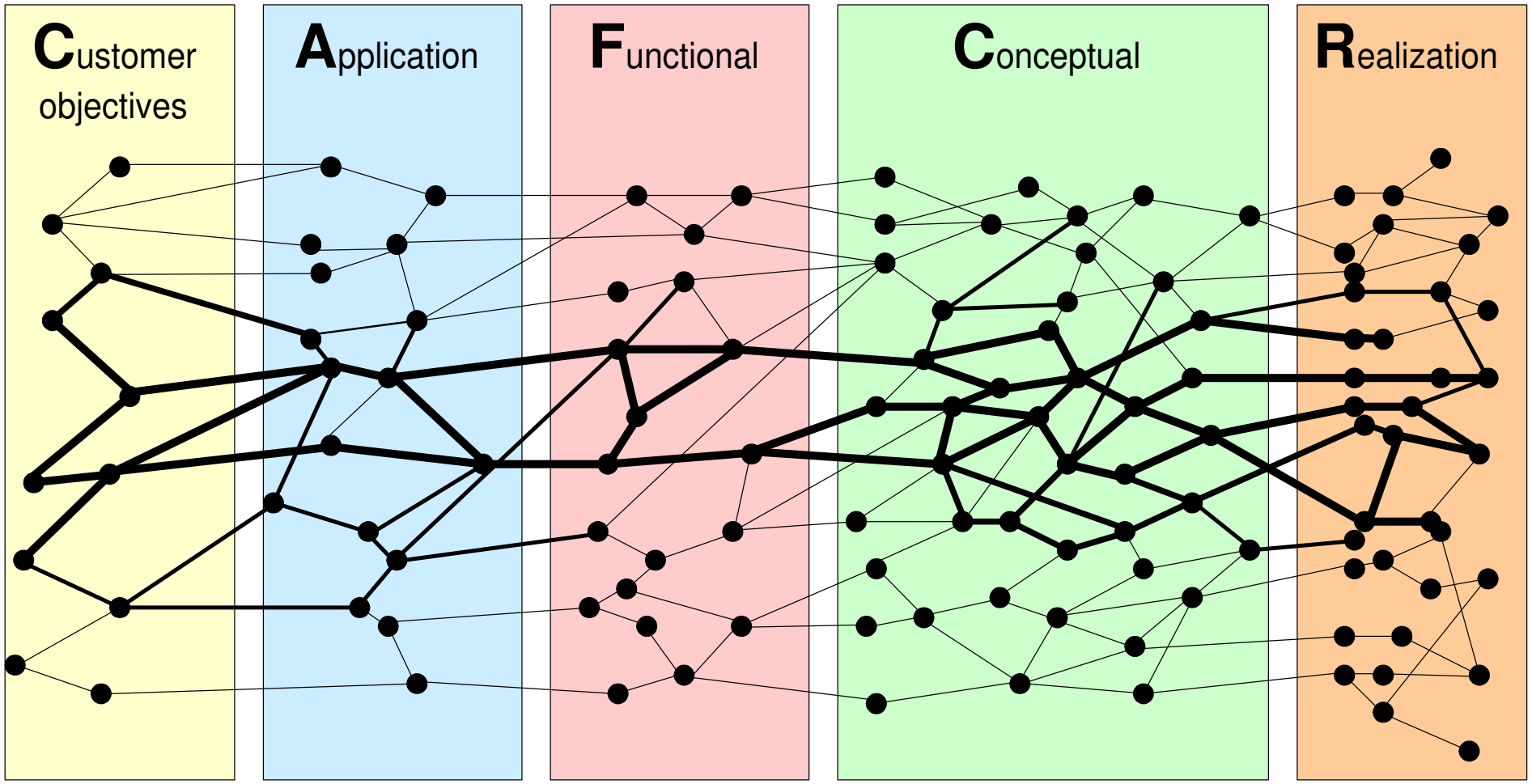
Problem identification and articulation



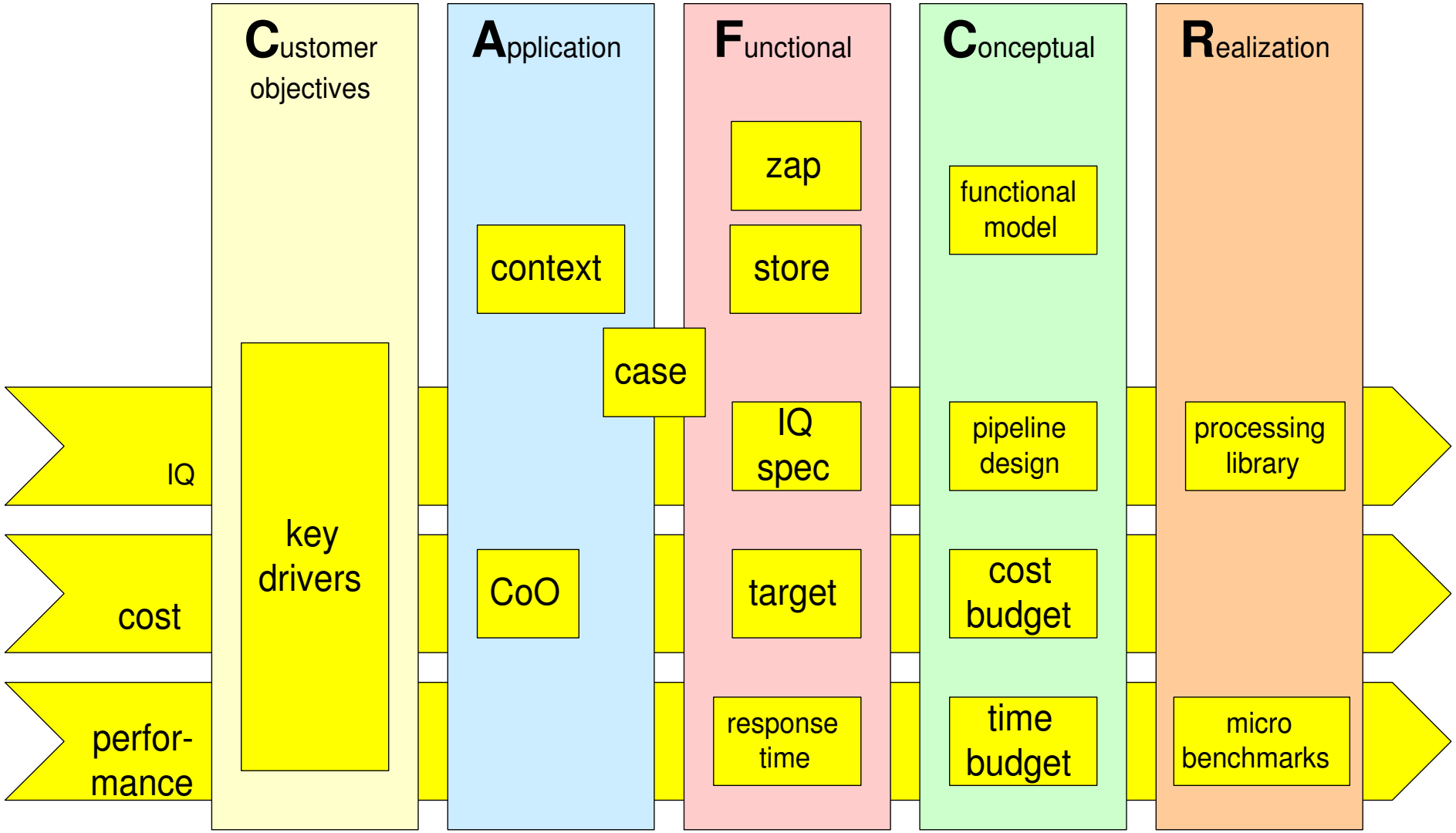
Iteration during the analysis



Thread of related issues



Documentation and communication structure



Threads of reasoning illustrated by medical imaging case

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

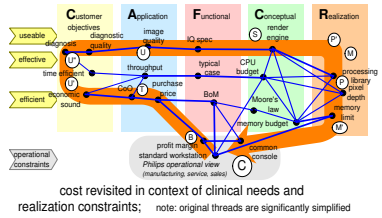
Abstract

The medical imaging workstation case is introduced. An architecting method based on the CAFCR viewpoints is explained, consisting of 4 elements:

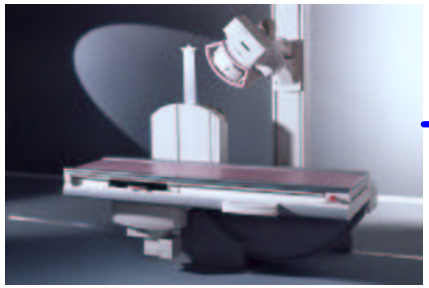
- the CAFCR viewpoints
- qualities as integrating needles
- story telling
- threads of reasoning

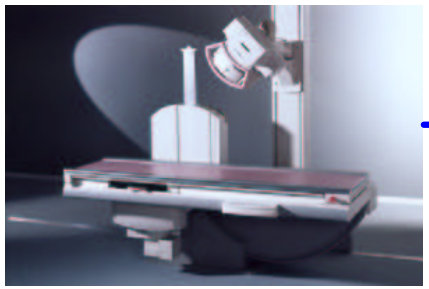
A thread of reasoning is build up in steps, based on this case. The underlying reasoning is explained.

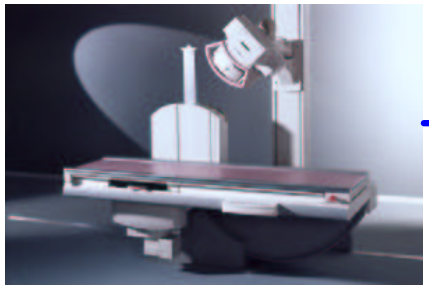
December 21, 2011
status: preliminary draft
version: 0

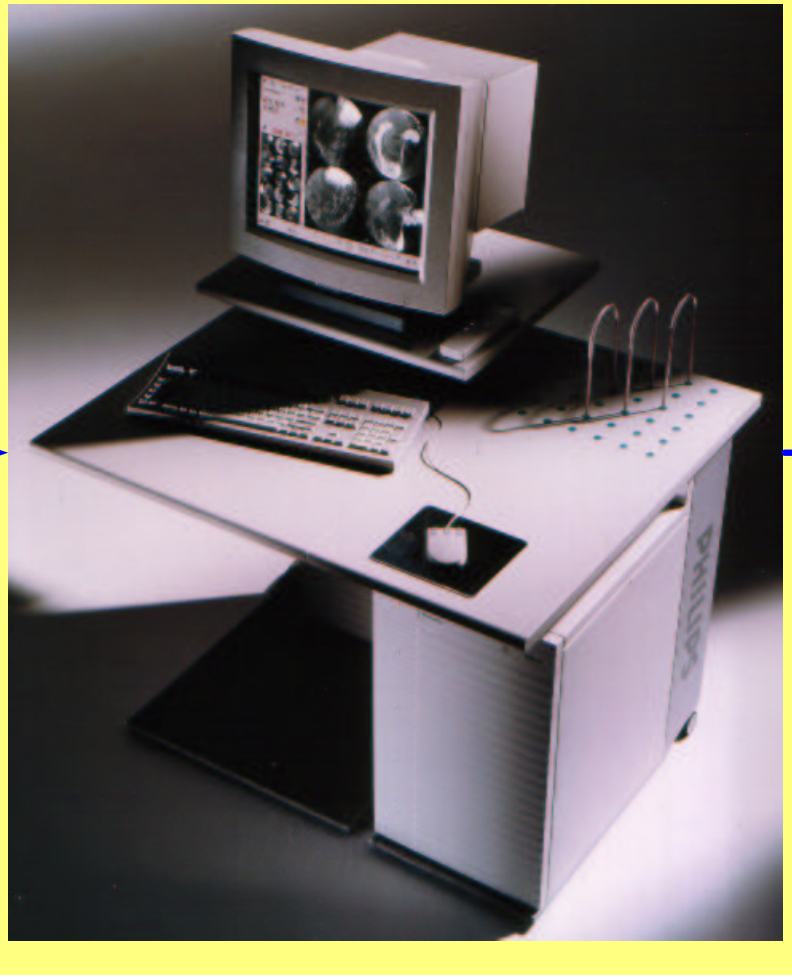


Easyvision serving three URF examination rooms







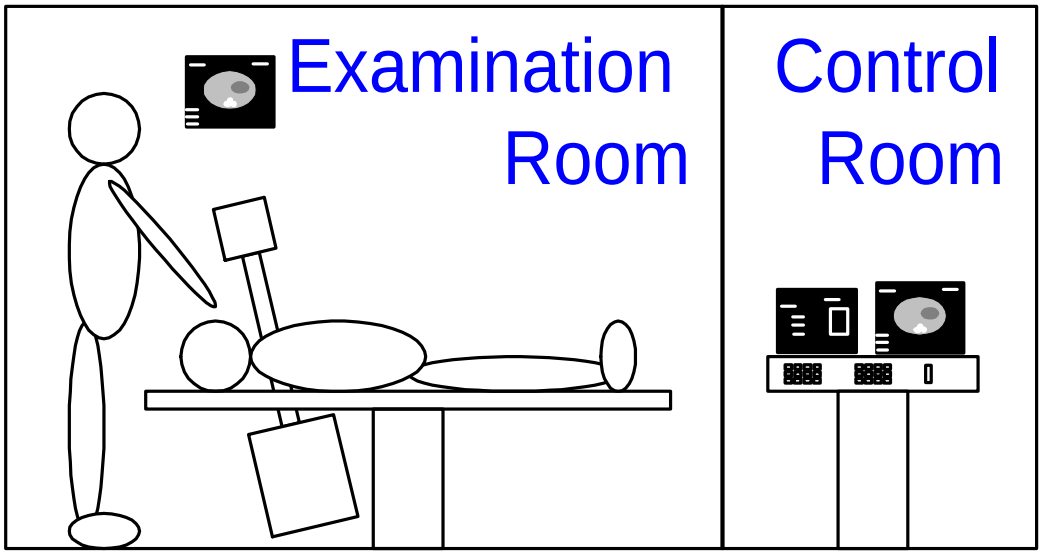


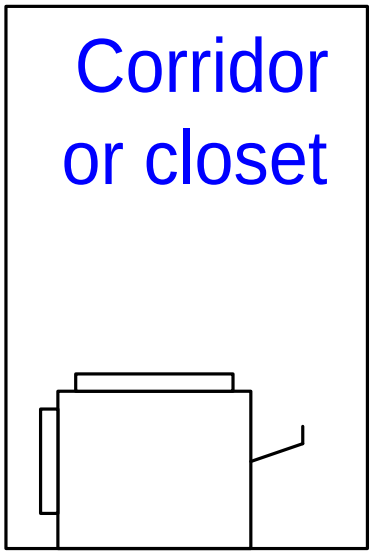
typical clinical image (intestines)

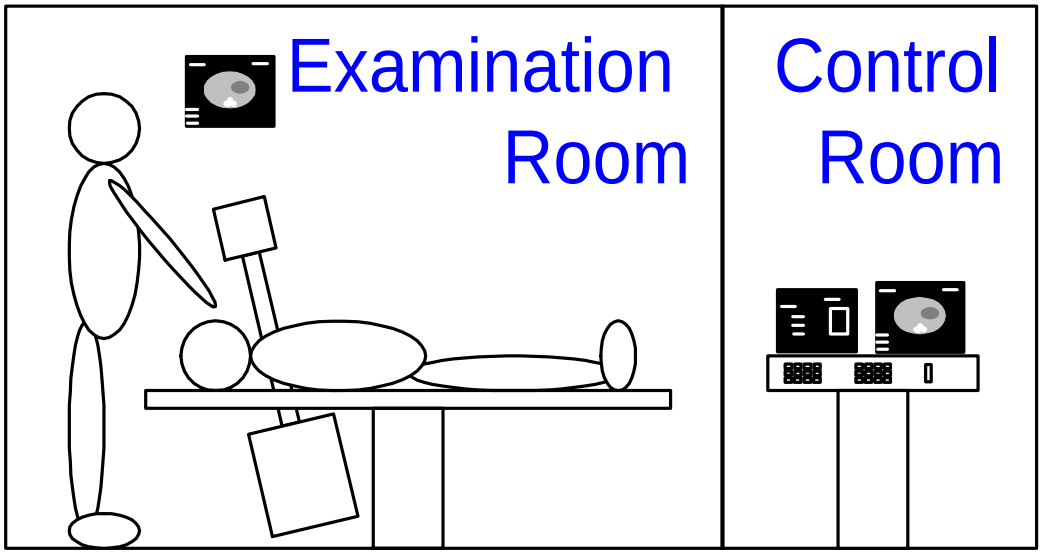
URF-systems

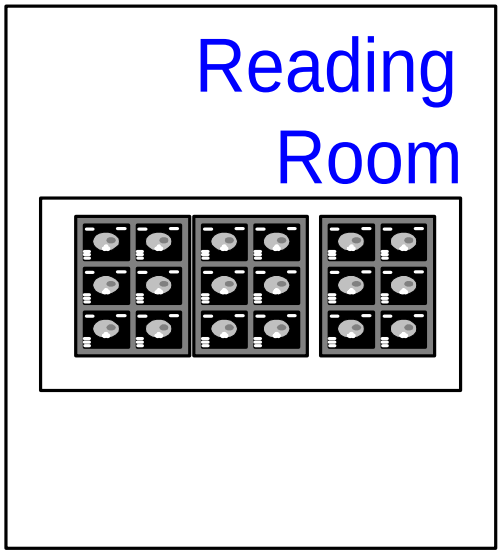
EasyVision: Medical Imaging Workstation

X-ray rooms from examination to reading around 1990

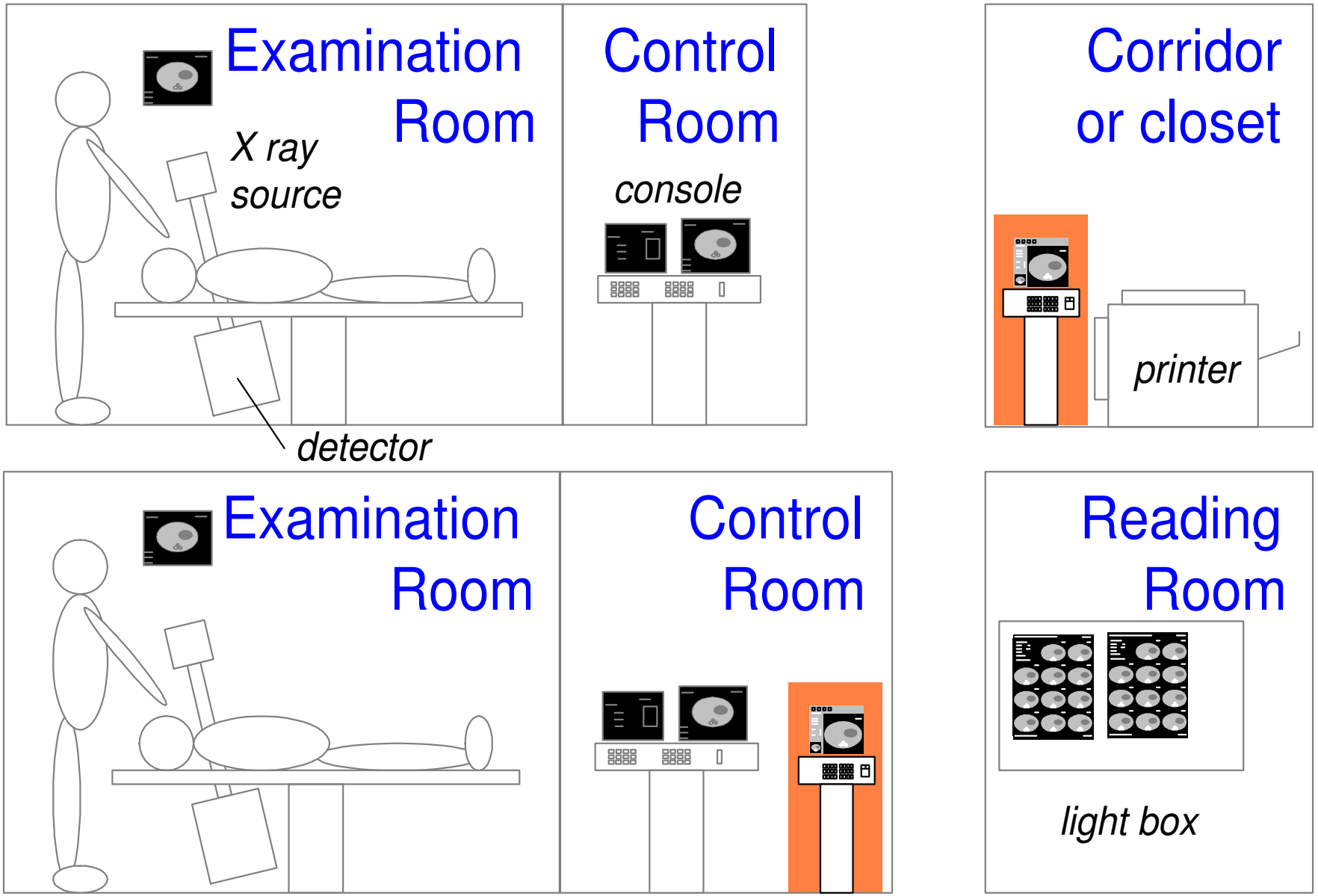




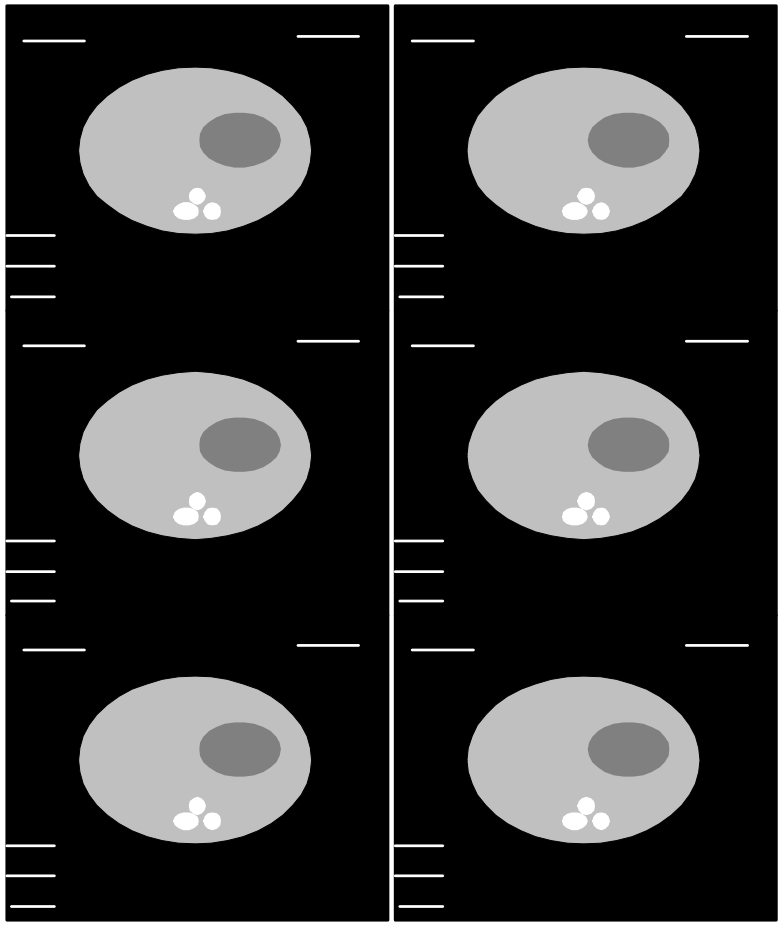




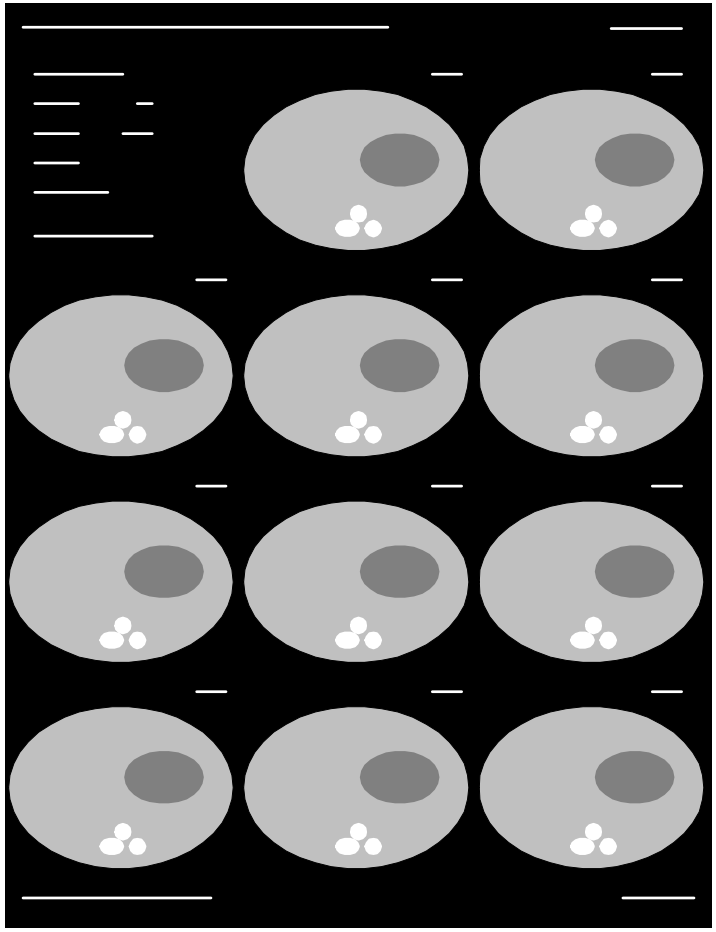
X-ray rooms with Easyvision applied as printserver



