

Module Reasoning: Linking Business to Technology

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

Abstract

This module addresses *Threads of Reasoning* as a means to connect business and operational needs to design and technology choices.

The complete course MA 611TM is owned by Embedded Systems Institute. To teach this course a license from Embedded Systems Institute is required. This material is preliminary course material. The final material and course information can be found at: www.esi.nl/cursus.

July 1, 2011
status: preliminary
draft
version: 0.2

Module Content

goal of this module

Be able to relate *Customer* and *Operational* objectives to design and technology choices.

Be able to provide rationale for design decisions.

content of this module

Key driver method and recommendations

Threads of reasoning approach

Example in Health Care domain

exercise

Key driver graph

Key Drivers How To

by *Gerrit Muller* Buskerud University College

e-mail: `gerrit.muller@embeddedsystems.nl`

`www.gaudisite.nl`

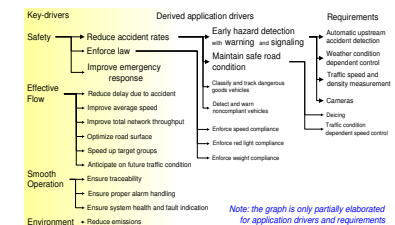
Abstract

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

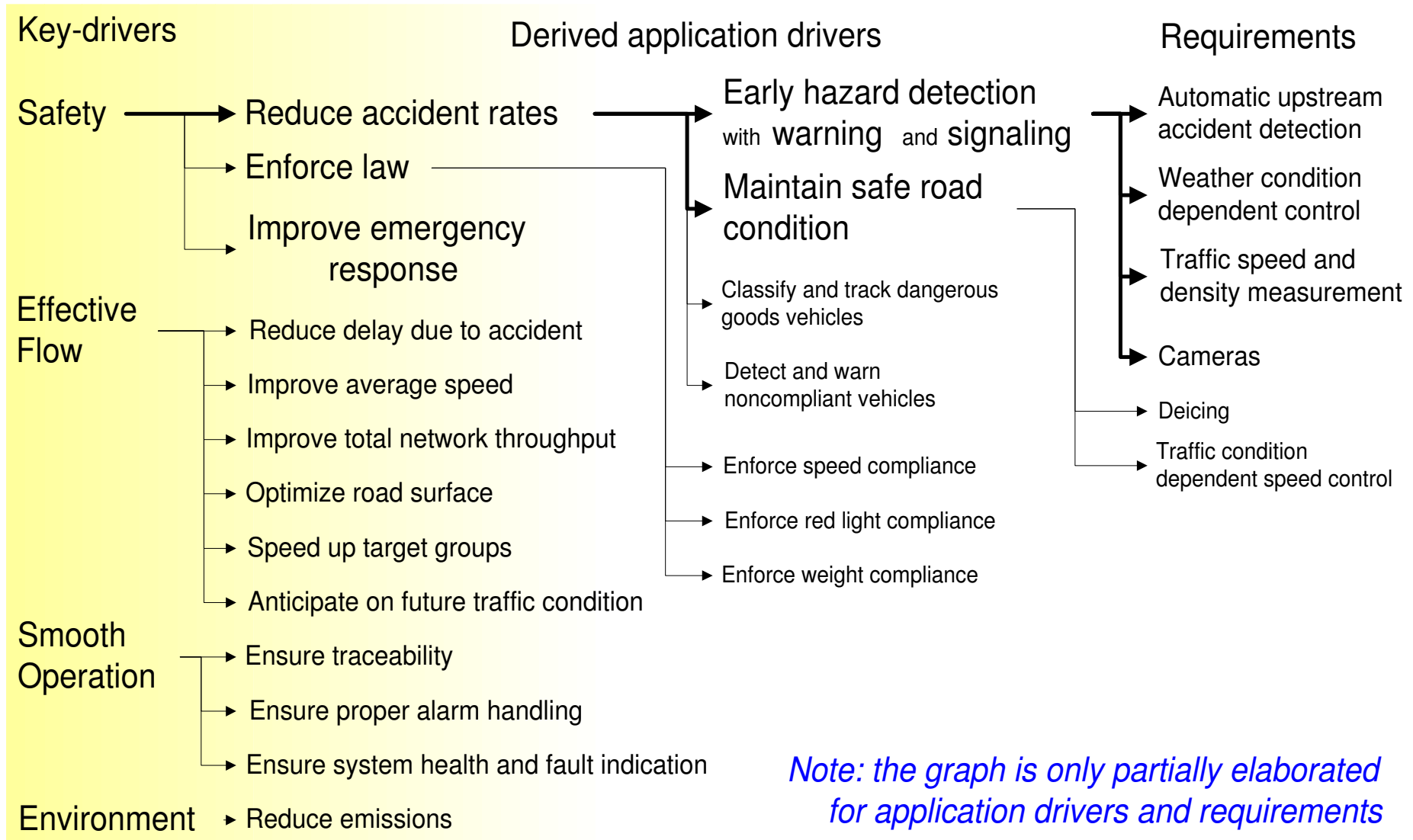
Distribution

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.