Comparison screen copy versus optimized film



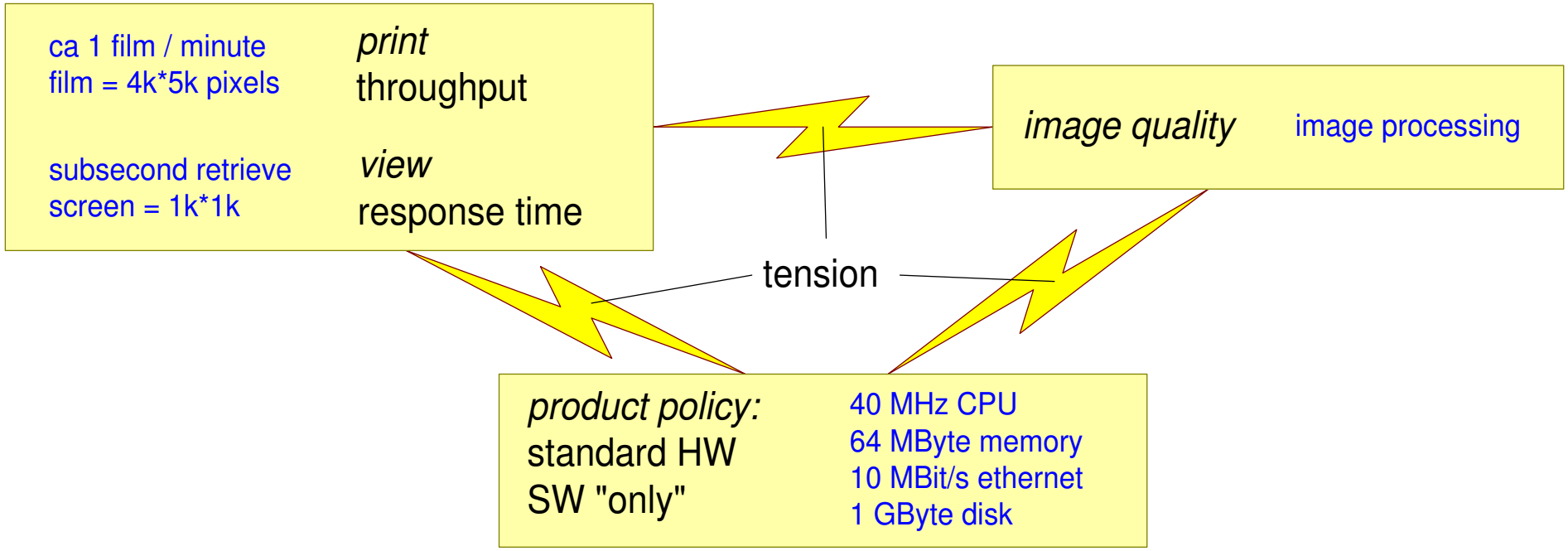
old: screen copy



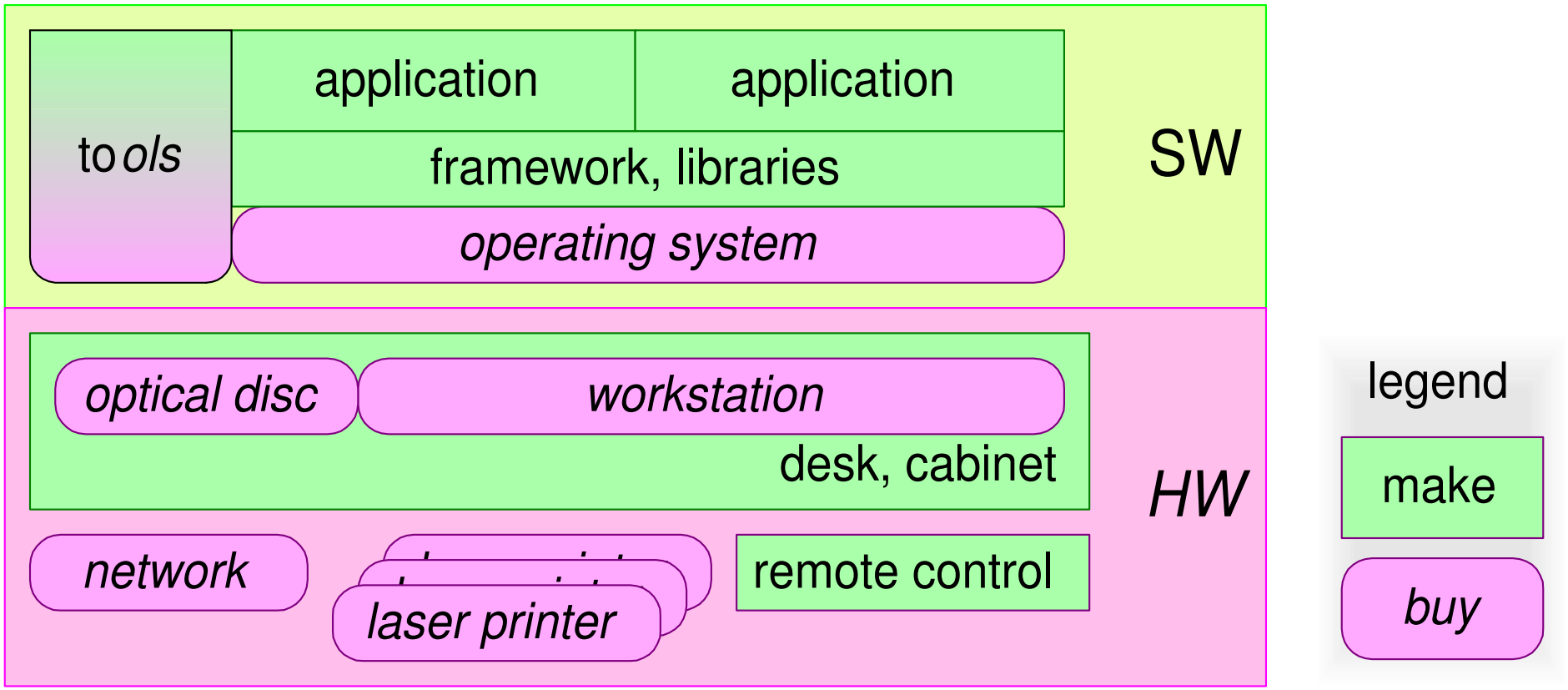
new: SW formatting

20 to 50% less film needed

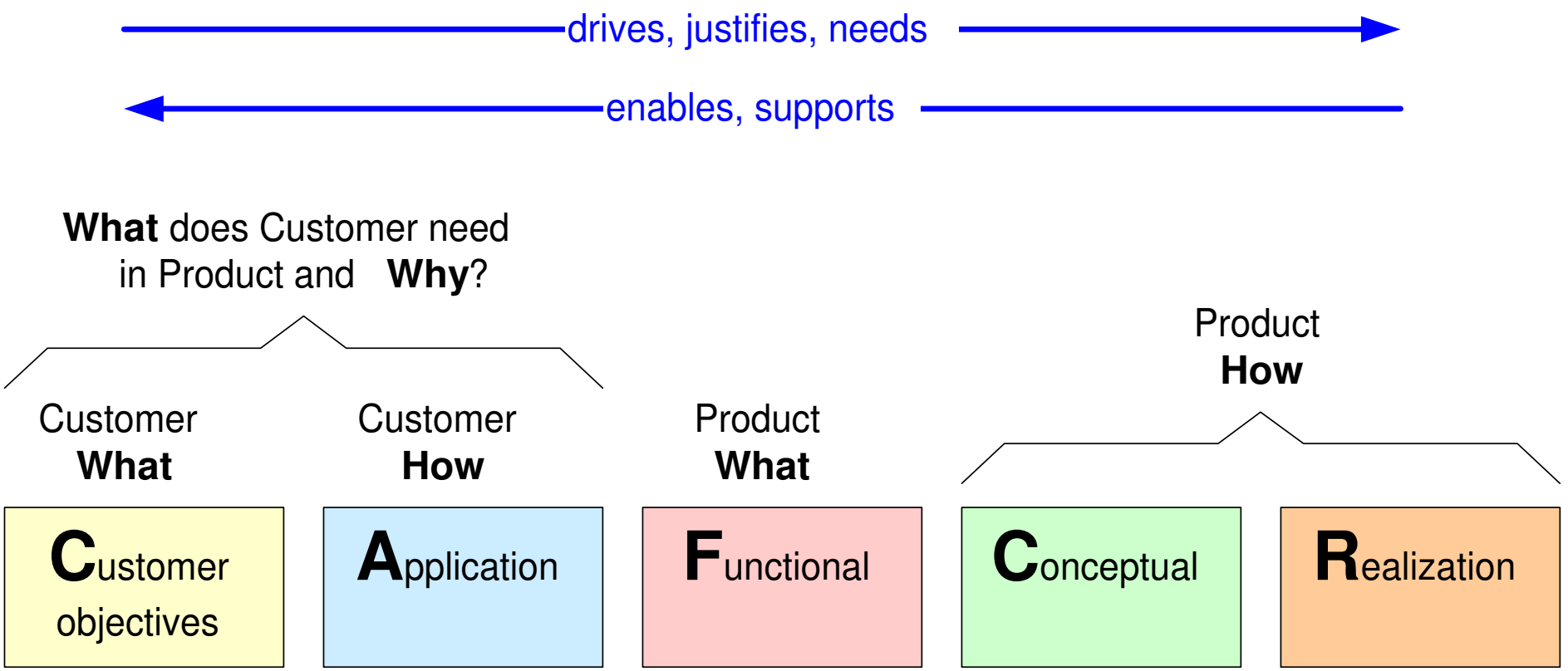
Challenges for product creation



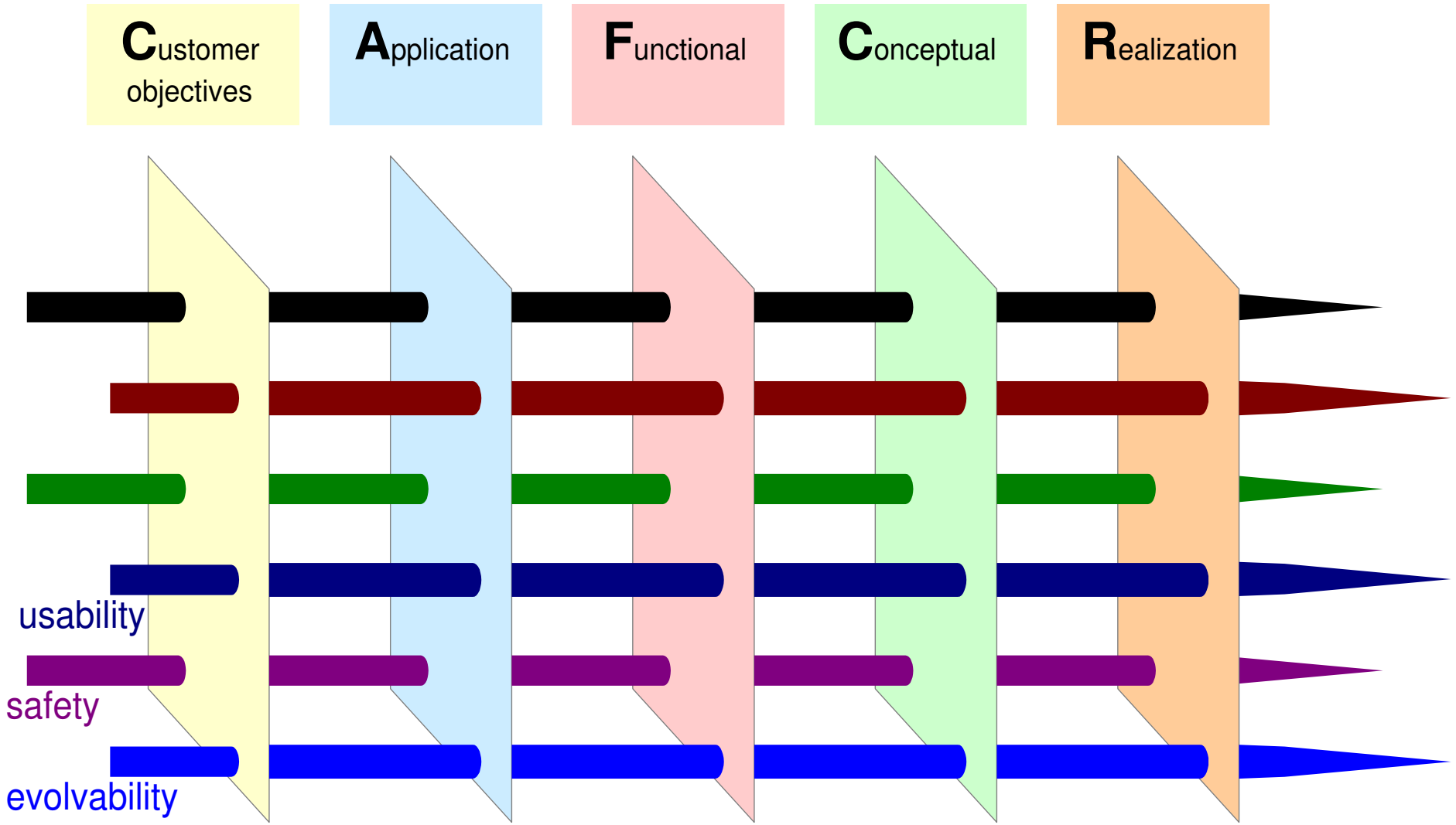
Top level decomposition



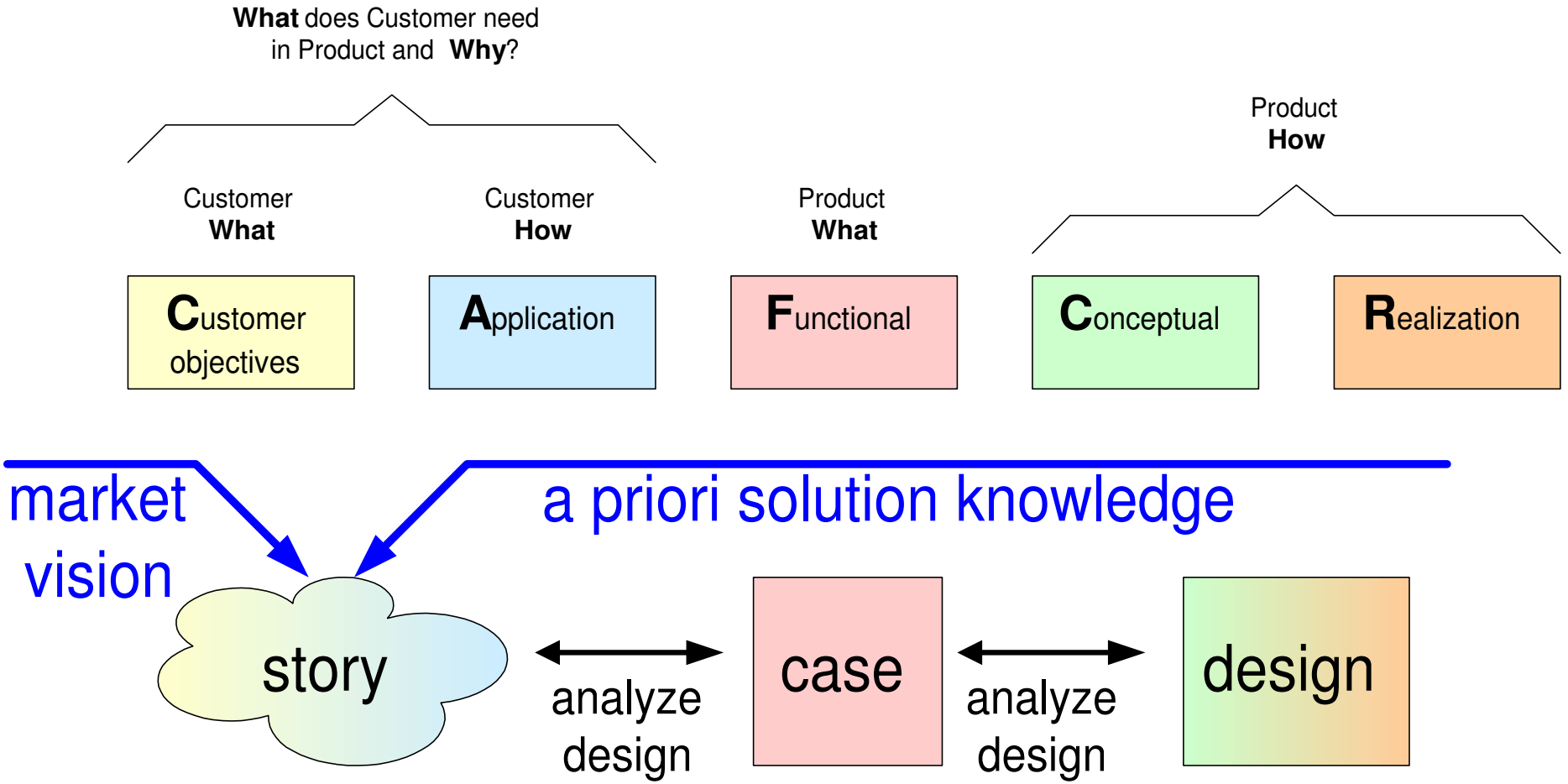
CAFCR viewpoints



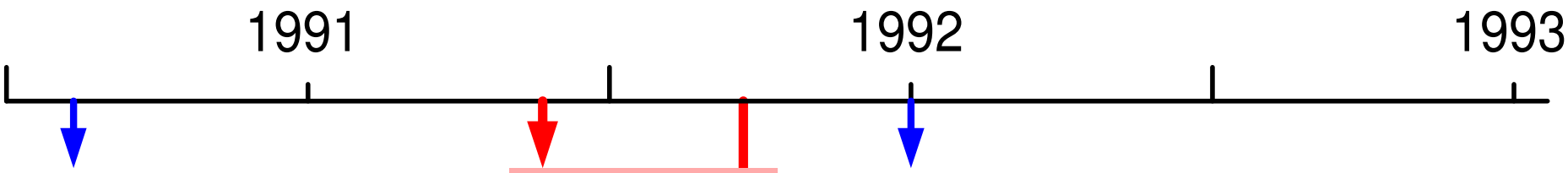
Quality needles as generic integrating concepts



From story to design



Chronology of Easyvision RF R1 development



basic application
toolboxes
100 kloc

interactive viewing

performance
problems

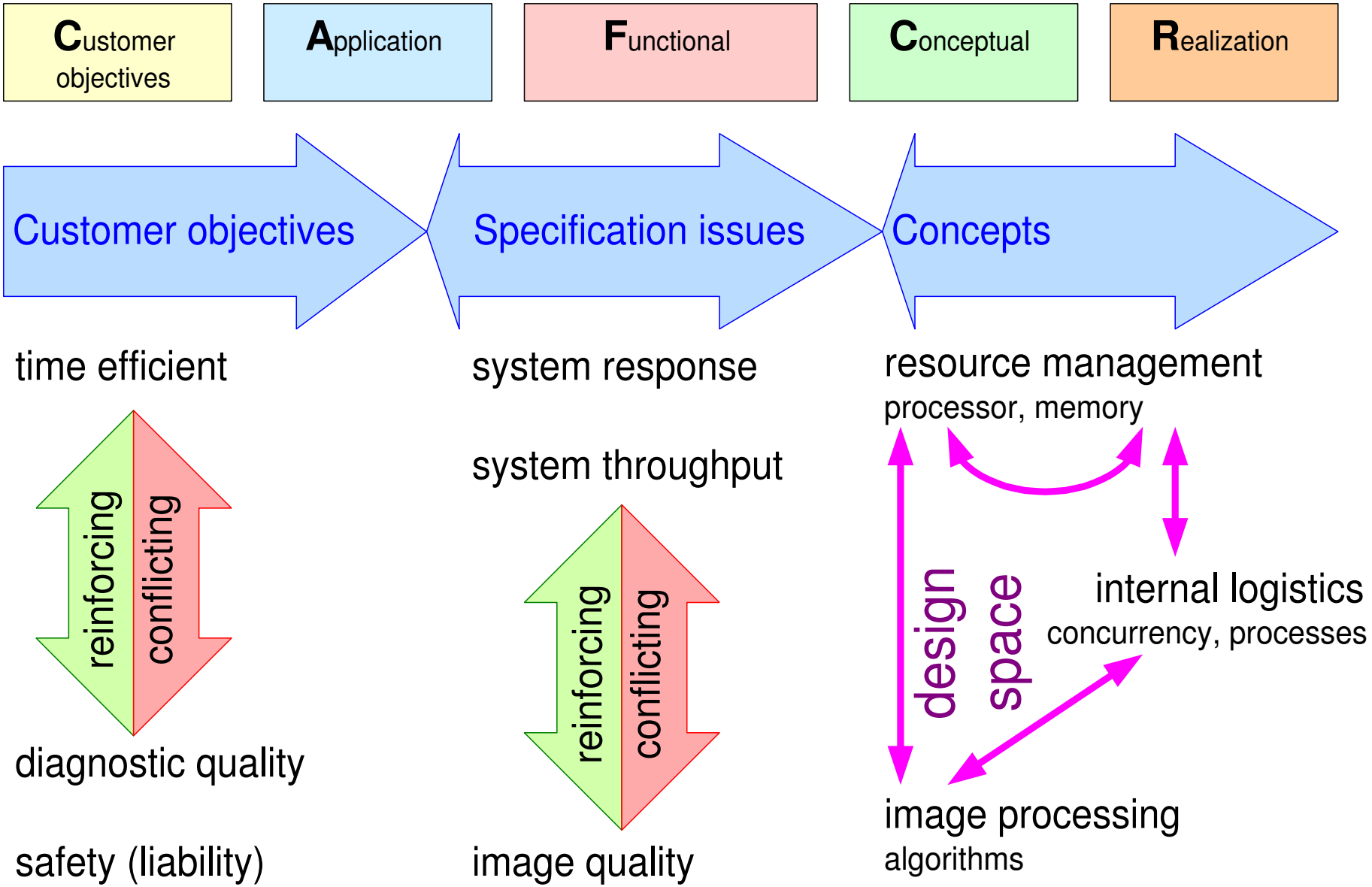
IQ
problems

Easyvision RF
integrated product
360 kloc

print server +
communication +
interactive viewing

marketing opinion:
"All the functionality is available,
we only have to provide a clinical UI"

Thread of reasoning based on efficiency-quality tension



Technology innovations

performance

cost

standard UNIX based workstation

full SW implementation, more flexible

object oriented design and implementation (Objective-C)

graphical User Interface, with windows, mouse etcetera

call back scheduling, fine-grained notification

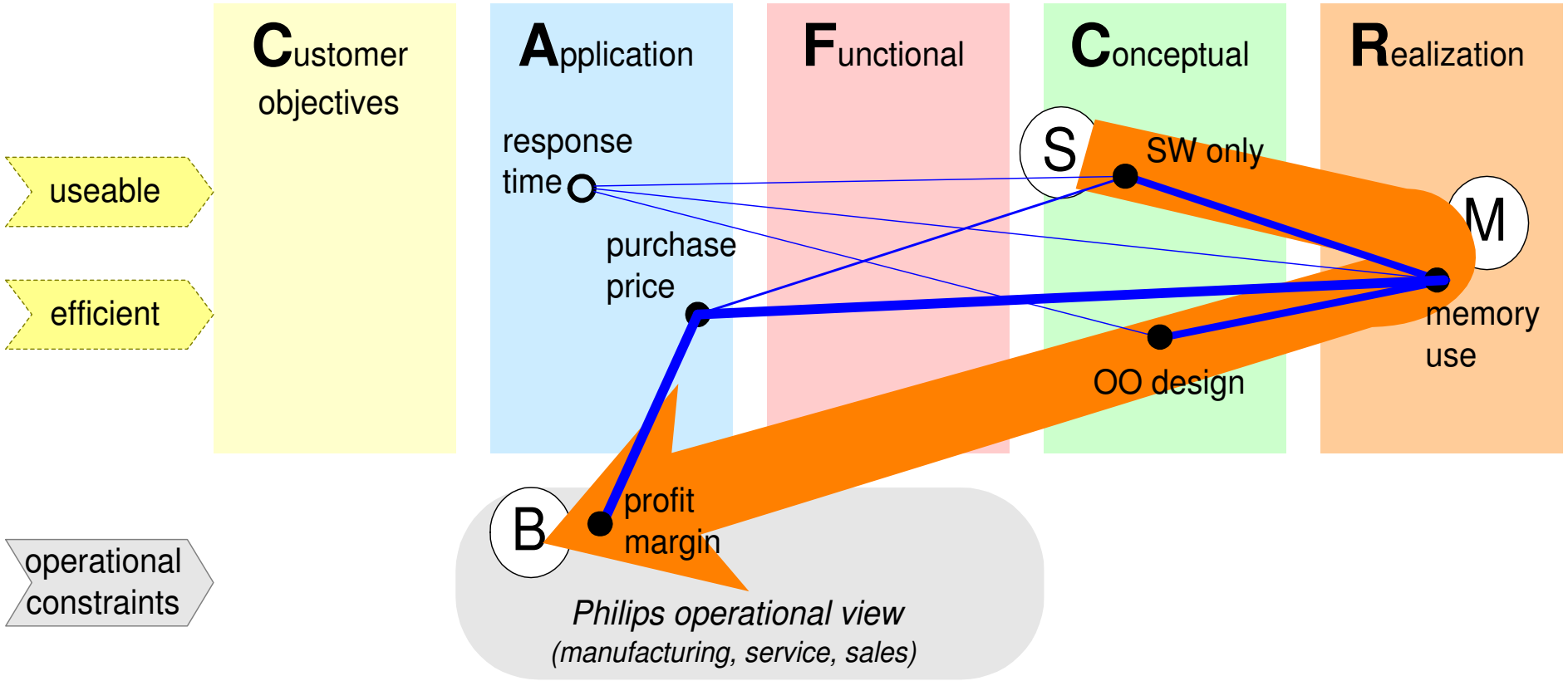
data base engine, fast, reliable and robust

extensive set of toolboxes

property based configuration

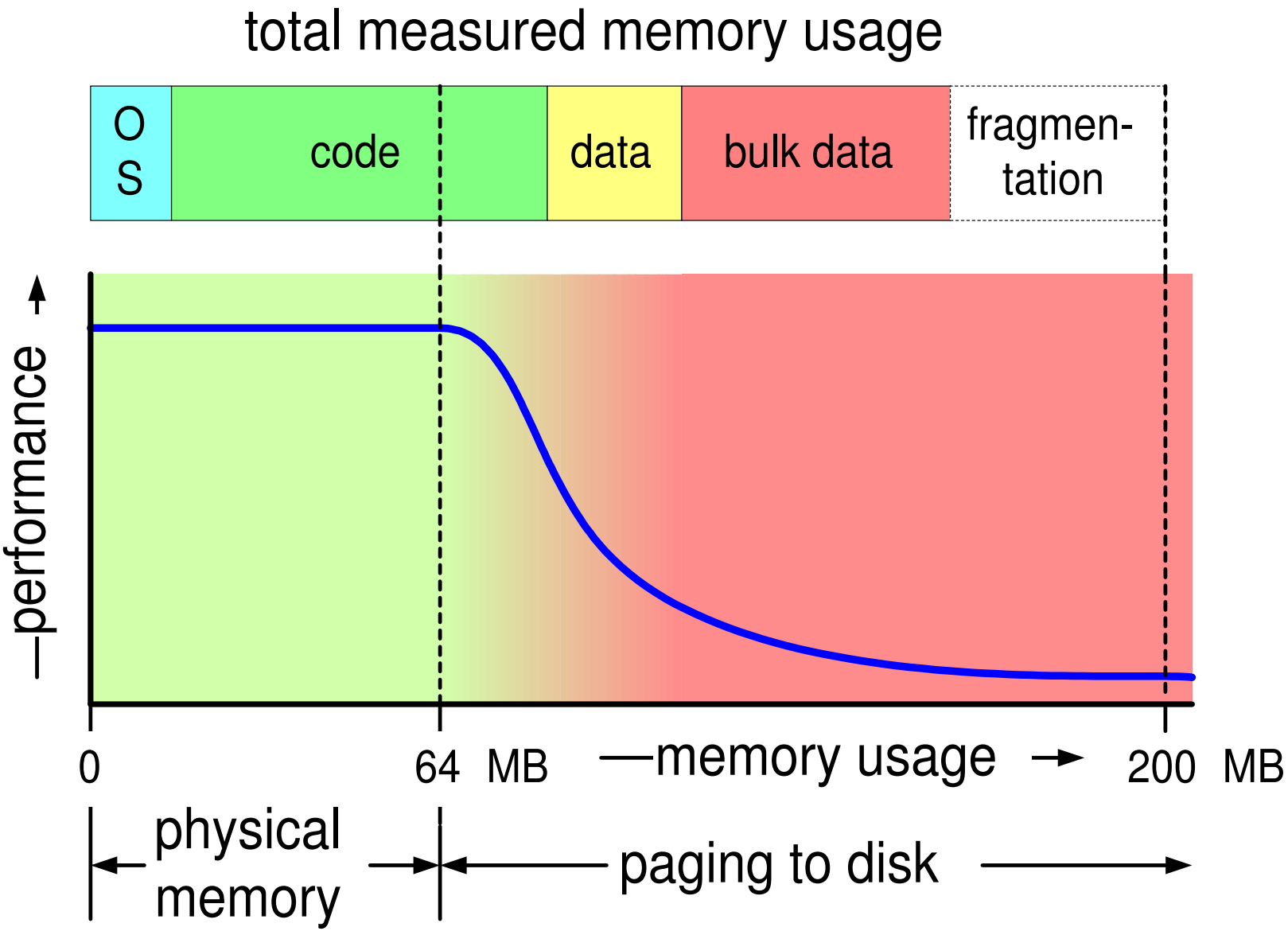
multiple coordinate spaces

Thread of reasoning; introvert phase

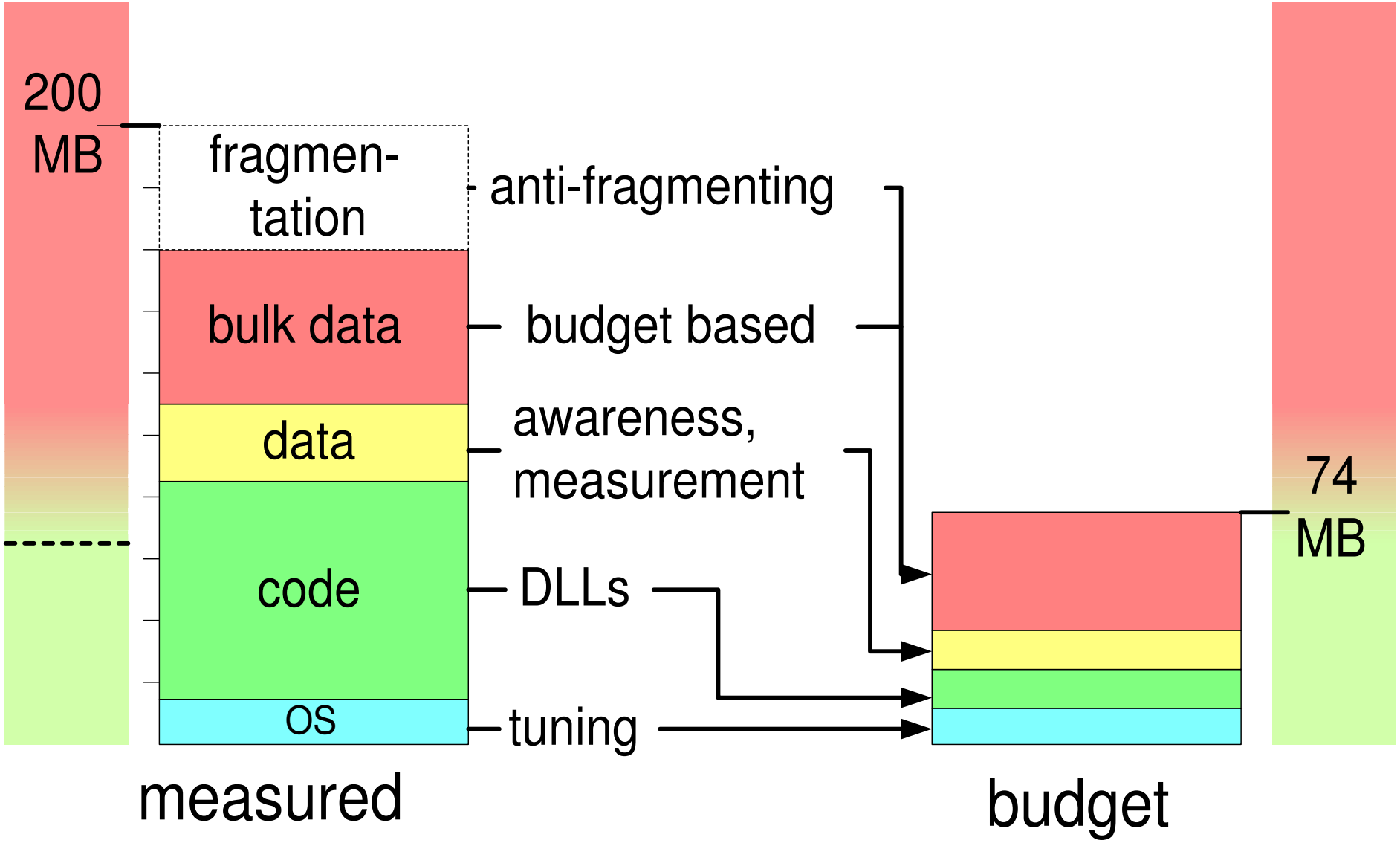


Introvert view: cost and impact of new technologies

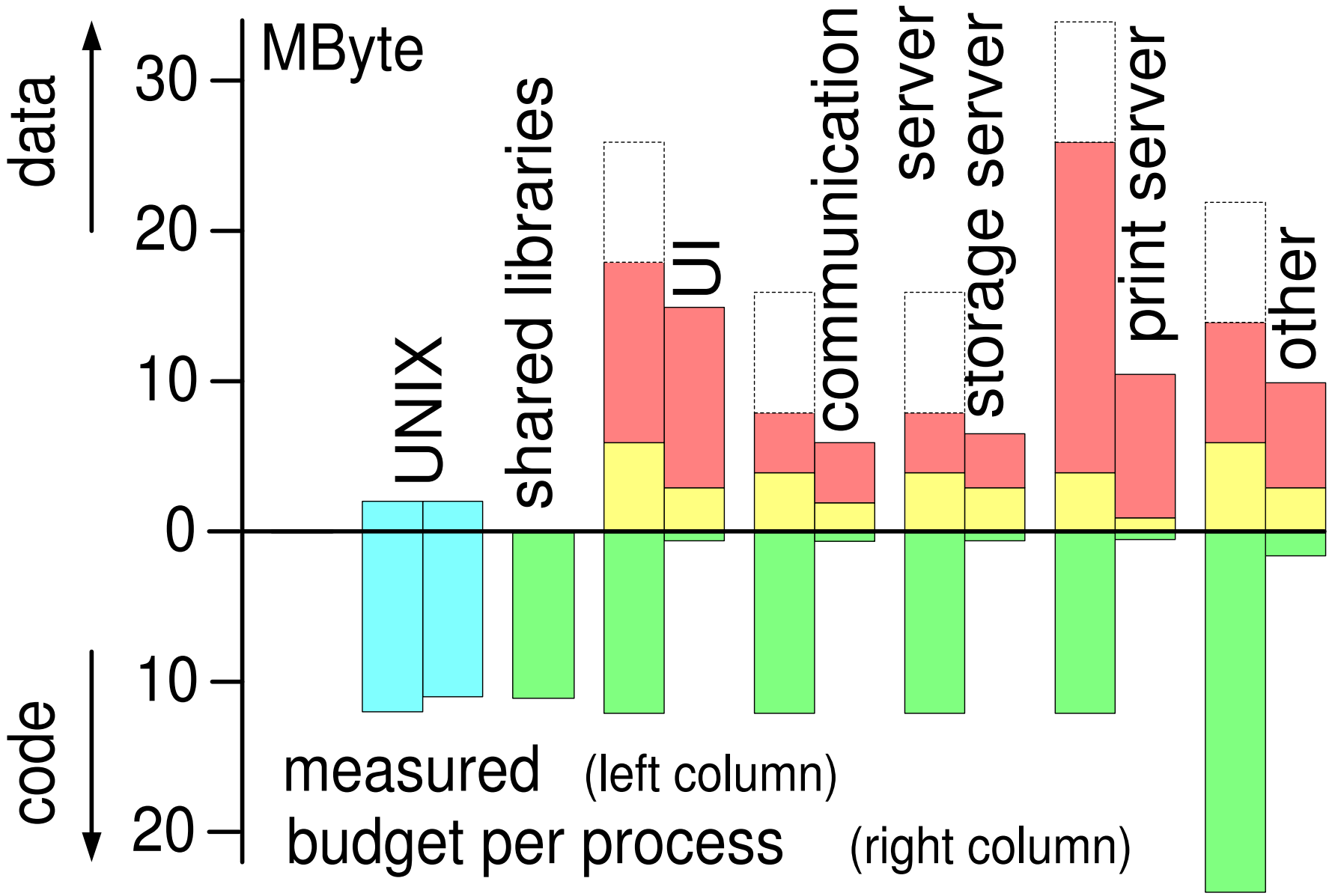
Memory usage half way R1



Solution of memory performance problem



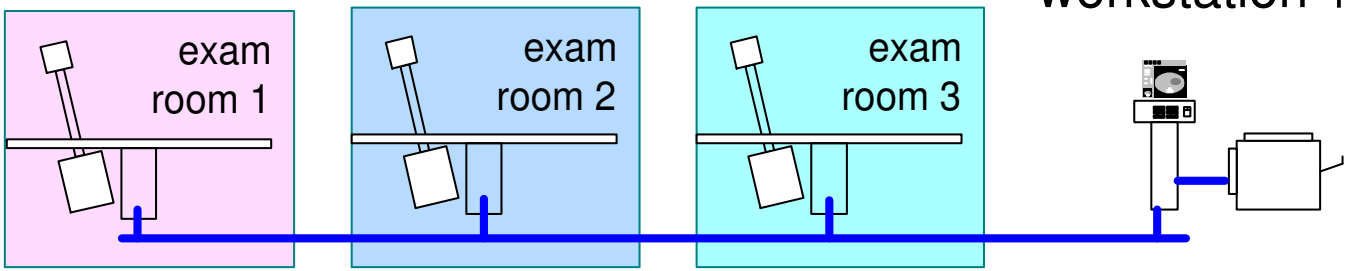
Visualization memory use per process



Typical case URF examination

3 examination rooms connected to

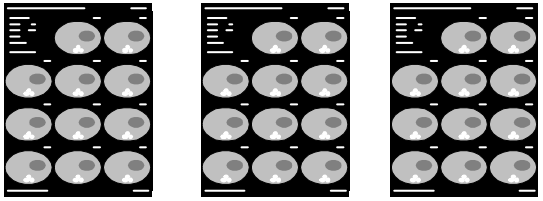
1 medical imaging workstation + printer



examination room: average 4 interleaved examinations / hour

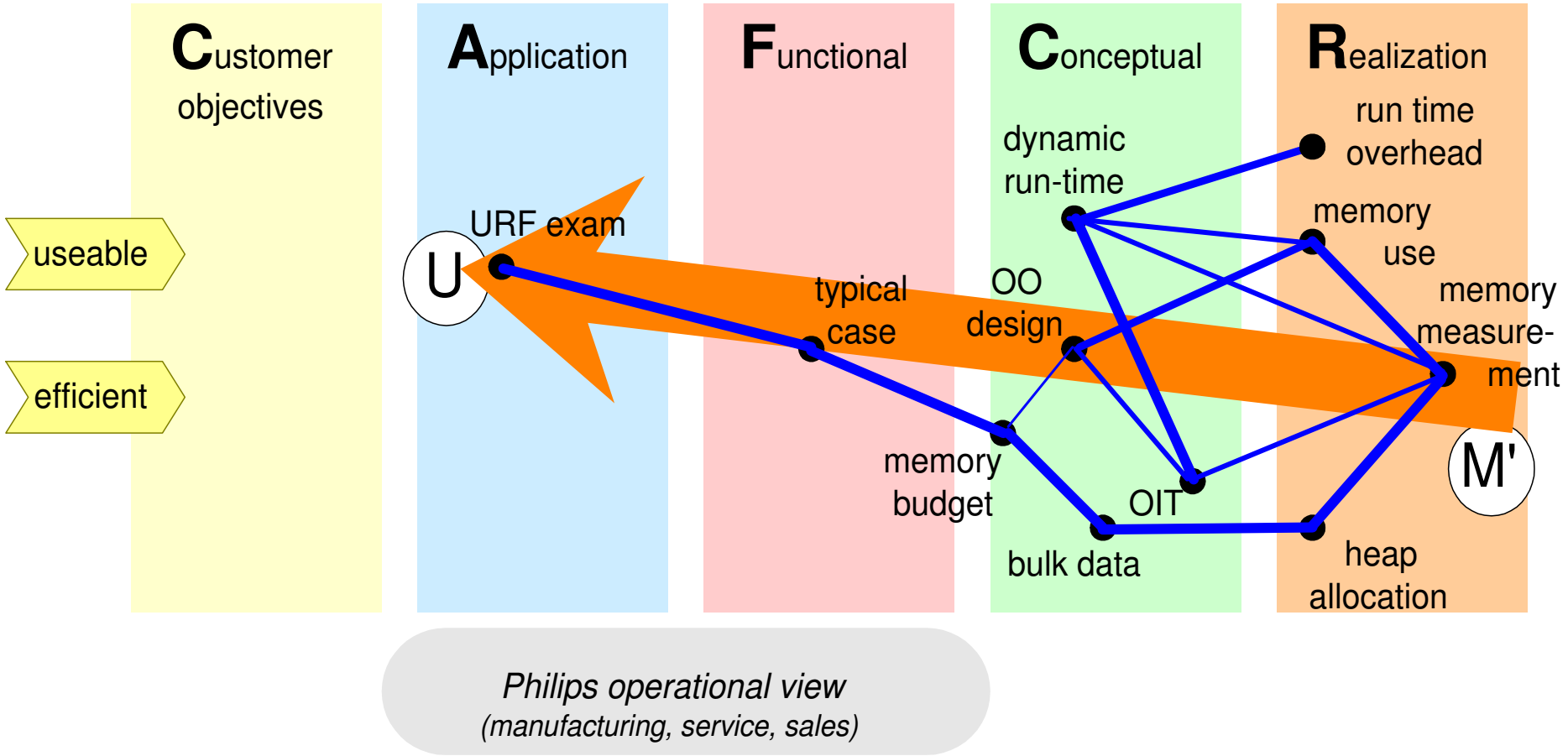
image production: 20 1024 ^{2} 8 bit images per examination

film production: 3 films of 4k*5k pixels each



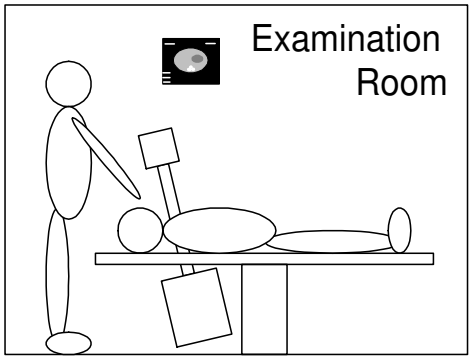
high quality output
(bi-cubic interpolation)

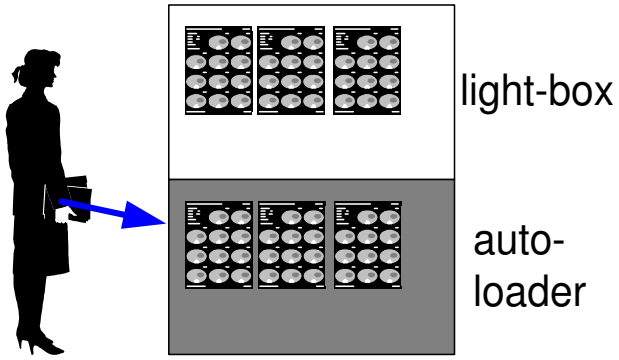
Thread of reasoning; phase 2

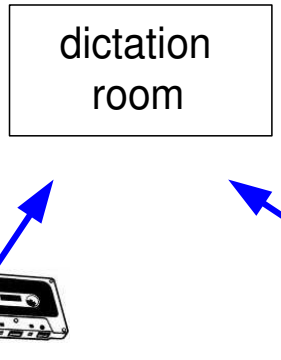


How to measure memory, how much is needed?
from introvert to extrovert

Radiologist workspots and activities







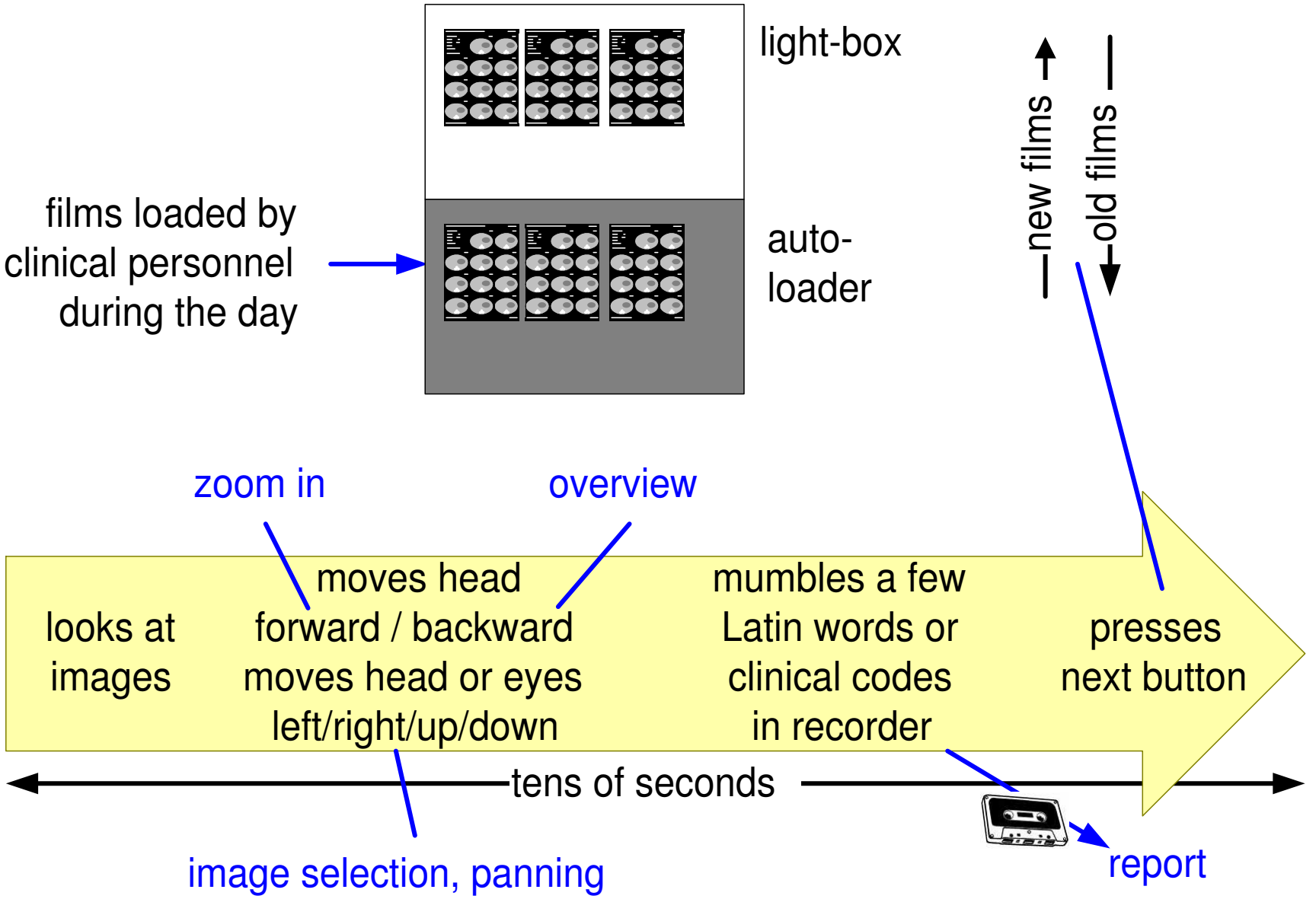
supervision of the examination

view and diagnose, dictate report

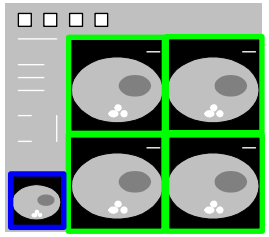
verify and authorise report

activities of the radiologist

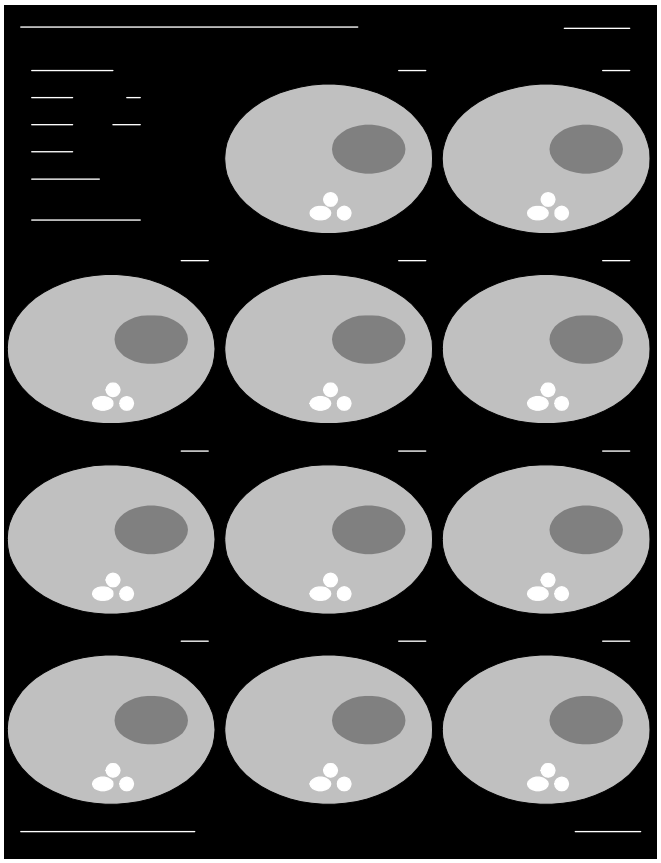
Diagnosis in tens of seconds



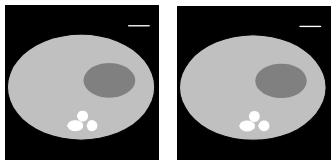
Rendered images at different destinations