July 1, 2011
status: draft
version: 0.2



Example Motorway Management Analysis



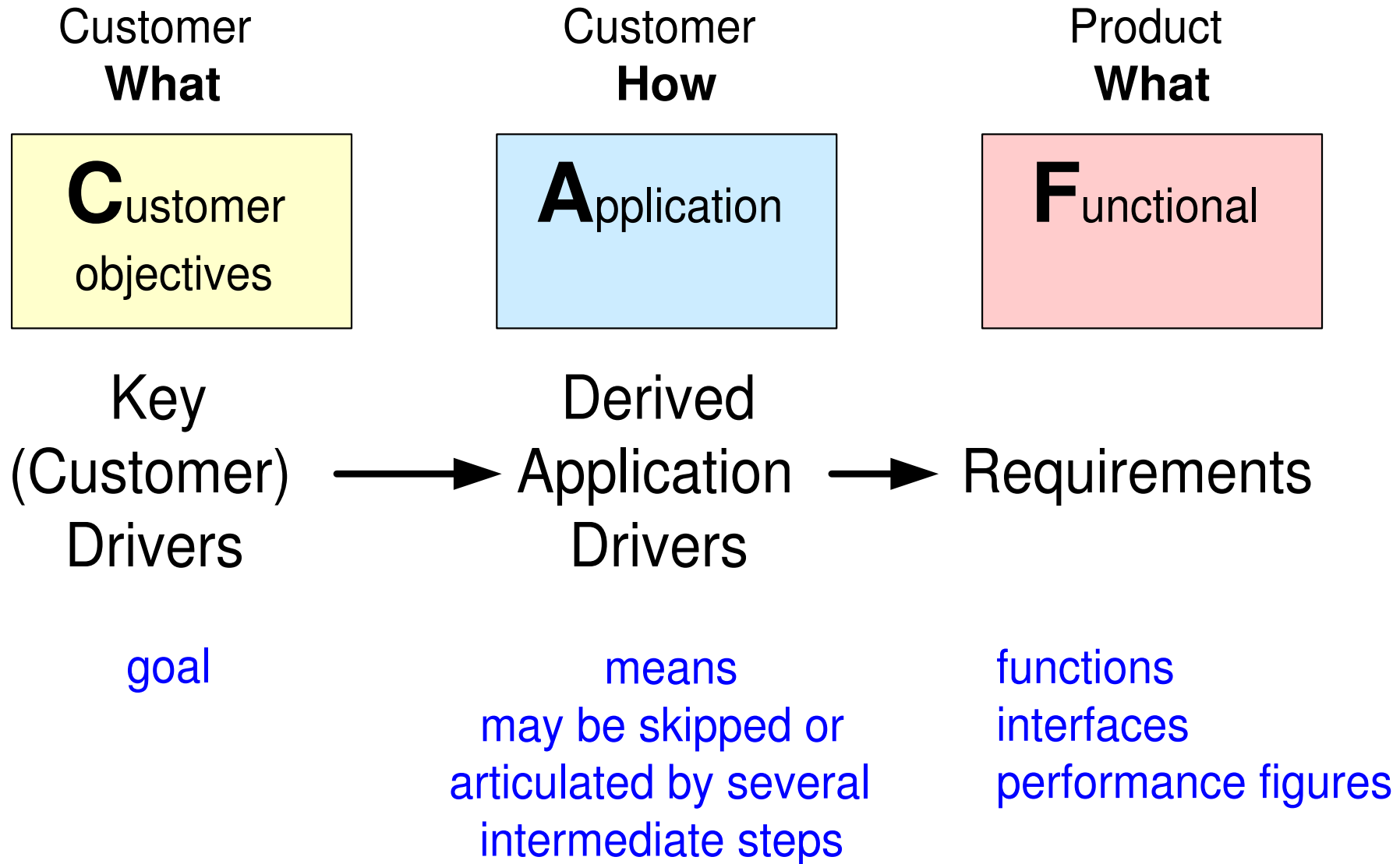
Method to create Key Driver Graph

- | | |
|--|--|
| • 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



Threads of Reasoning

by *Gerrit Muller* Embedded Systems Institute

e-mail: `gerrit.muller@embeddedsystems.nl`

`www.gaudisite.nl`

Abstract

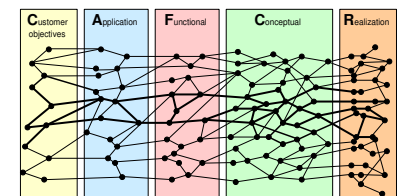
A method of reasoning is described, which addresses cross-cutting issues. The basis is fast iteration in the problem and solution space.

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.