Screen:
low resolution
fast response

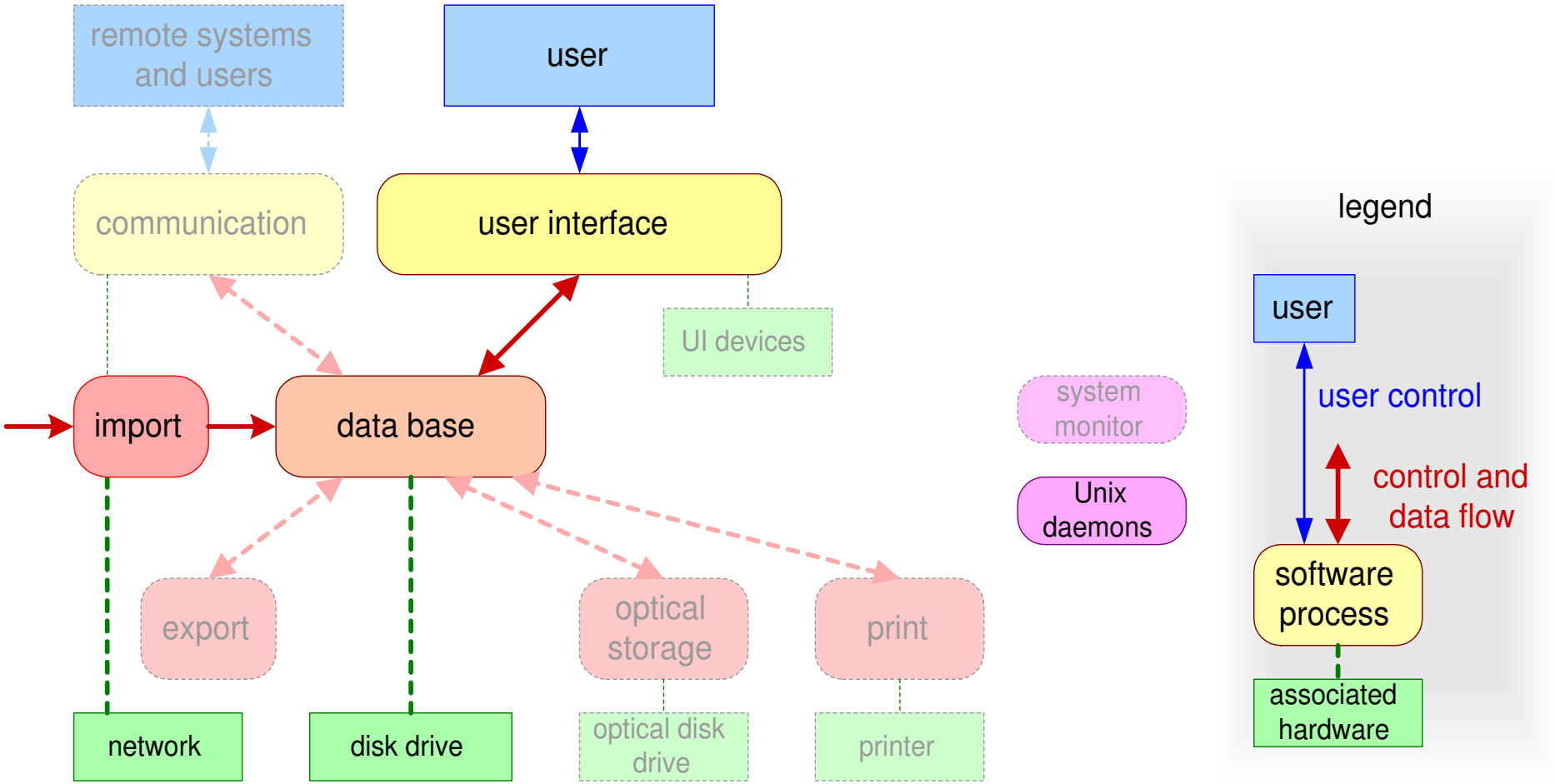


Film:
high resolution
high throughput

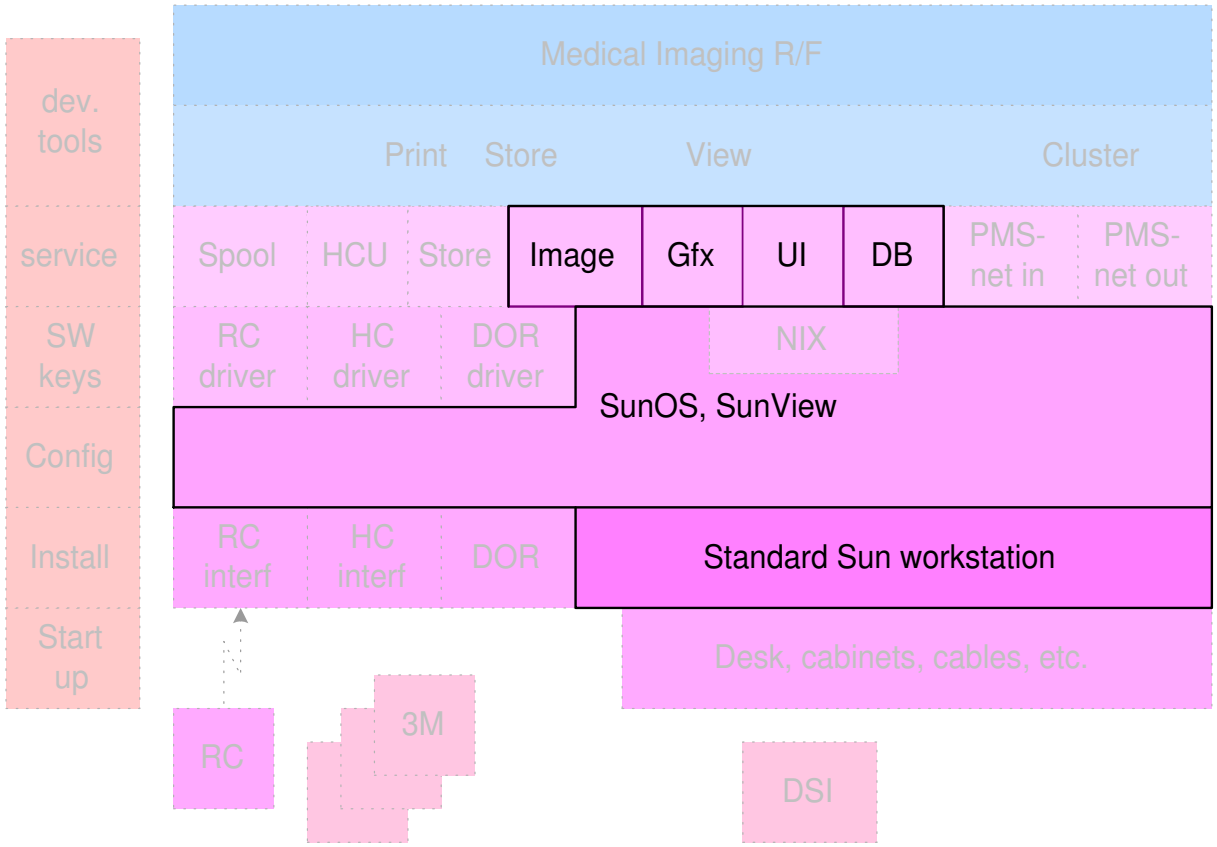


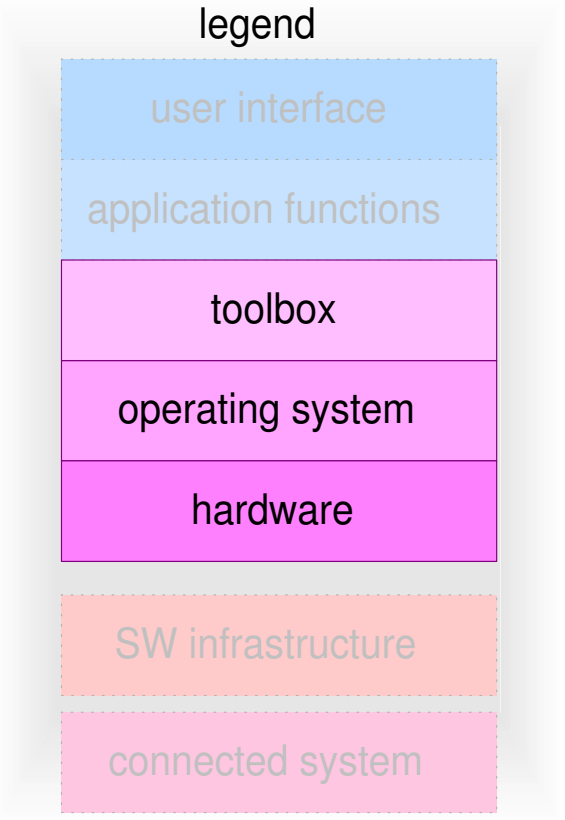
Network:
medium resolution
high throughput

SW Process structure 1991

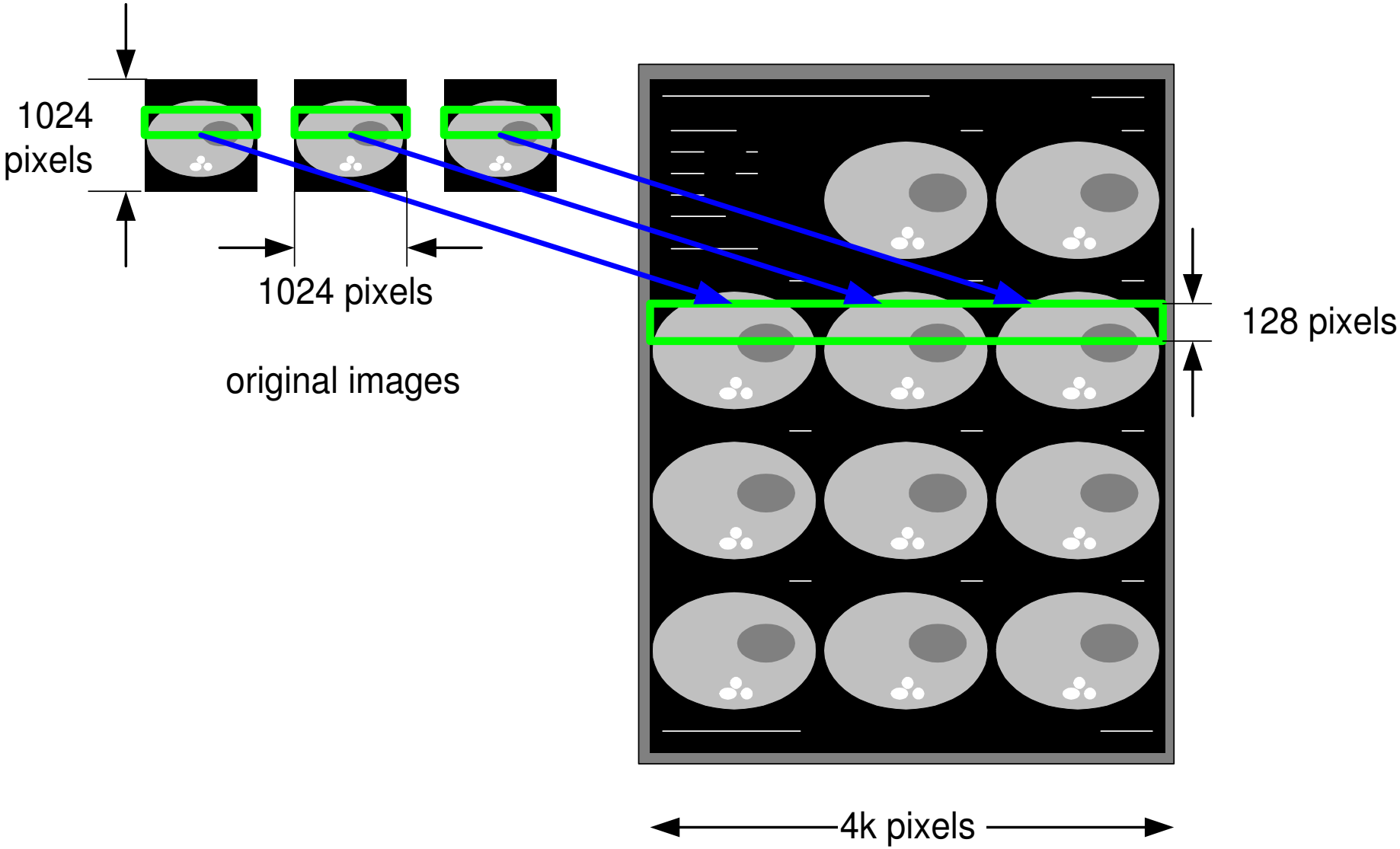


SW layers 1991

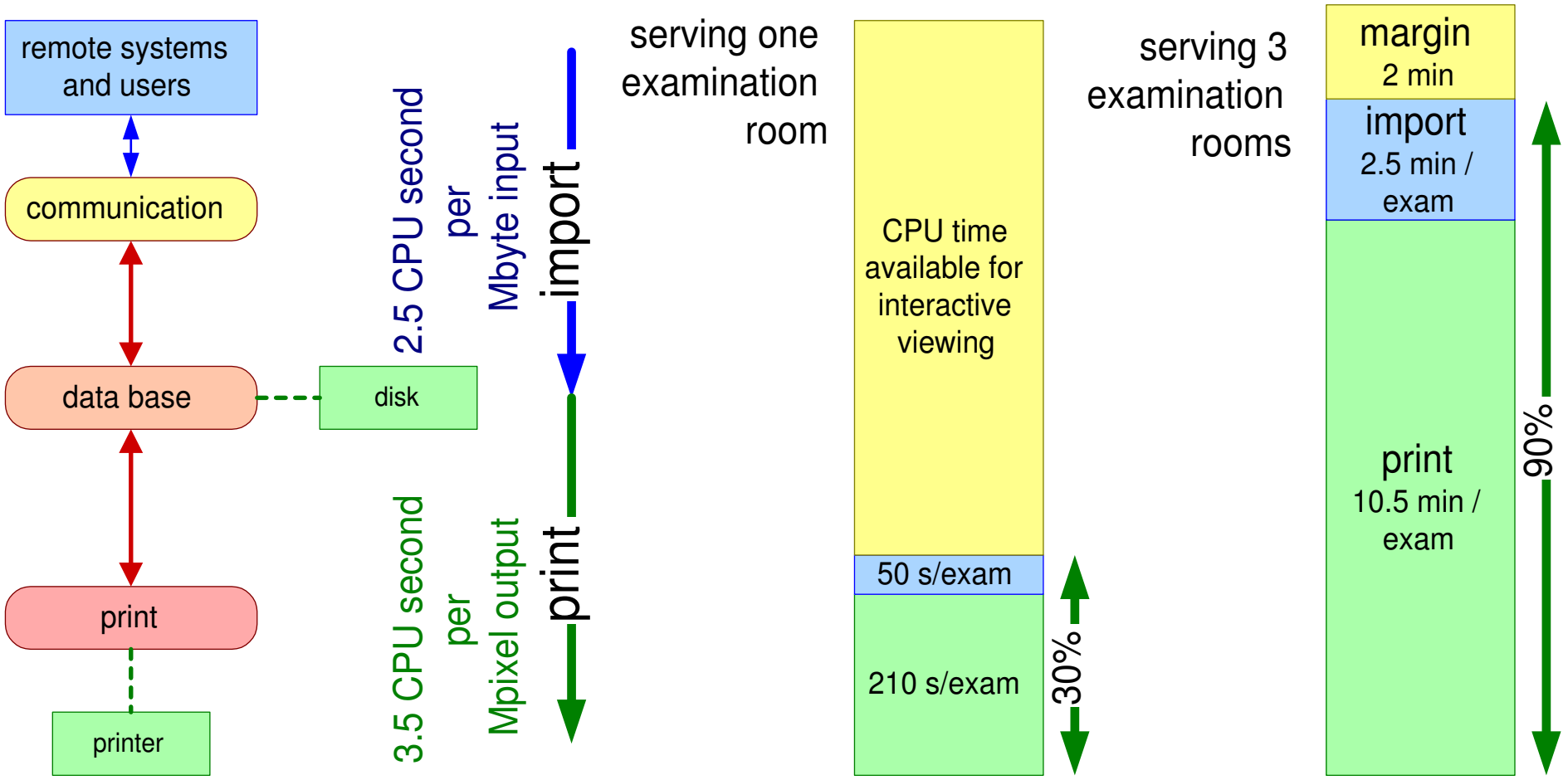




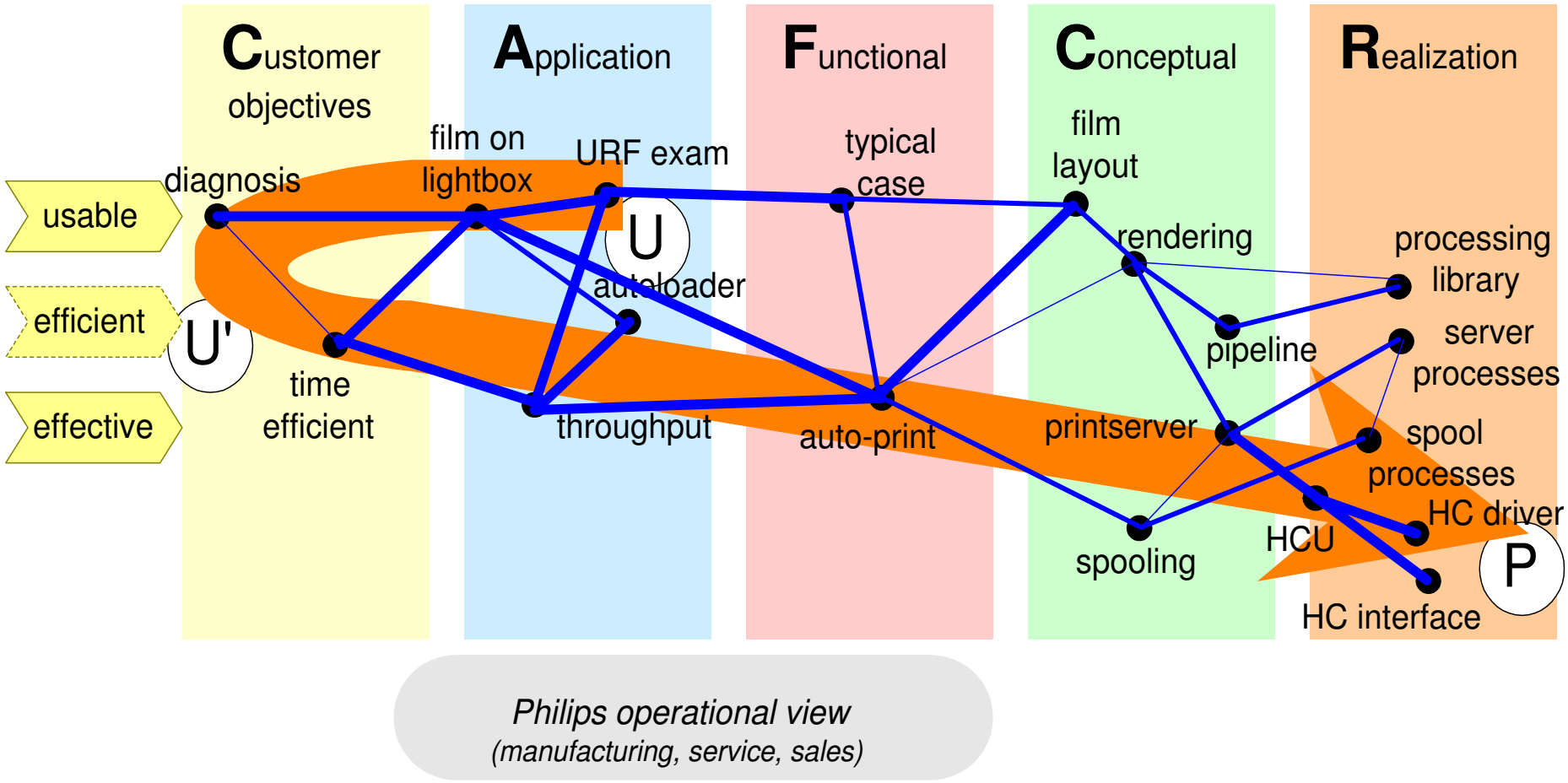
Print server is based on banding



Server CPU load



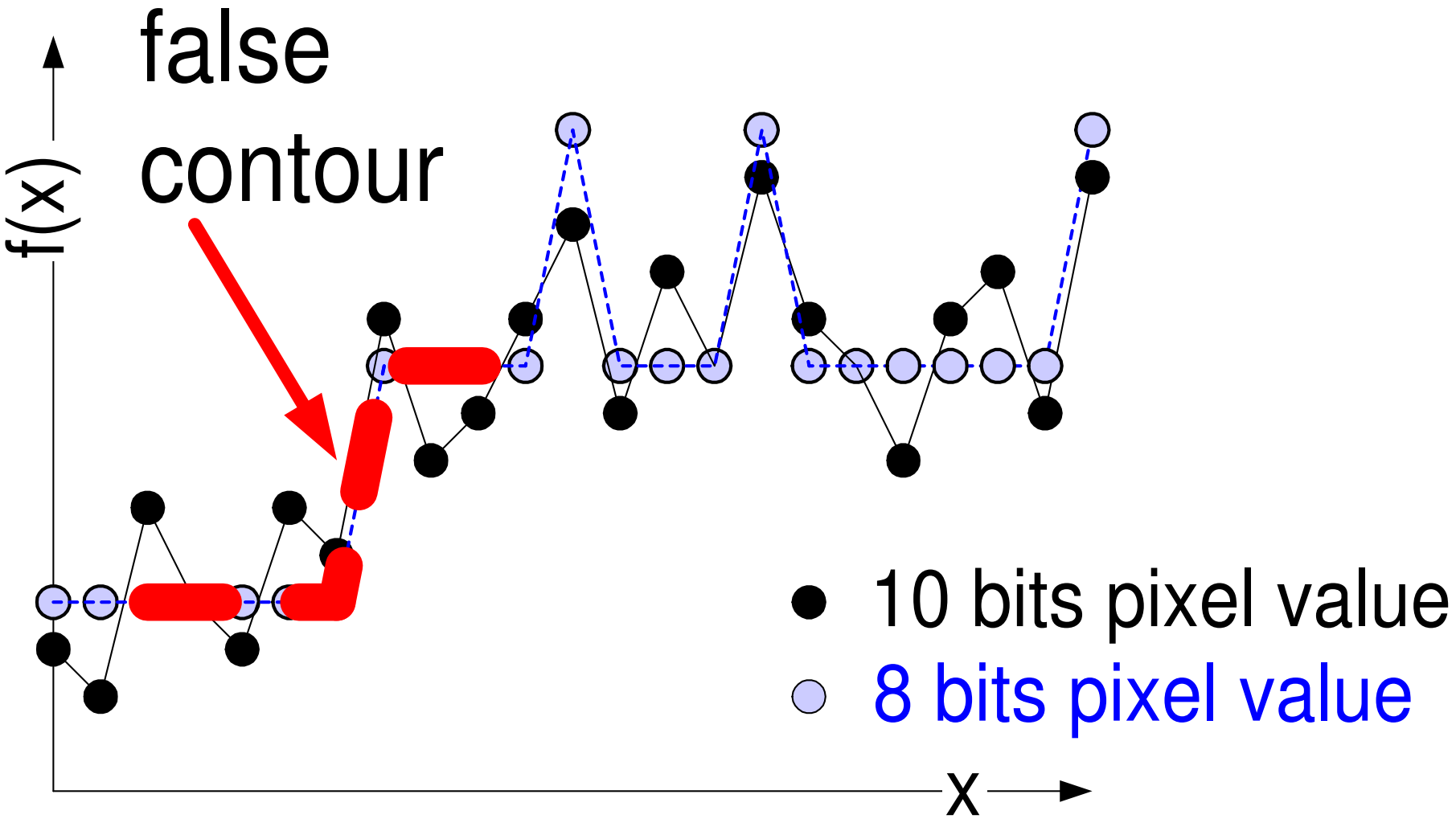
Thread of reasoning; phase 3



Philips operational view
(manufacturing, service, sales)

Radiologists diagnose from film, throughput is important
Extrovert view shows conceptual and realization gaps!

Image quality and safety problem



Presentation pipeline for X-ray images

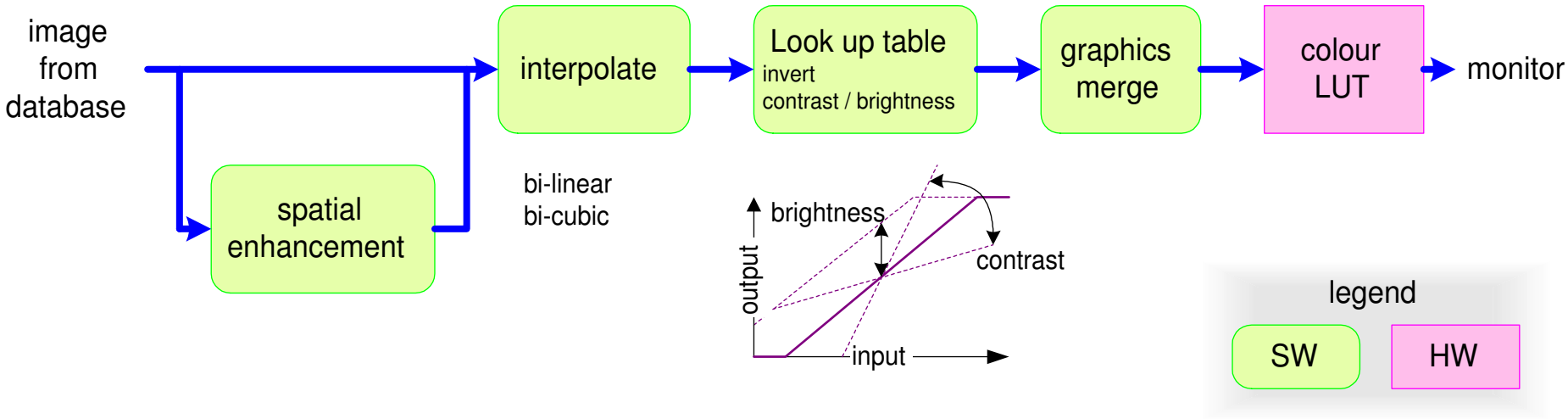
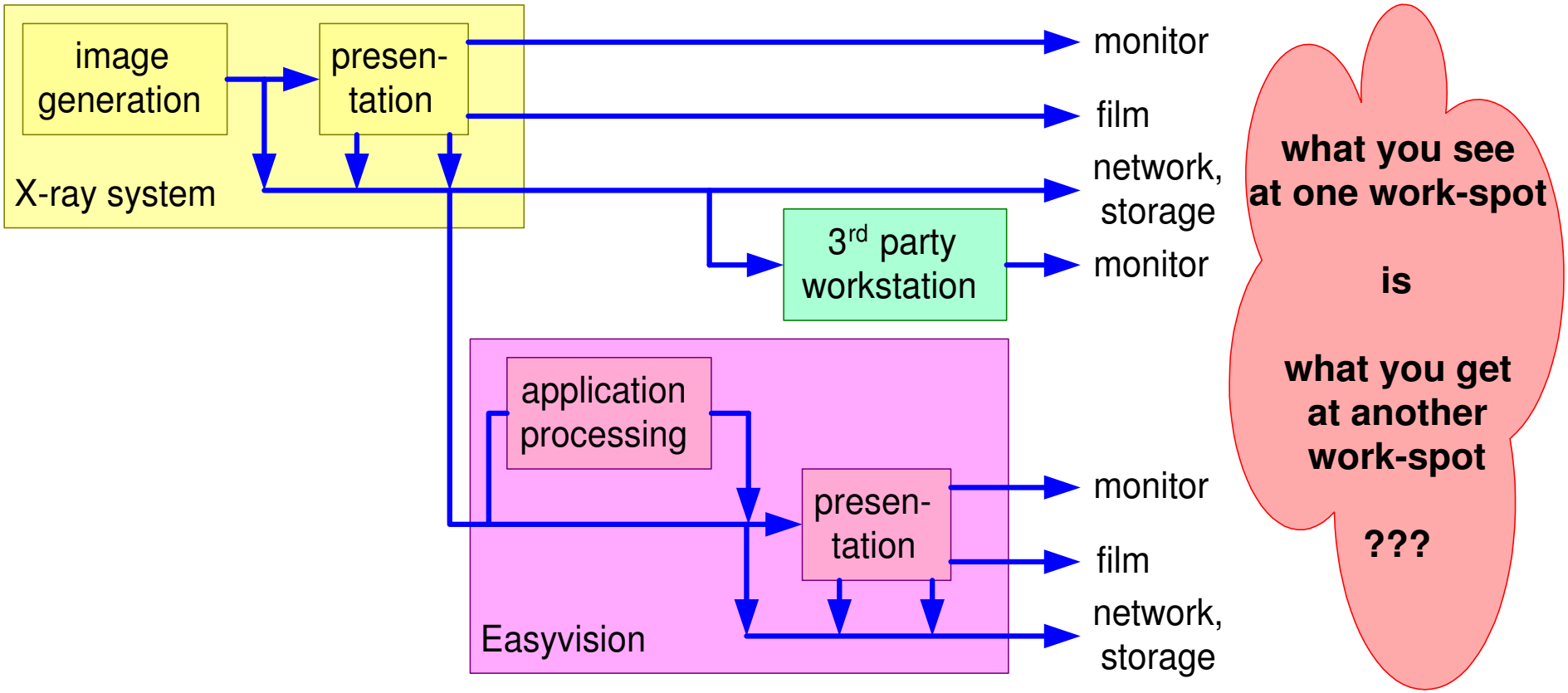
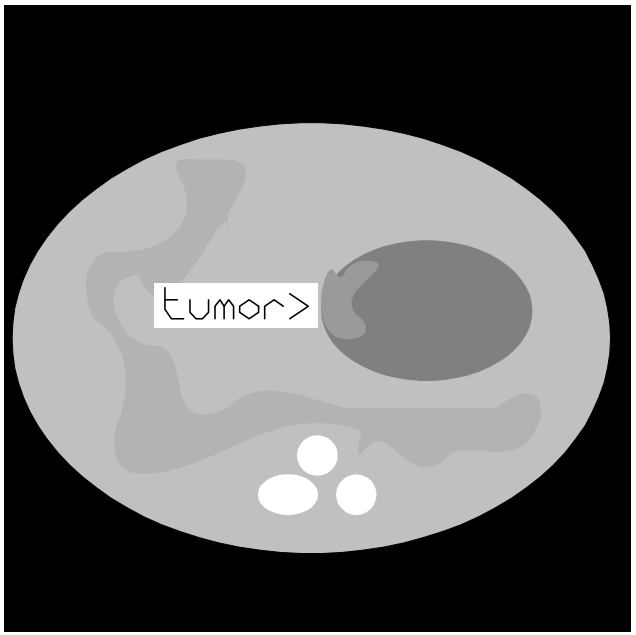
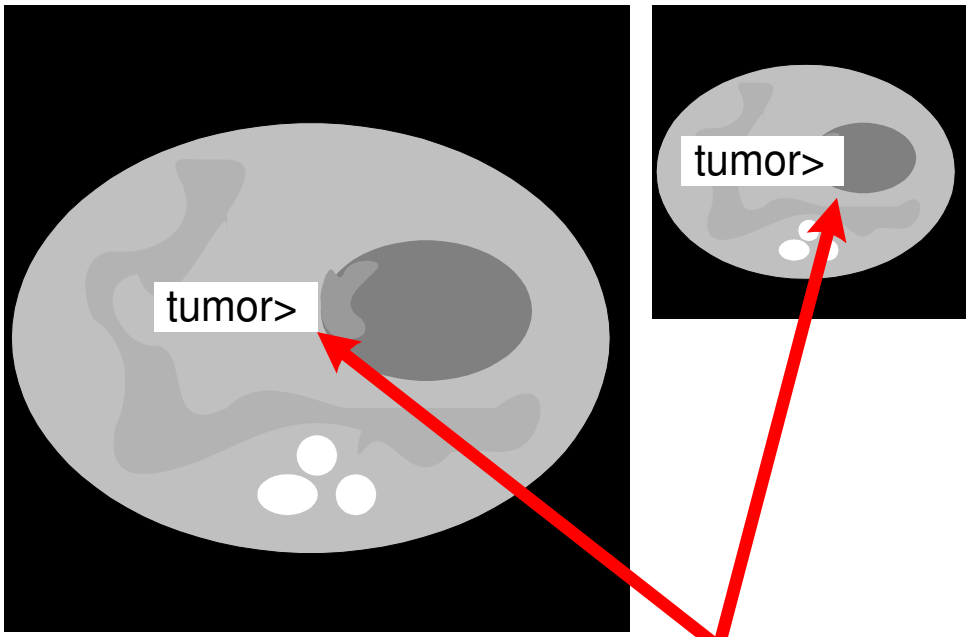


Image Quality expectation WYSIWYG



Safety problem



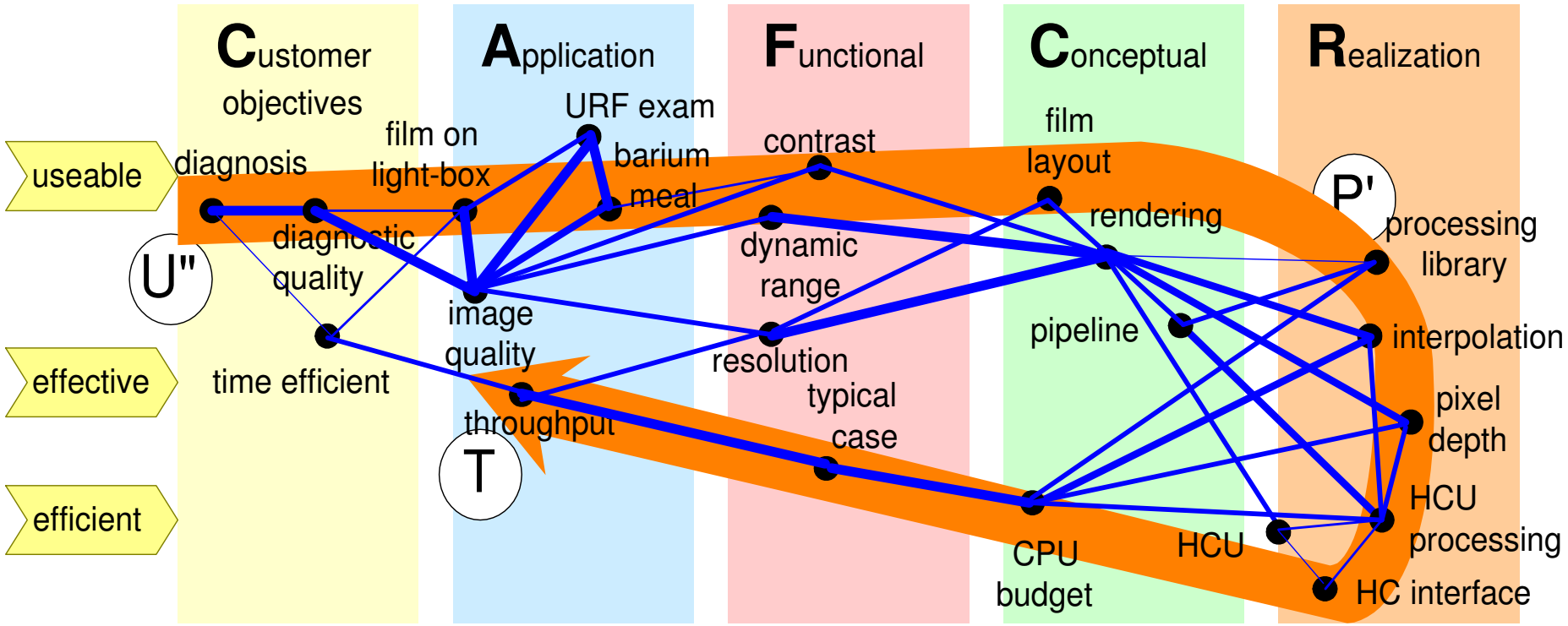


for user readability the font-size was determined "intelligently"; causing a dangerous mismatch between text and image

URF monitor output:
fixed size letters at fixed grid

EV output: scaleable fonts in graphics overlay

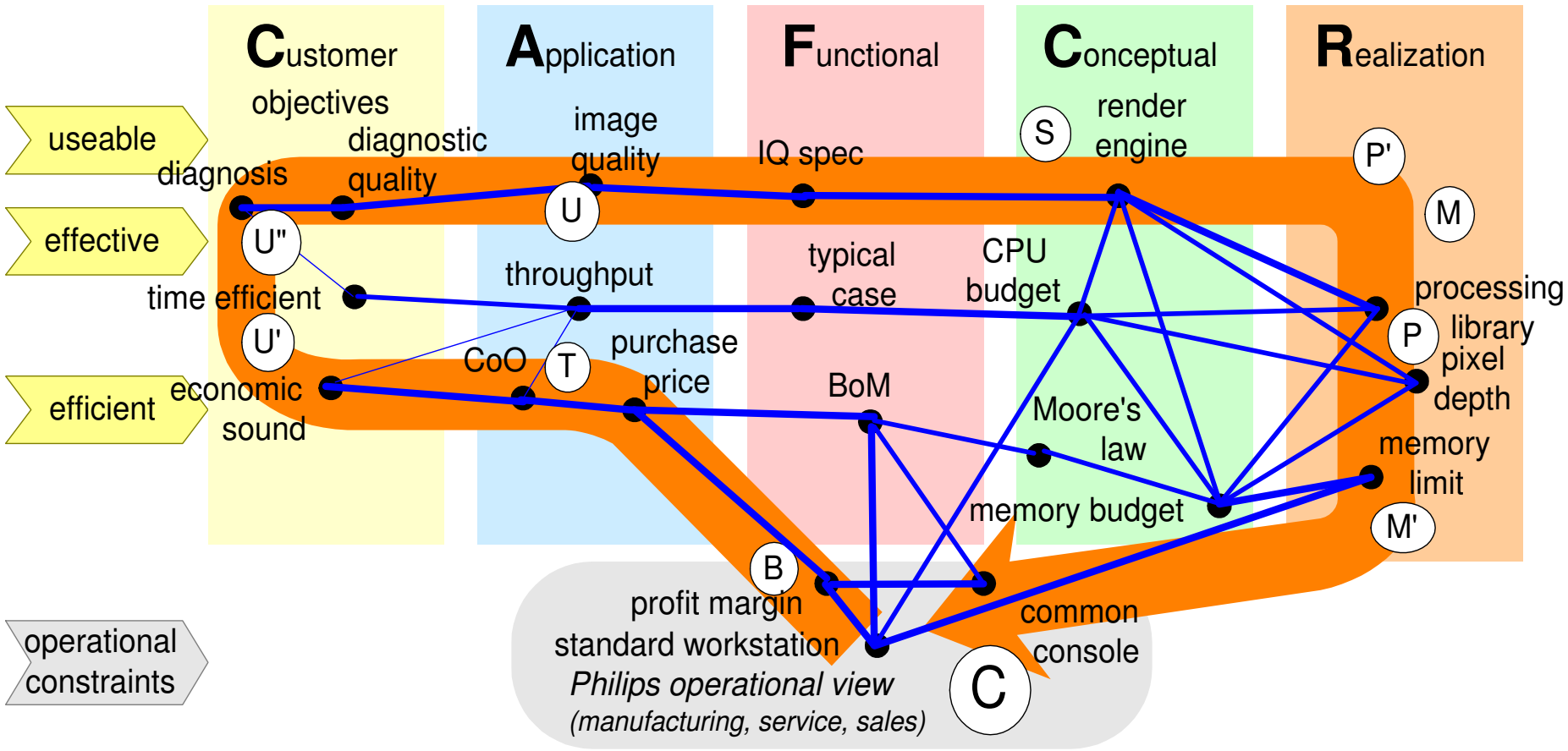
Thread of reasoning; phase 4



Philips operational view
(manufacturing, service, sales)

from extrovert diagnostic quality, via image quality, algorithms and load, to extrovert throughput

Thread of reasoning; phase 5



cost revisited in context of clinical needs and realization constraints; note: original threads are significantly simplified

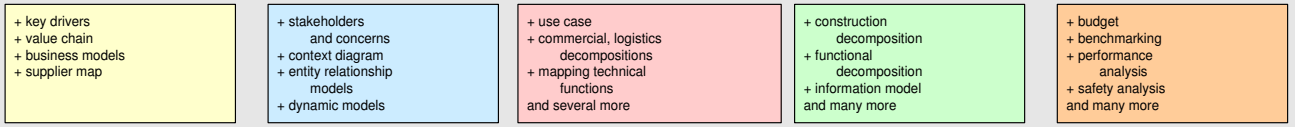
Overview of architecting method

method outline

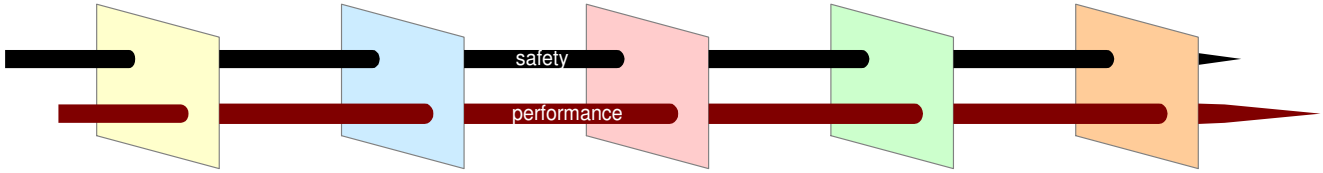
method visualization

framework

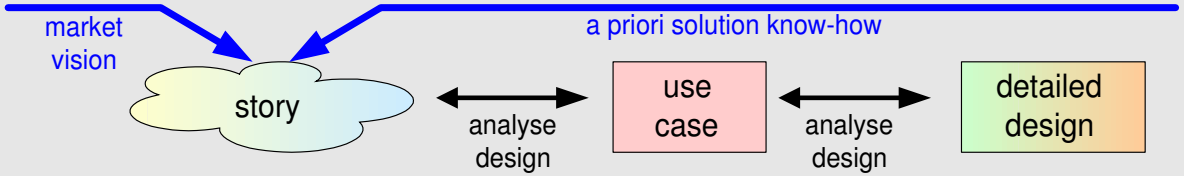
submethods



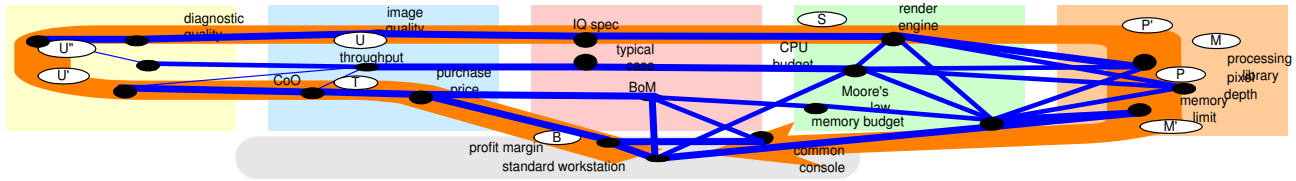
integration via qualities



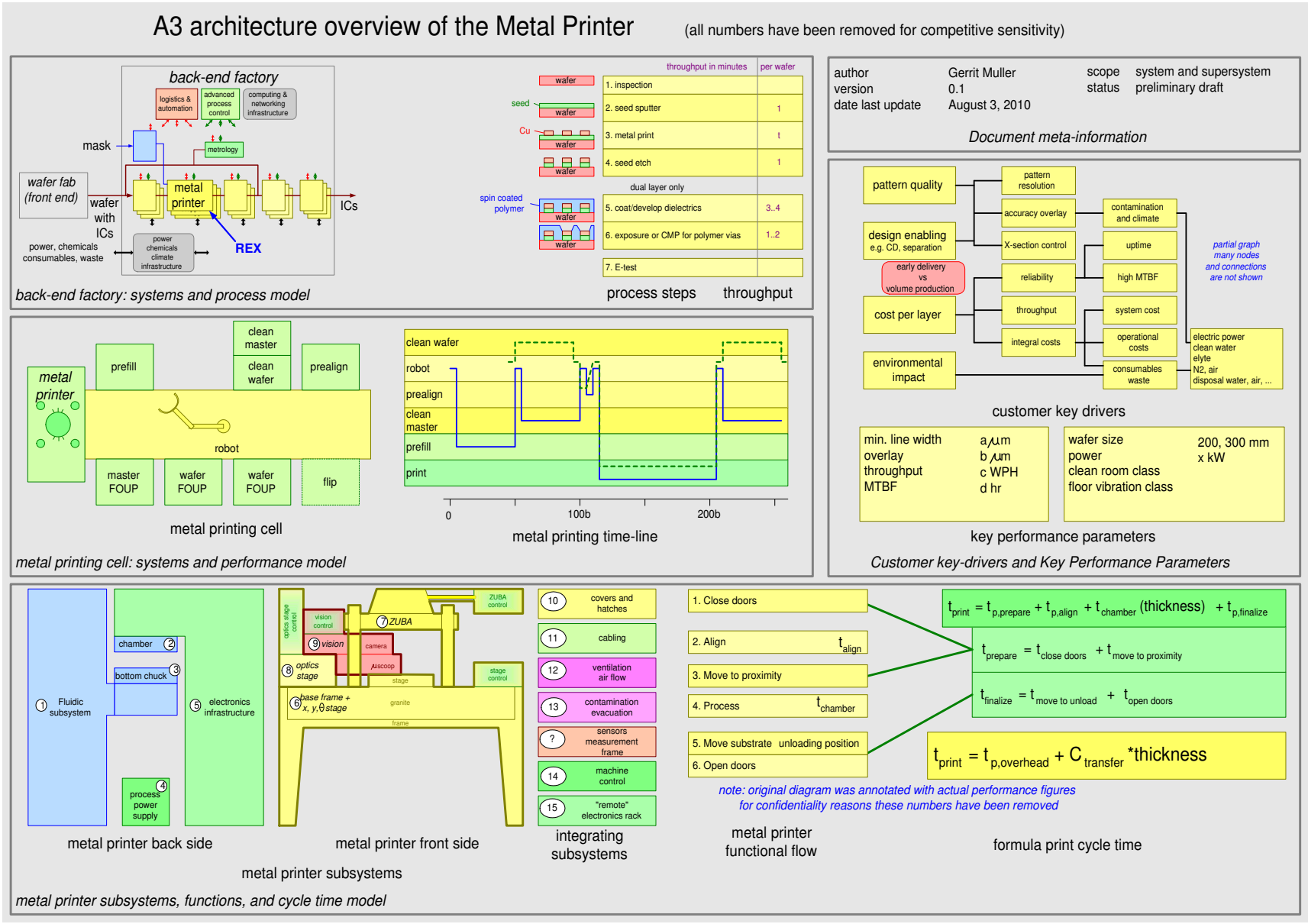
explore specific details



reasoning



Example of A3 Architecture Overview



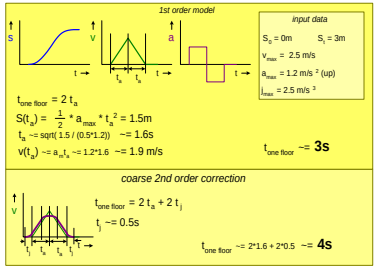
Physical Models of an Elevator

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

Abstract

An elevator is used as a simple system to model a few physical aspects. We will show simple kinematic models and we will consider energy consumption. These low level models are used to understand (physical) design considerations. Elsewhere we discuss higher level models, such as use cases and throughput, which complement these low level models.

December 21, 2011
status: planned
version: 0.1



warning

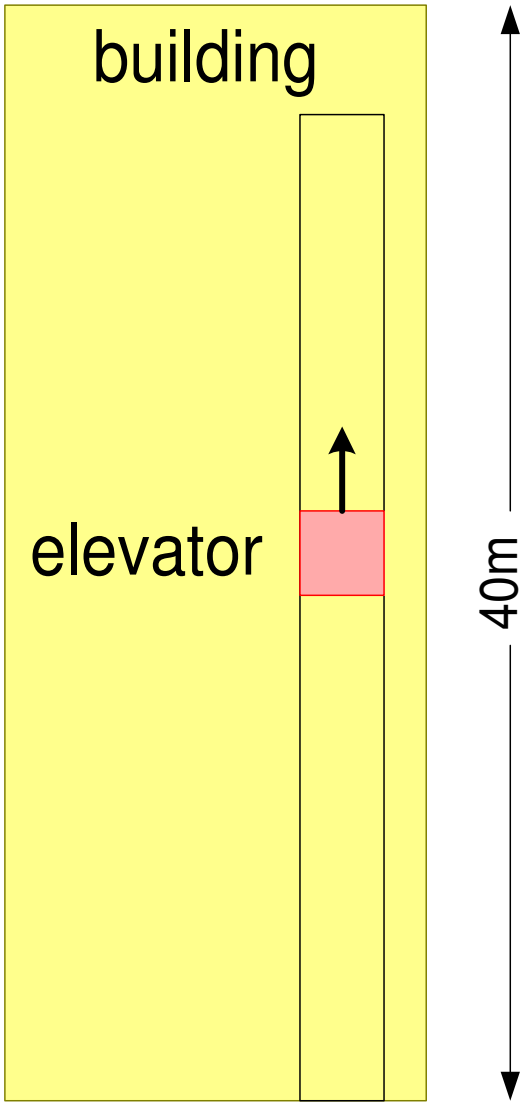
This presentation starts with a trivial problem.