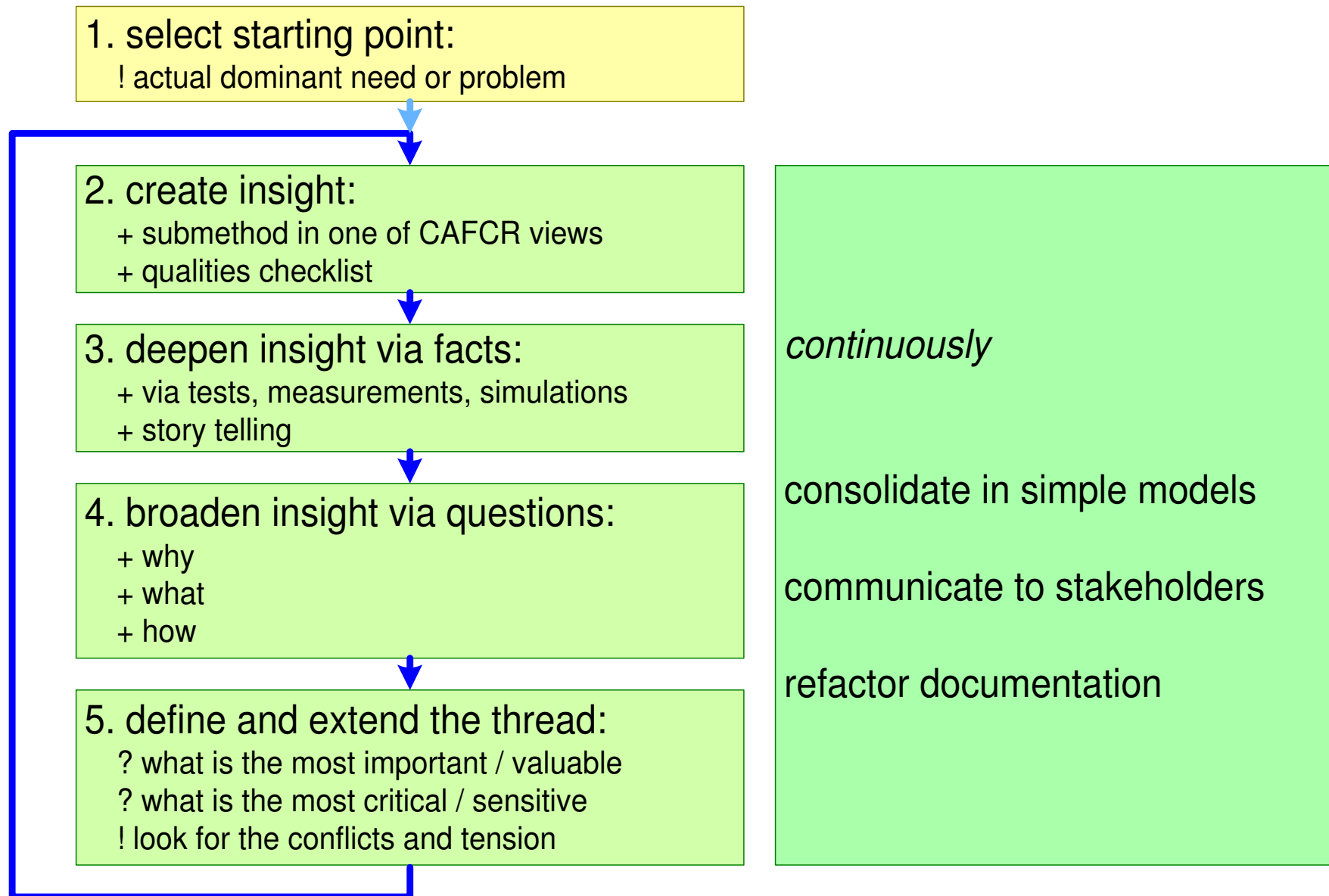
Distribution

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.