Have patience!

Extensions to the trivial problem are used to illustrate many different modeling aspects.

Feedback on correctness and validity is appreciated

The Elevator in the Building



inhabitants want to reach
their destination fast and comfortable

building owner and *service operator*
have economic constraints:
space, cost, energy, ...

Elementary Kinematic Formulas

S_t = position at time t

v_t = velocity at time t

a_t = acceleration at time t

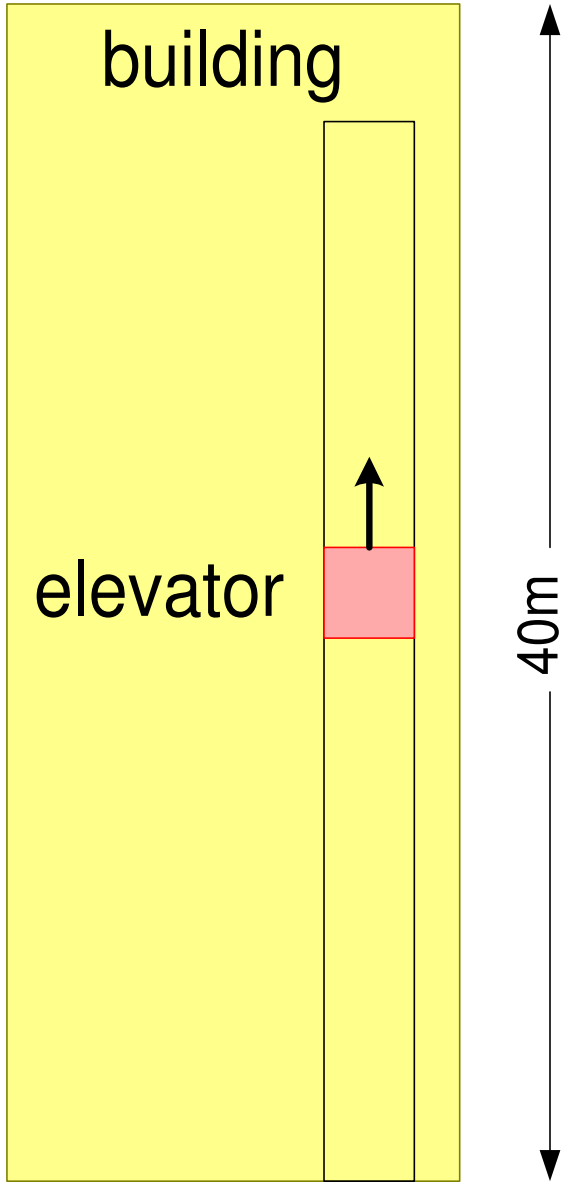
j_t = jerk at time t

$$v = \frac{dS}{dt} \qquad a = \frac{dv}{dt} \qquad j = \frac{da}{dt}$$

Position in case of uniform acceleration:

$$S_t = S_0 + v_0 t + \frac{1}{2} a_0 t^2$$

Initial Expectations



What values do you expect or prefer for these quantities? Why?

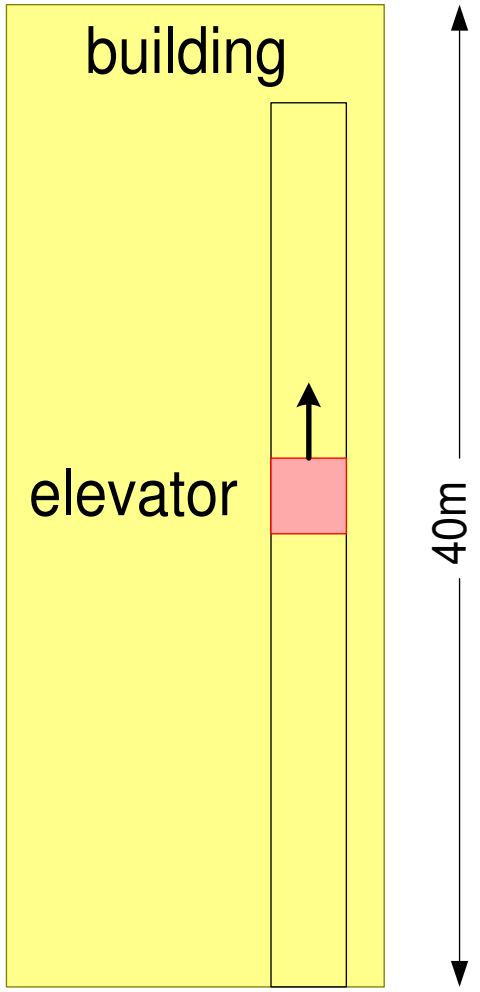
$t_{top\ floor}$ = time to reach top floor

v_{max} = maximum velocity

a_{max} = maximum acceleration

j_{max} = maximum jerk

Initial Estimates via Googling



Google "elevator" and "jerk":

$t_{top\ floor}$ ~= 16 s

v_{max} ~= 2.5 m/s

relates to motor design and energy consumption

12% of gravity; weight goes up

a_{max} ~= 1.2 m/s ^{2} (up)

j_{max} ~= 2.5 m/s ^{3} — relates to control design

humans feel changes of forces
high jerk values are uncomfortable

numbers from: http://www.sensor123.com/vm_eva625.htm
CEP Instruments Pte Ltd Singapore

Exercise Time to Reach Top Floor Kinematic

input data

$S_0 = 0m$ $S_t = 40m$

$v_{max} = 2.5\ m/s$

$a_{max} = 1.2\ m/s^2$ (up)

$j_{max} = 2.5\ m/s^3$

elementary formulas

$$v = \frac{dS}{dt} \qquad a = \frac{dv}{dt} \qquad j = \frac{da}{dt}$$

Position in case of uniform acceleration:

$$S_t = S_0 + v_0 t + \frac{1}{2} a_0 t^2$$

exercises

Make a model for $t_{\text{top floor}}$

Make 0 ^{e} order model, based on constant velocity

Make 1 ^{e} order model, based on constant acceleration

What do you conclude from these models?

Models for Time to Reach Top Floor

input data

$S_0 = 0m$ $S_t = 40m$

$v_{max} = 2.5\ m/s$

$a_{max} = 1.2\ m/s^2$ (up)

$j_{max} = 2.5\ m/s^3$

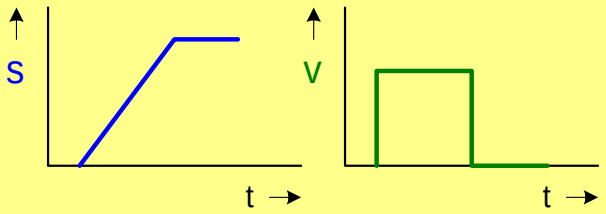
elementary formulas

$v = \frac{dS}{dt}$ $a = \frac{dv}{dt}$ $j = \frac{da}{dt}$

Position in case of uniform acceleration:

$S_t = S_0 + v_0 t + \frac{1}{2} a_0 t^2$

0th order model

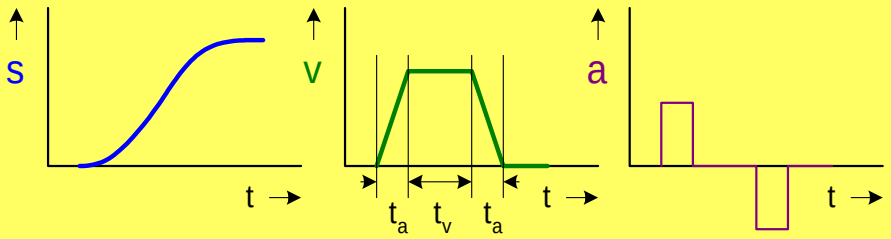


$S_{top\ floor} = v_{max} * t_{top\ floor}$

$40 = 2.5 * t_{top\ floor}$

$t_{top\ floor} = 40/2.5 =$ **16s**

1st order model



$t_{top\ floor} = t_a + t_v + t_a$

$t_a = v_{max} / a_{max}$

$S(t_a) = \frac{1}{2} * a_{max} * t_a^2$

$S_{linear} = S_{top\ floor} - 2 * S(t_a)$

$t_v = S_{linear} / v_{max}$

$t_a \sim= 2.5/1.2 \sim= 2s$

$S(t_a) \sim= 0.5 * 1.2 * 2^2$

$S(t_a) \sim= 2.4m$

$t_v \sim= (40-2*2.4)/2.5$

$t_v \sim= 14s$

$t_{top\ floor} \sim= 2 + 14 + 2$

$t_{top\ floor} \sim=$ **18s**

Conclusions Move to Top Floor

Conclusions

v_{max} dominates traveling time

The model for the large height traveling time can be simplified into:

$$t_{travel} = S_{travel}/v_{max} + (t_a + t_j)$$

Exercise Elevator Performance

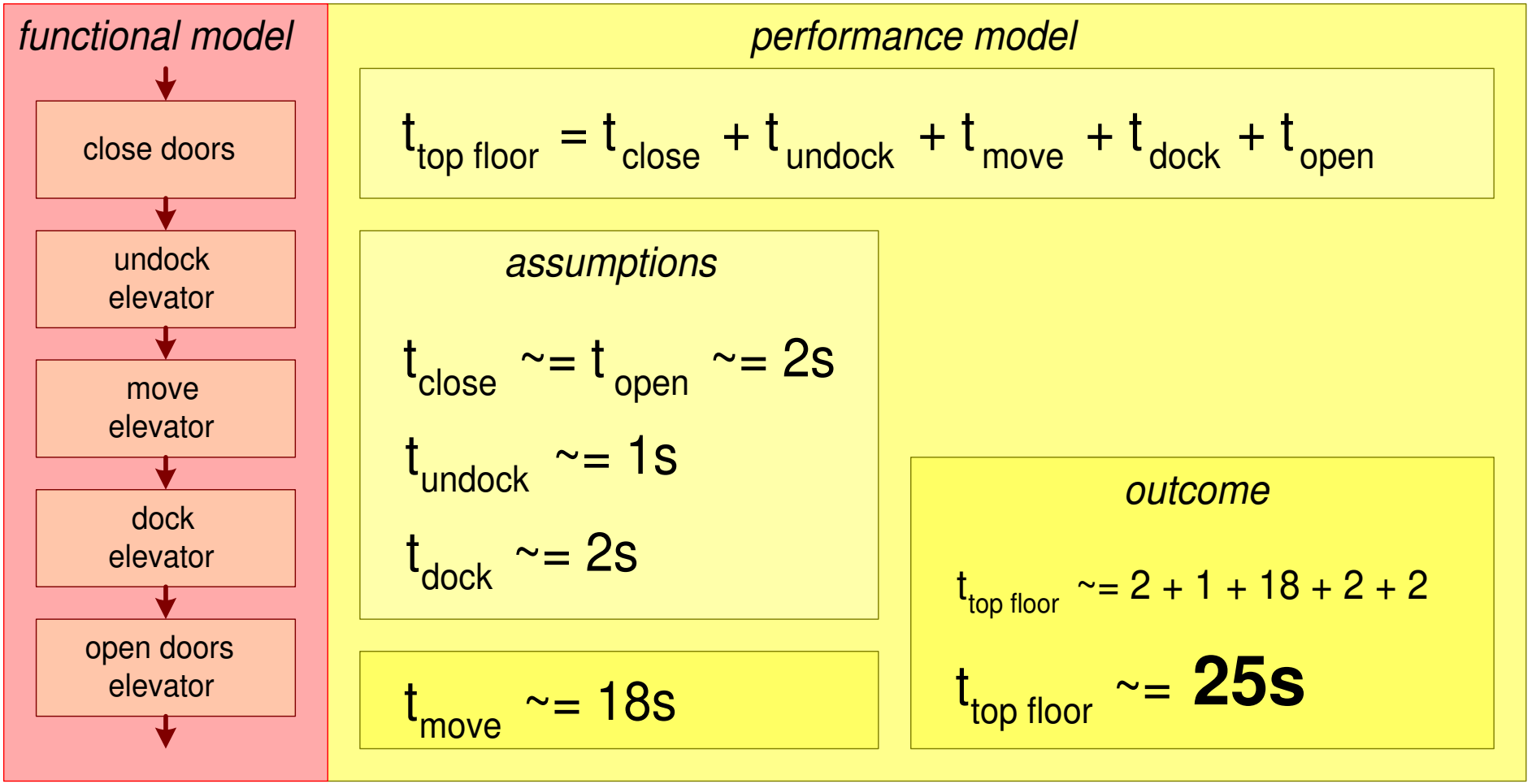
exercises

Make a model for $t_{\text{top floor}}$

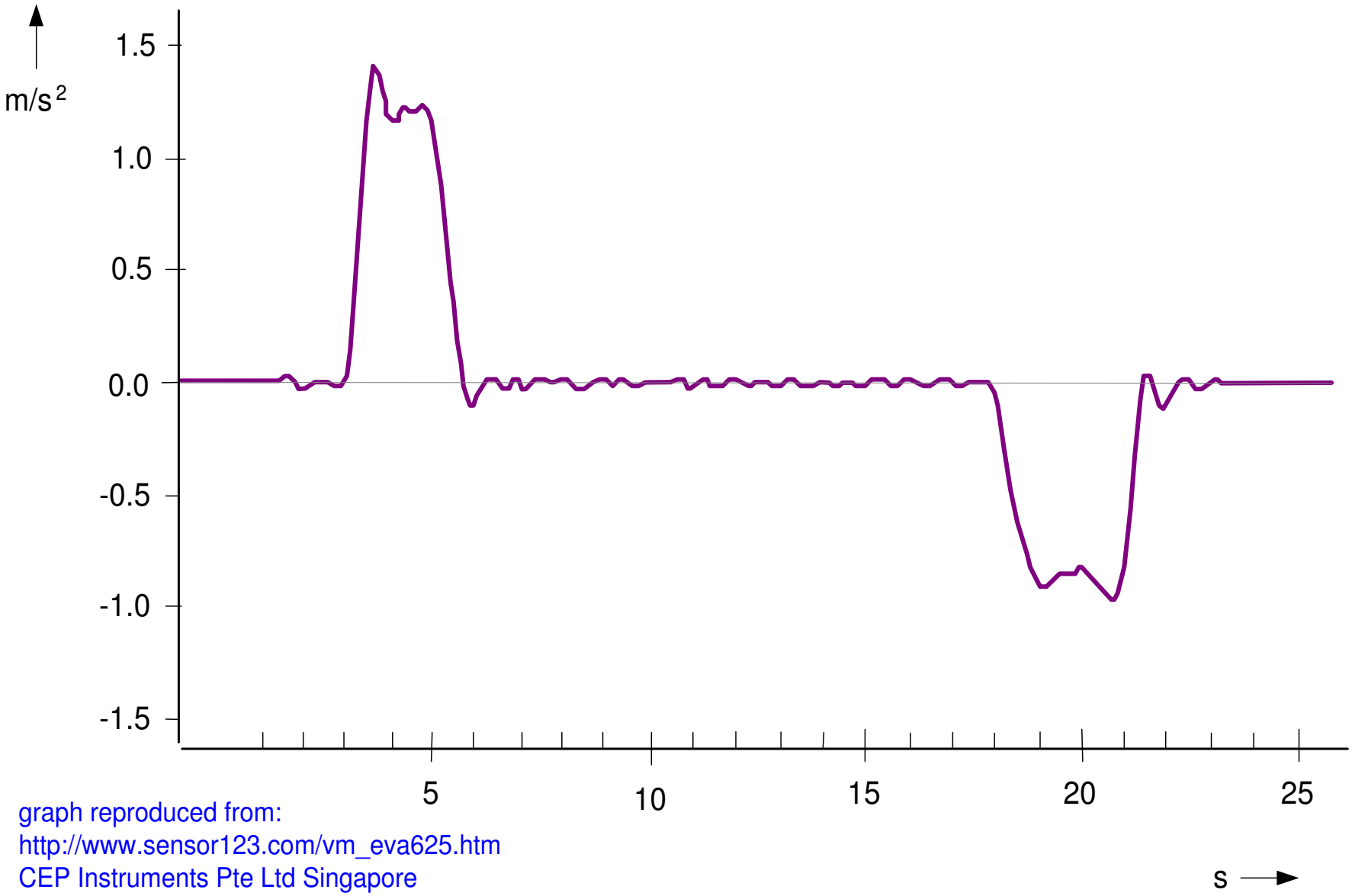
Take door opening and docking into account

What do you conclude from this model?

Elevator Performance Model



Measured Elevator Acceleration



graph reproduced from:
http://www.sensor123.com/vm_eva625.htm
CEP Instruments Pte Ltd Singapore

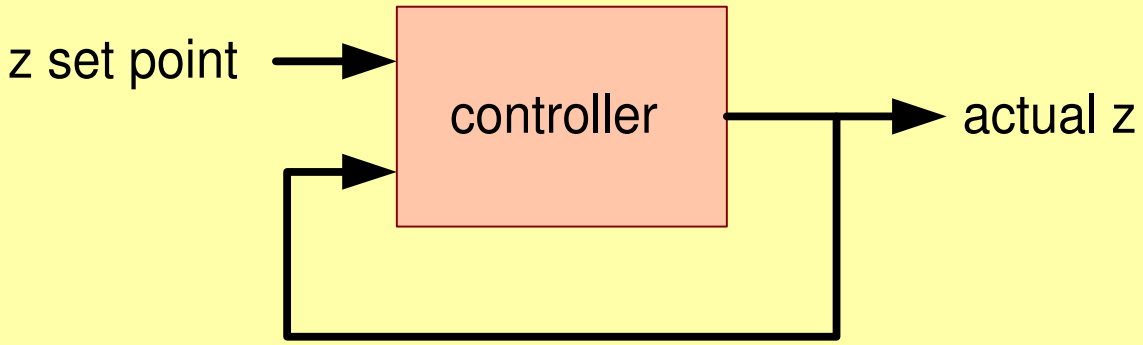
Theory versus Practice

What did we ignore or forget?

acceleration: up <> down 1.2 m/s ^{2} vs 1.0 m/s ^{2}

slack, elasticity, damping et cetera of cables, motors....

controller impact



.....

Conclusions Performance Model Top Floor

Conclusions

The time to move is dominating the traveling time.

Docking and door handling is significant part of the traveling time.

$$t_{\text{top floor}} = t_{\text{travel}} + t_{\text{elevator overhead}}$$

Exercise Elevator Performance (2)

exercises

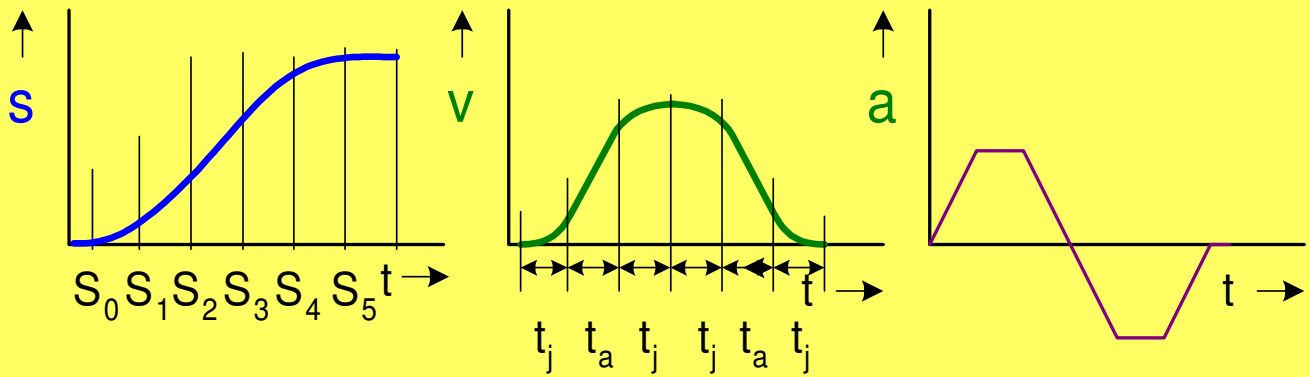
Make a model for t one floor

Take door opening and docking into account

What do you conclude from this model?

2nd Order Model Moving One Floor

2nd order model



input data

$S_0 = 0m$ $S_t = 3m$

$v_{max} = 2.5$ m/s

$a_{max} = 1.2$ m/s ^{2} (up)

$j_{max} = 2.5$ m/s ^{3}

$t_{one\ floor} = 2\,t_a + 4\,t_j$

$t_j = a_{max} / j_{max}$

$S_1 = 1/6 * j_{max}\, t_j^3$

$v_1 = 0.5\ j_{max}\, t_j^2$

$S_2 = S_1 + v_1 t_a + 0.5\, a_{max}\, t_a^2$

$v_2 = v_1 + a_{max}\, t_a$

$S_3 = S_2 + v_2 t_j + 0.5\, a_{max}\, t_j^2 - 1/6\, j_{max}\, t_j^3$

$S_3 = 0.5\, S_t$

t_j ~= 1.2/2.5 ~= 0.5s

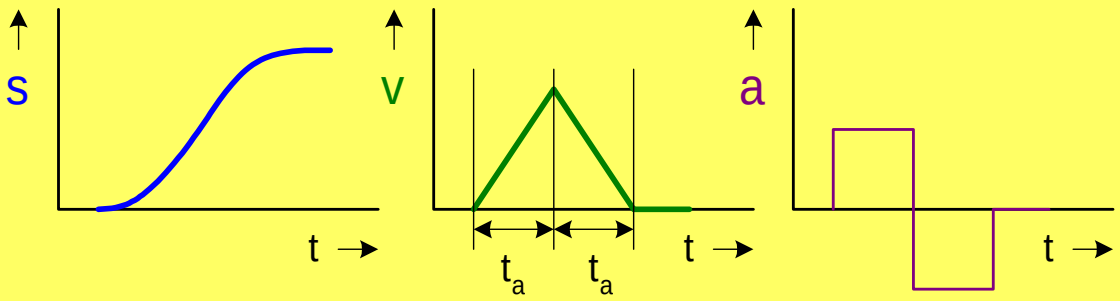
S_1 ~= 1/6 * 2.5 * 0.5^3 ~= 0.05m

v_1 ~= 0.5 * 2.5 * 0.5^2 ~= 0.3m/s

et cetera

1st Order Model Moving One Floor

1st order model



input data

S_0 = 0m S_t = 3m

v_{max} = 2.5 m/s

a_{max} = 1.2 m/s ^{2} (up)

j_{max} = 2.5 m/s ^{3}

$t_{one\ floor} = 2\ t_a$

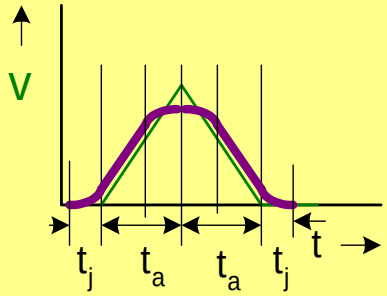
$S(t_a) = \frac{1}{2} * a_{max} * t_a^2 = 1.5m$

t_a ~= sqrt(1.5 / (0.5*1.2)) ~= 1.6s

$v(t_a)$ ~= $a_m t_a$ ~= 1.2*1.6 ~= 1.9 m/s

$t_{one\ floor}$ ~= **3s**

coarse 2nd order correction



$t_{one\ floor} = 2\ t_a + 2\ t_j$

t_j ~= 0.5s

$t_{one\ floor}$ ~= 2*1.6 + 2*0.5 ~= **4s**

Conclusions Move One Floor

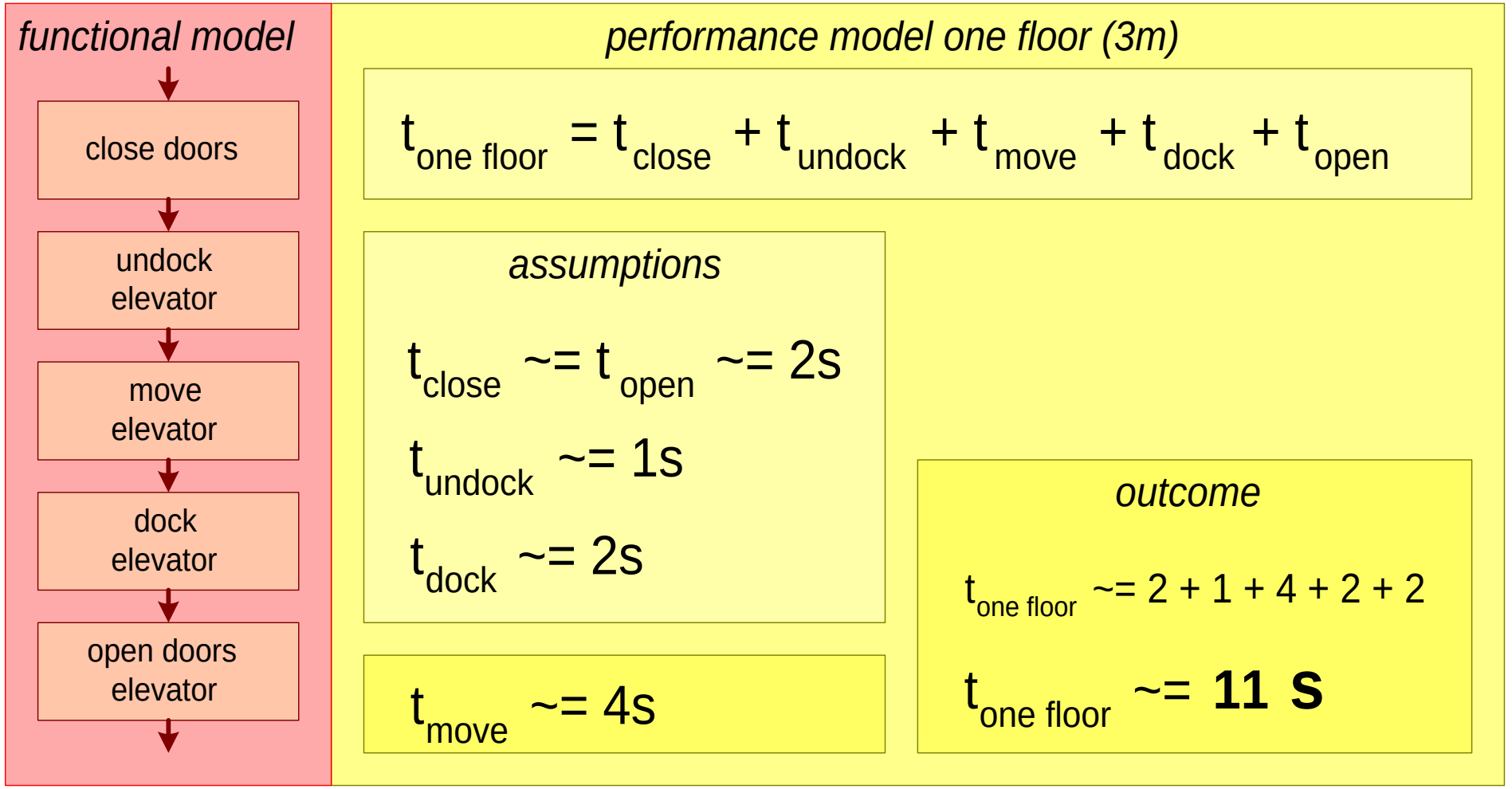
Conclusions

a_{max} dominates travel time

The model for small height traveling time can be simplified into:

$$t_{travel} = 2\ sqrt(S_{travel}/a_{max}) + t_j$$

Elevator Performance Model



Conclusions Performance Model One Floor

Conclusions

Overhead of docking and opening and closing doors is dominating traveling time.

Fast docking and fast door handling has significant impact on traveling time.

$$t_{\text{one floor}} = t_{\text{travel}} + t_{\text{elevator overhead}}$$

Exercise Time Line

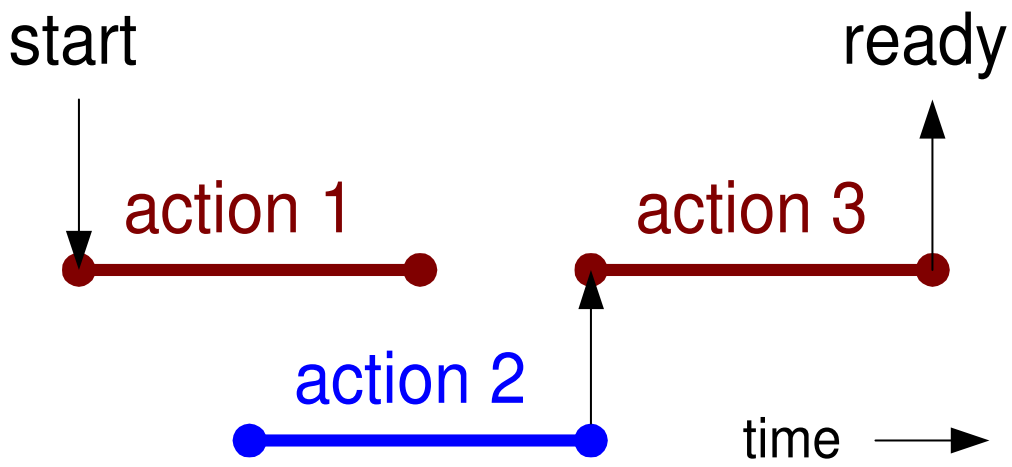
Exercise

Make a time line of people using the elevator.

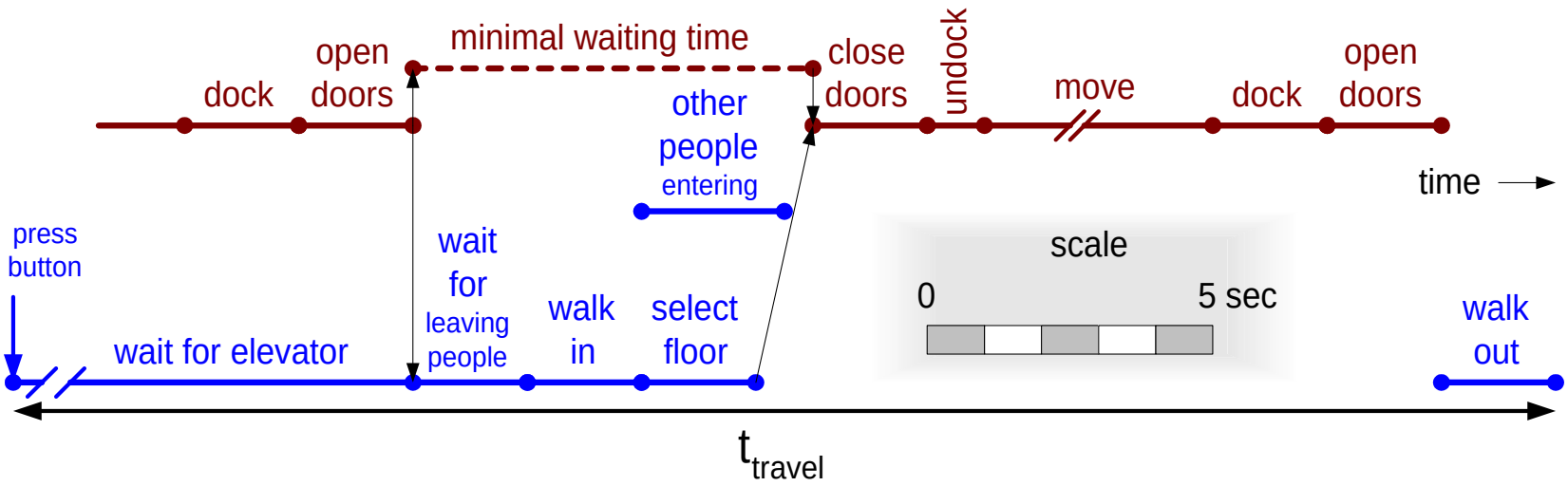
Estimate the time needed to travel to the top floor.

Estimate the time needed to travel one floor.

What do you conclude?



Time Line; Humans Using the Elevator



assumptions human dependent data

$t_{wait\ for\ elevator}$ = [0..2 minutes] depends heavily on use

$t_{wait\ for\ leaving\ people}$ = [0..20 seconds] idem

$t_{walk\ in}$ ~= 2 s

$t_{select\ floor}$ ~= 2 s

assumptions additional elevator data

$t_{minimal\ waiting\ time}$ ~= 8s

$t_{top\ floor}$ ~= 25s

$t_{one\ floor}$ ~= 11s

outcome

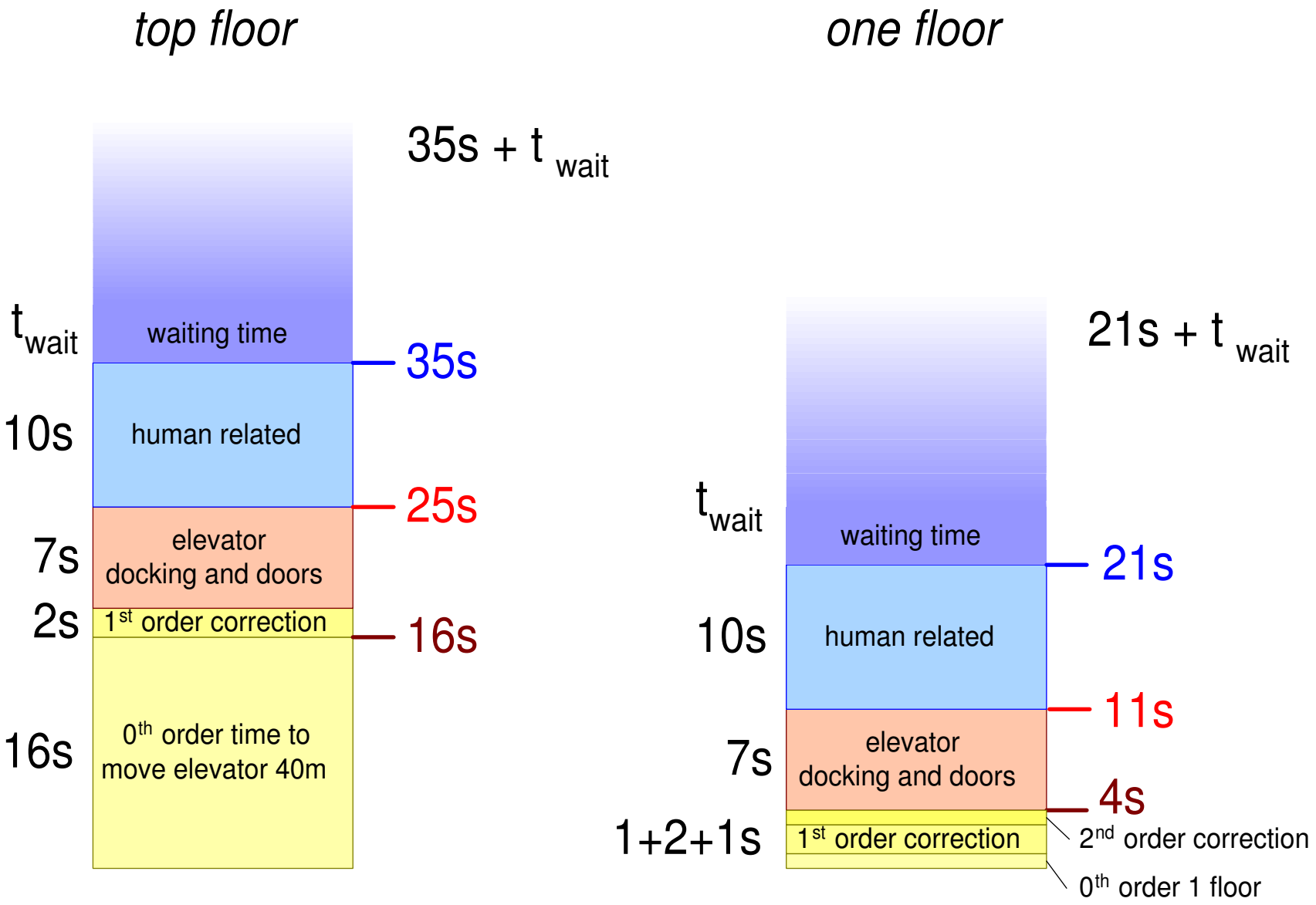
$t_{one\ floor}$ ~= 8 +2 + 11 + t_{wait}

~= **21 s** + t_{wait}

$t_{top\ floor}$ ~= 8 +2 + 25 + t_{wait}

~= **35 s** + t_{wait}

Overview of Results for One Elevator



Conclusions Time Line

Conclusions

The human related activities have significant impact on the end-to-end time.

The waiting times have significant impact on the end-to-end time and may vary quite a lot.

$$t_{\text{end-to-end}} = t_{\text{human activities}} + t_{\text{wait}} + t_{\text{elevator travel}}$$

Exercise Energy and Power

Exercise

Estimate the energy consumption and the average and peak power needed to travel to the top floor.

What do you conclude?

Energy and Power Model

input data

$S_0 = 0m$ $S_t = 40m$

$v_{max} = 2.5$ m/s $m_{elevator} = 1000$ Kg (incl counter weight)

$a_{max} = 1.2$ m/s^2 (up) $m_{passenger} = 100$ Kg

$j_{max} = 2.5$ m/s^3 1 passenger going up

$g = 10$ m/s^2

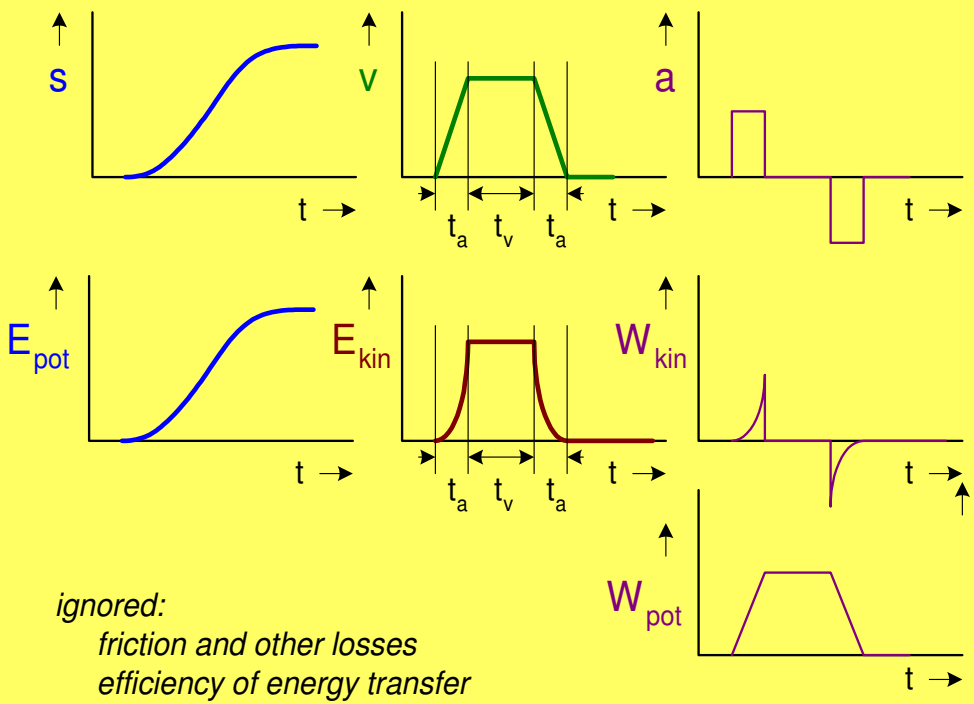
elementary formulas

$E_{kin} = 1/2\ mv^2$

$E_{pot} = mgh$

$W = \frac{dE}{dt}$

1st order model



ignored:
friction and other losses
efficiency of energy transfer

$E_{kin\ max} = 1/2\ m\ v_{max}^2$

$\sim= 0.5 * 1100 * 2.5^2$

$\sim=$ **3.4 kJ**

$W_{kin\ max} = m\ v_{max}\ a_{max}$

$\sim= 1100 * 2.5 * 1.2$

$\sim=$ **3.3 kW**

$E_{pot} = mgh$

$\sim= 100 * 10 * 40$

$\sim=$ **40 kJ**

$W_{pot\ max} \sim= E_{pot}/t_v$

$\sim= 40/16$

$\sim=$ **2.5 kW**

Energy and Power Conclusions

Conclusions

E_{pot} dominates energy balance

W_{pot} is dominated by v_{max}

W_{kin} causes peaks in power consumption and absorption

W_{kin} is dominated by v_{max} and a_{max}

$E_{kin\ max} = 1/2\ m\ v_{max}^2$

$\sim= 0.5 * 1100 * 2.5^2$

$\sim=$ **3.4 kJ**

$W_{kin\ max} = m\ v_{max}\ a_{max}$

$\sim= 1100 * 2.5 * 1.2$

$\sim=$ **3.3 kW**

$E_{pot} = mgh$

$\sim= 100 * 10 * 40$

$\sim=$ **40 kJ**

$W_{pot\ max} \sim= E_{pot}/t_v$

$\sim= 40/16$

$\sim=$ **2.5 kW**

Exercise Qualities and Design Considerations

Exercise

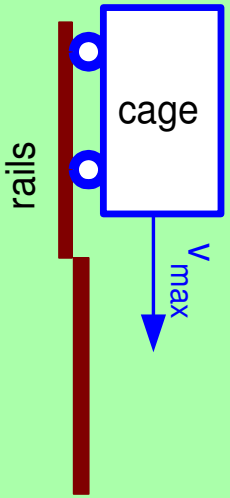
What other qualities and design considerations relate to the kinematic models?

Conclusions Qualities and Design Considerations

Examples of other qualities and design considerations

safety	v_{max}
acoustic noise	v_{max}, a_{max}, j_{max}
mechanical vibrations	v_{max}, a_{max}, j_{max}
air flow	?
operating life, maintenance	duty cycle, ?
...	

obstacles cause vibrations



Other Domains

applicability in other domains

kinematic modeling can be applied in a wide range of domains:

- transportation systems (trains, busses, cars, containers, ...)
- wafer stepper stages
- health care equipment patient handling
- material handling (printers, inserters, ...)
- MRI scanners gradient generation
- ...

Exercise Multiple Users

Exercise

Assume that a group of people enters the elevator at the ground floor. On every floor one person leaves the elevator.

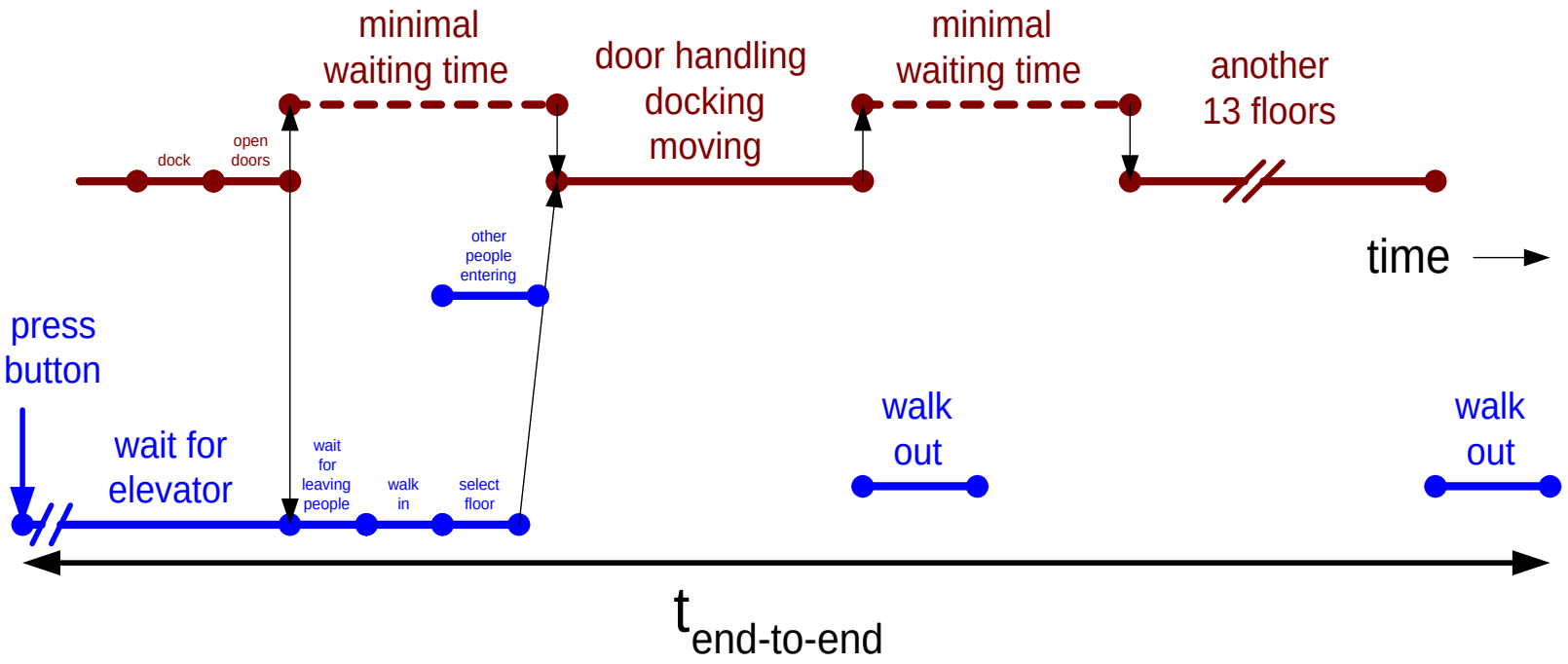
What is the end-to-end time for someone traveling to the top floor?

What is the desired end-to-end time?

What are potential solutions to achieve this?

What are the main parameters of the design space?

Multiple Users Model



elevator data

$t_{min\ wait}$ ~= 8s

$t_{one\ floor}$ ~= 11s

$t_{walk\ out}$ = 2s

n_{floors} = 40 div 3 +1 = 14

outcome

$t_{end\text{-}to\text{-}end}$ ~= 14 ($t_{min\ wait}$ + $t_{one\ floor}$) + $t_{walk\ out}$ + t_{wait}

~= 14 * (8 + 11) + 2 + t_{wait}

~= **268 s** + t_{wait}

$t_{non\text{-}stop}$ ~= 35 s + t_{wait}

Multiple Users Desired Performance

Considerations

desired time to travel to top floor ~< 1 minute

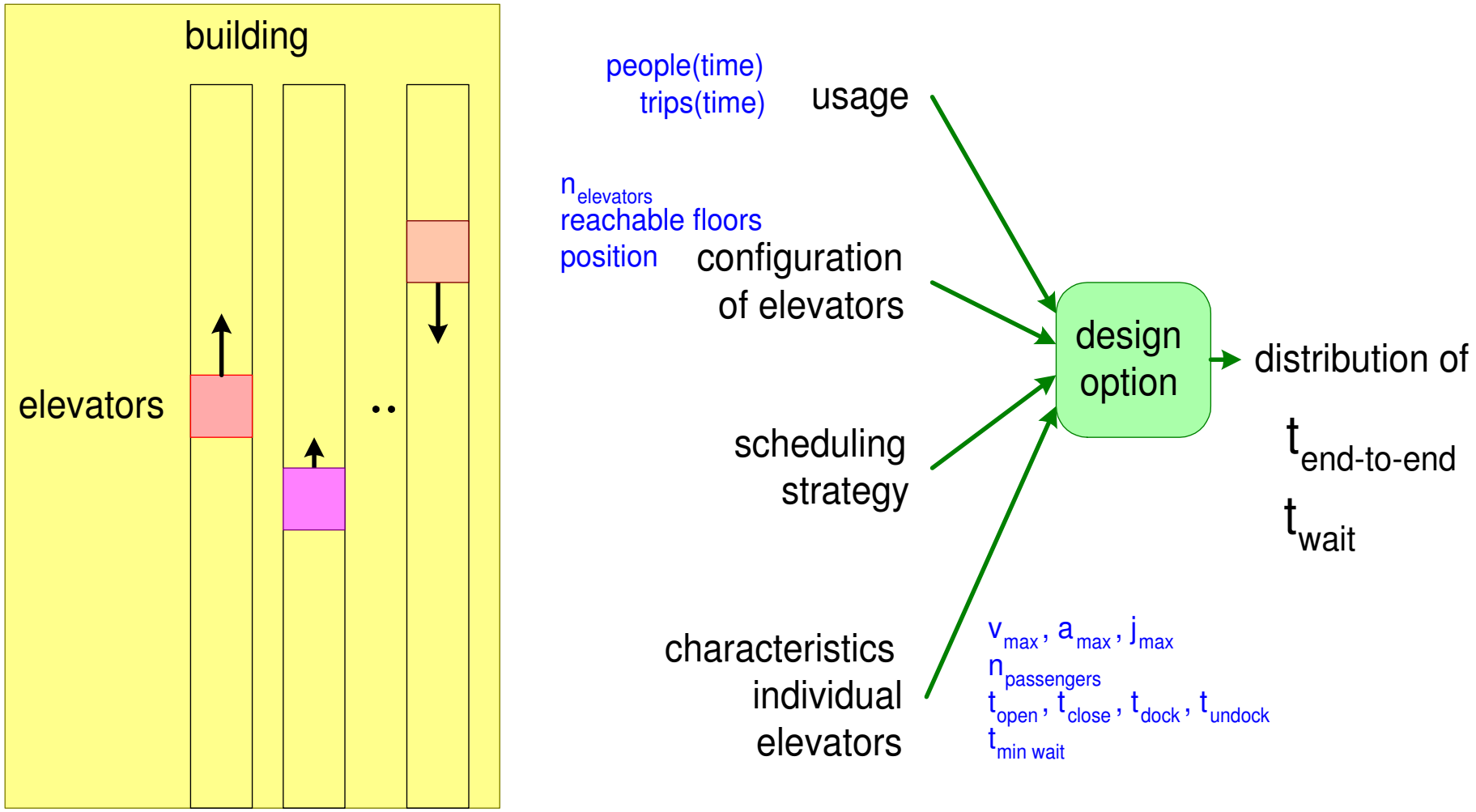
note that $t_{\text{wait next}} = t_{\text{travel up}} + t_{\text{travel down}}$

if someone just misses the elevator then the waiting time is

$$t_{\text{end-to-end}} = \underbrace{268}_{\text{missed trip}} + \underbrace{35}_{\text{return down}} + \underbrace{268}_{\text{trip up}} = 571\text{s} \sim= 10 \text{ minutes!}$$

desired waiting time ~< 1 minute

Design of Elevators System



Design of a system with multiple elevator requires a different kind of models: oriented towards logistics