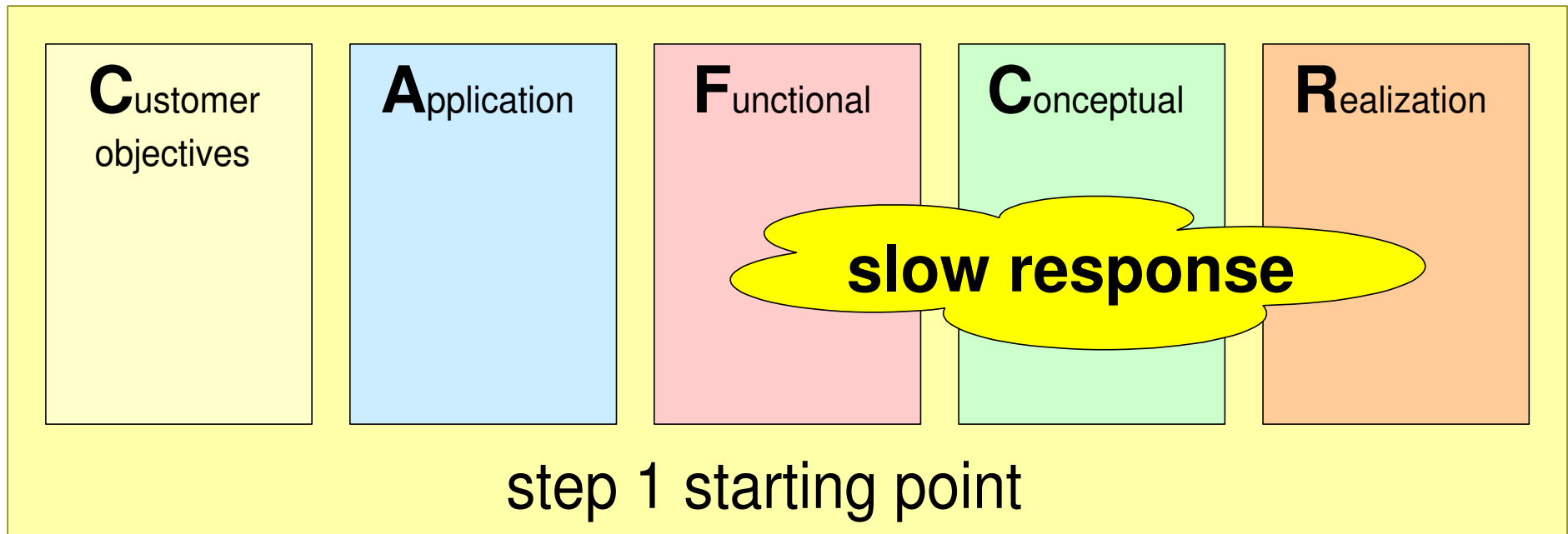
July 1, 2011
status: finished
version: 2.4