Exceptional Cases

Exceptional Cases

non-functioning elevator

maintenance, cleaning of elevator

elevator used by people moving household

rush hour

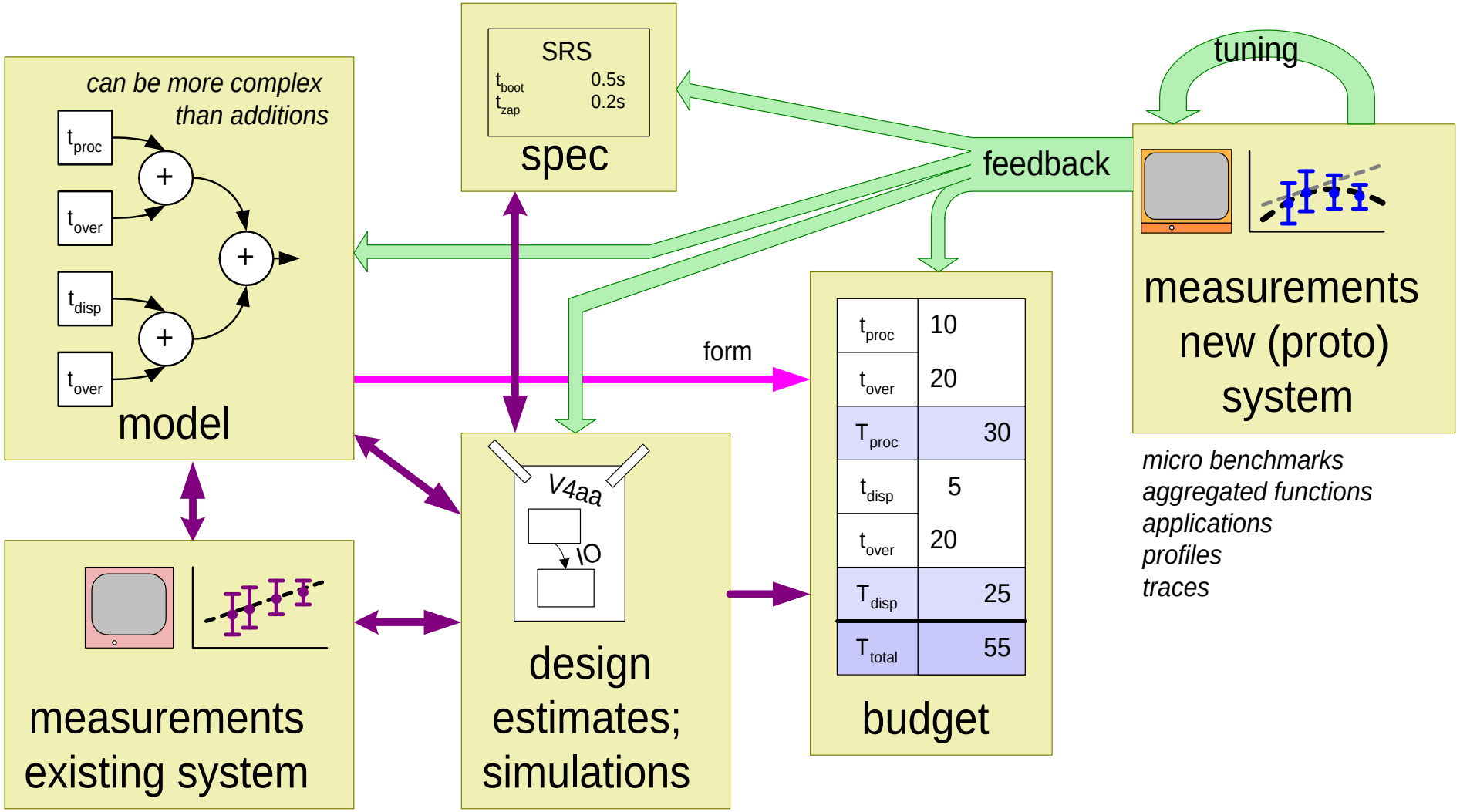
special events (e.g. party, new years eve)

special floors (e.g. restaurant)

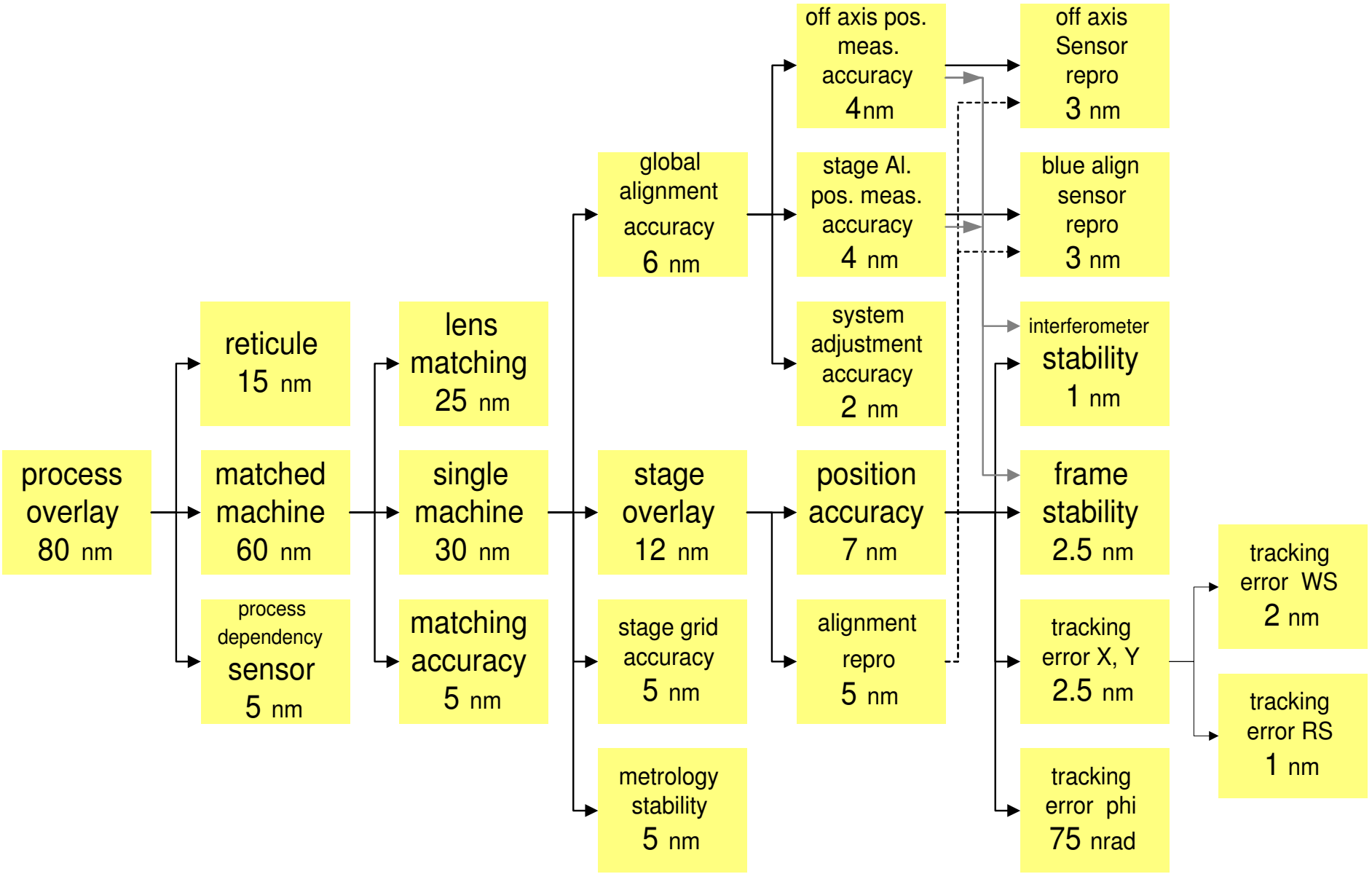
many elderly or handicapped people

playing children

Visualization of Budget Based Design Flow



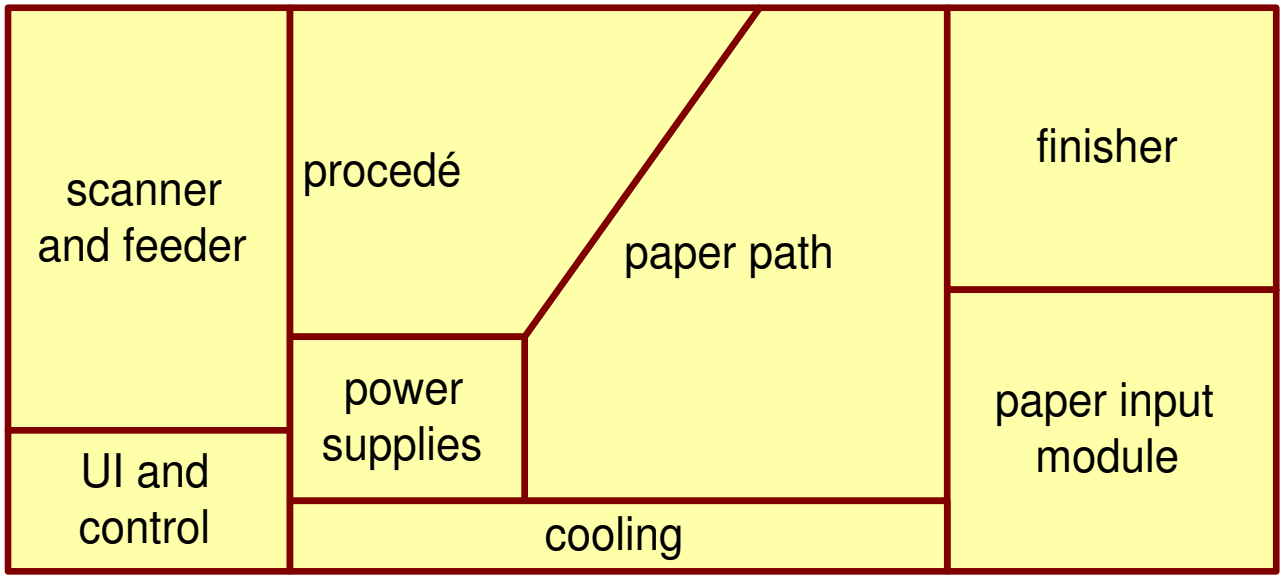
Budgets Applied on Waferstepper Overlay



Budgets Applied on Medical Workstation Memory Use

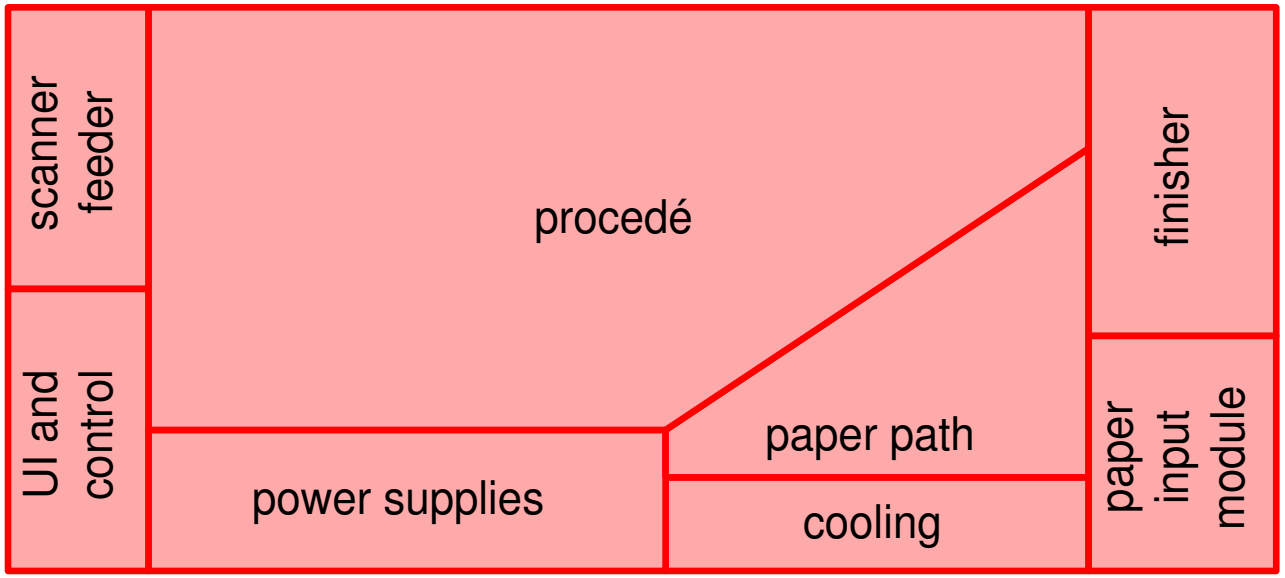
memory budget in Mbytes	code	obj data	bulk data	total
shared code	11.0			11.0
User Interface process	0.3	3.0	12.0	15.3
database server	0.3	3.2	3.0	6.5
print server	0.3	1.2	9.0	10.5
optical storage server	0.3	2.0	1.0	3.3
communication server	0.3	2.0	4.0	6.3
UNIX commands	0.3	0.2	0	0.5
compute server	0.3	0.5	6.0	6.8
system monitor	0.3	0.5	0	0.8
application SW total	13.4	12.6	35.0	61.0
UNIX Solaris 2.x				10.0
file cache				3.0
total				74.0

Power Budget Visualization for Document Handler



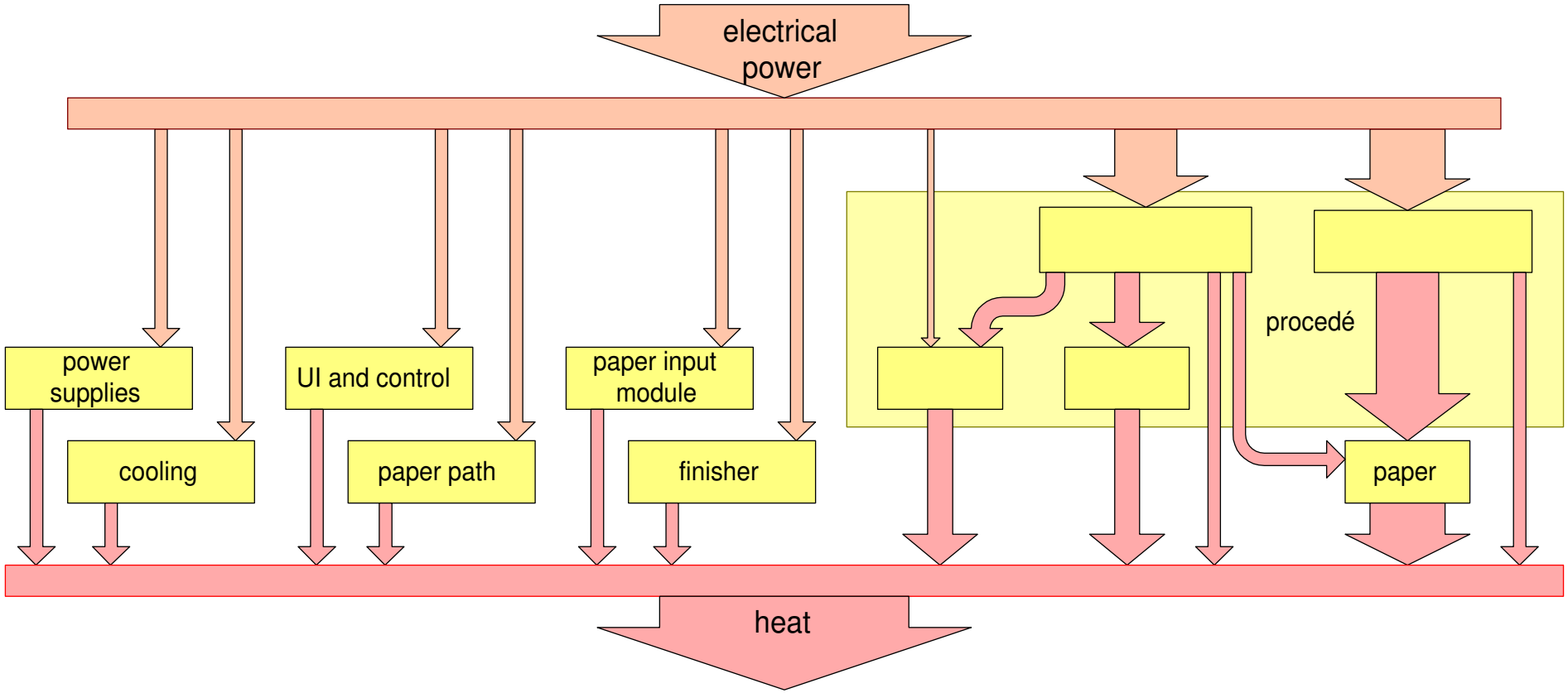
legend

physical layout

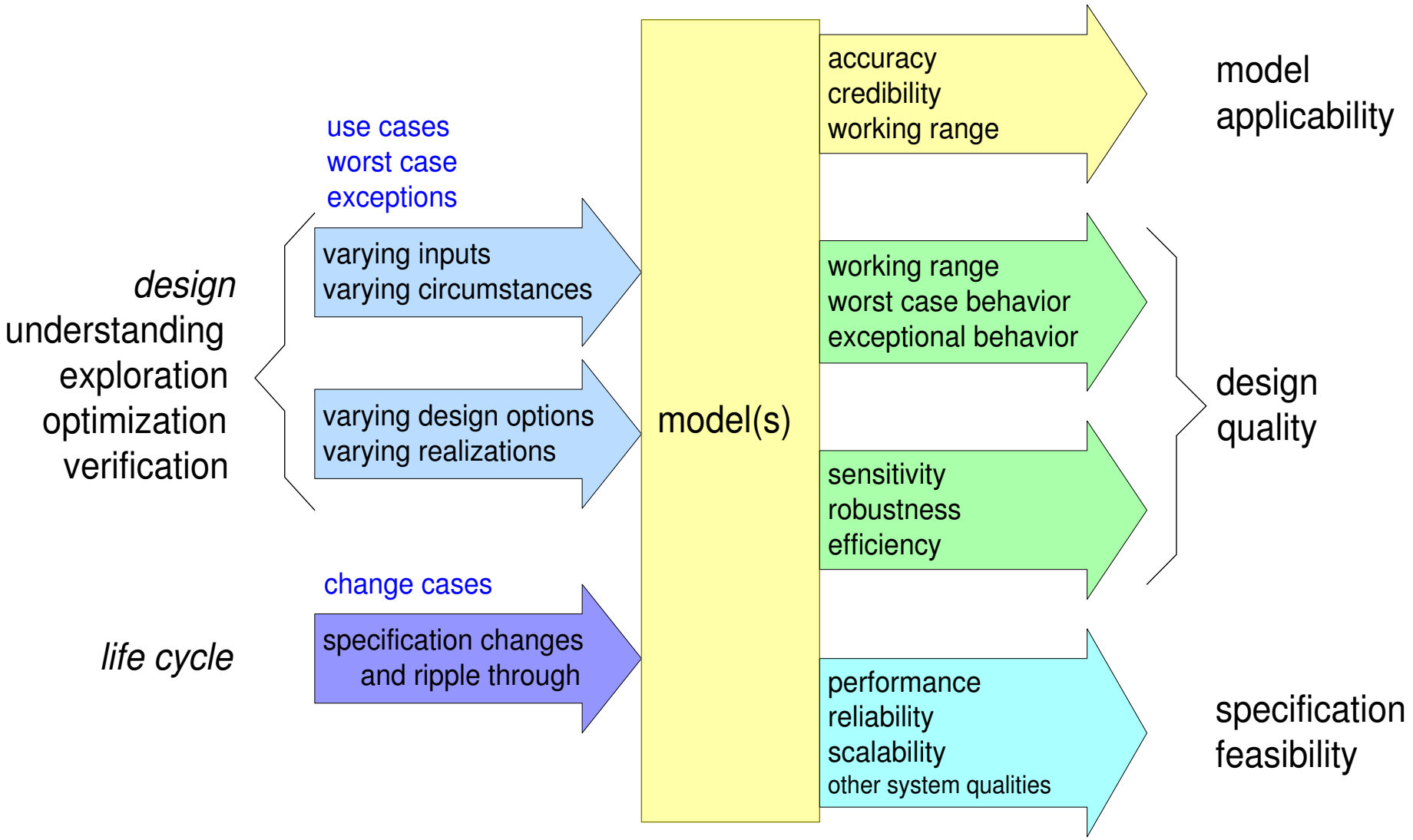


size proportional to power

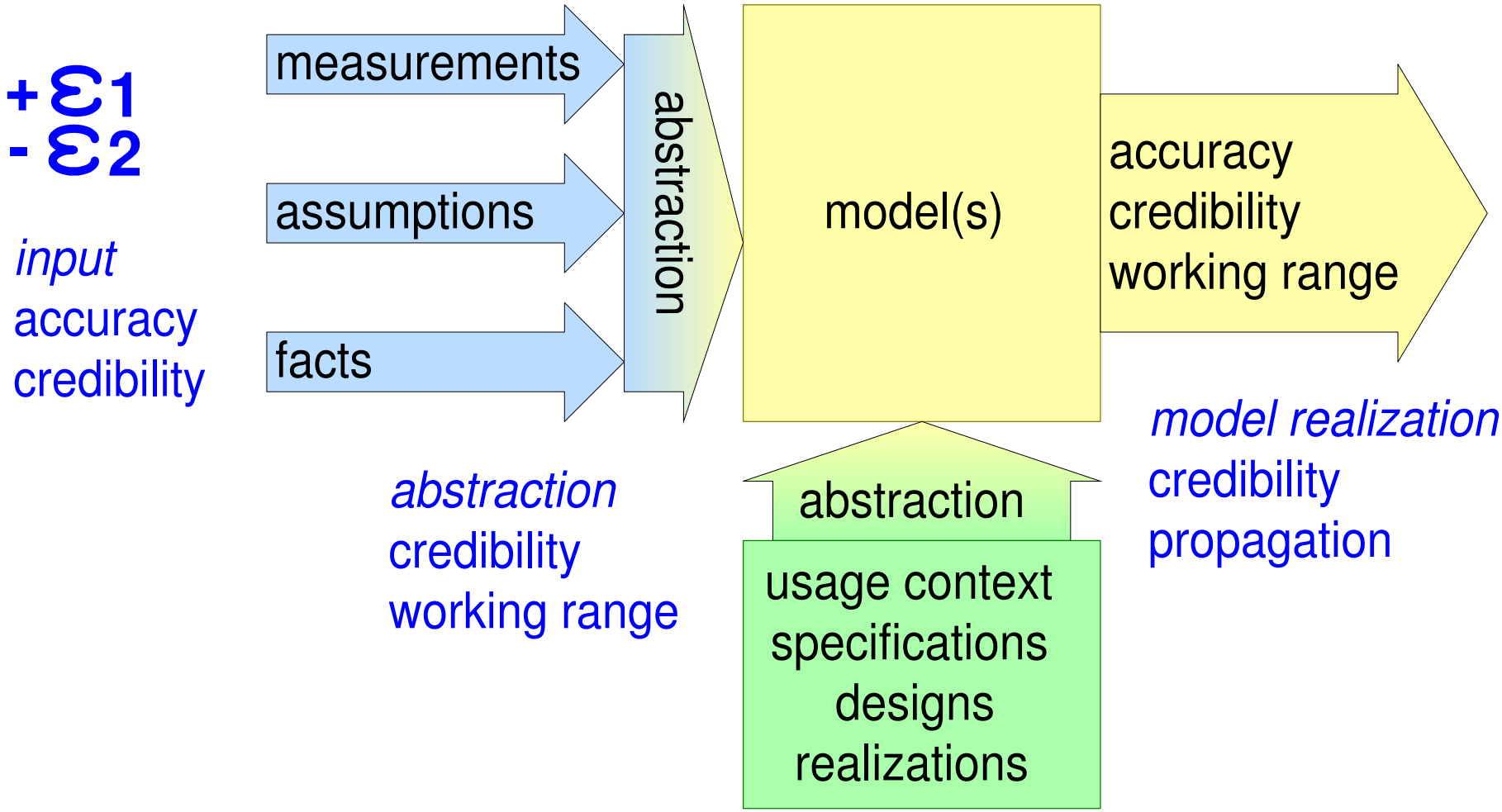
Alternative Power Visualization



What Comes out of a Model



Applicability of the Model



T-shape Presentation