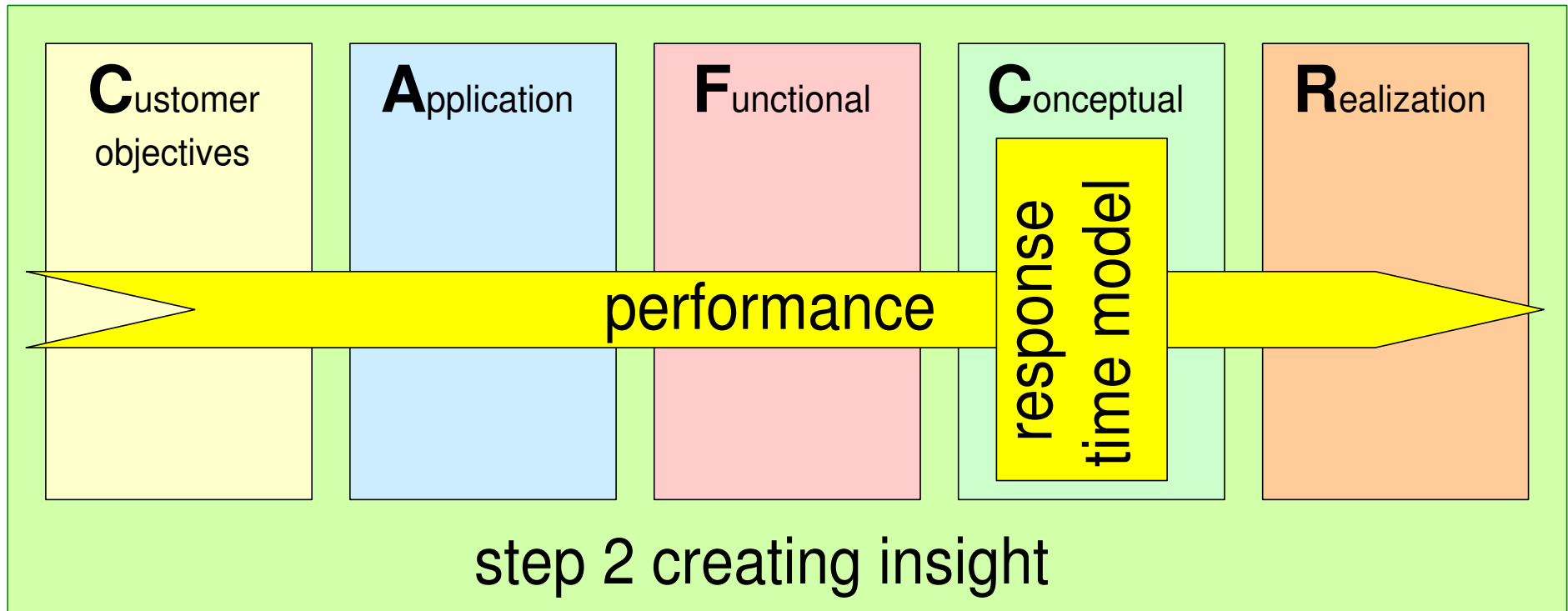
Overview of the reasoning approach



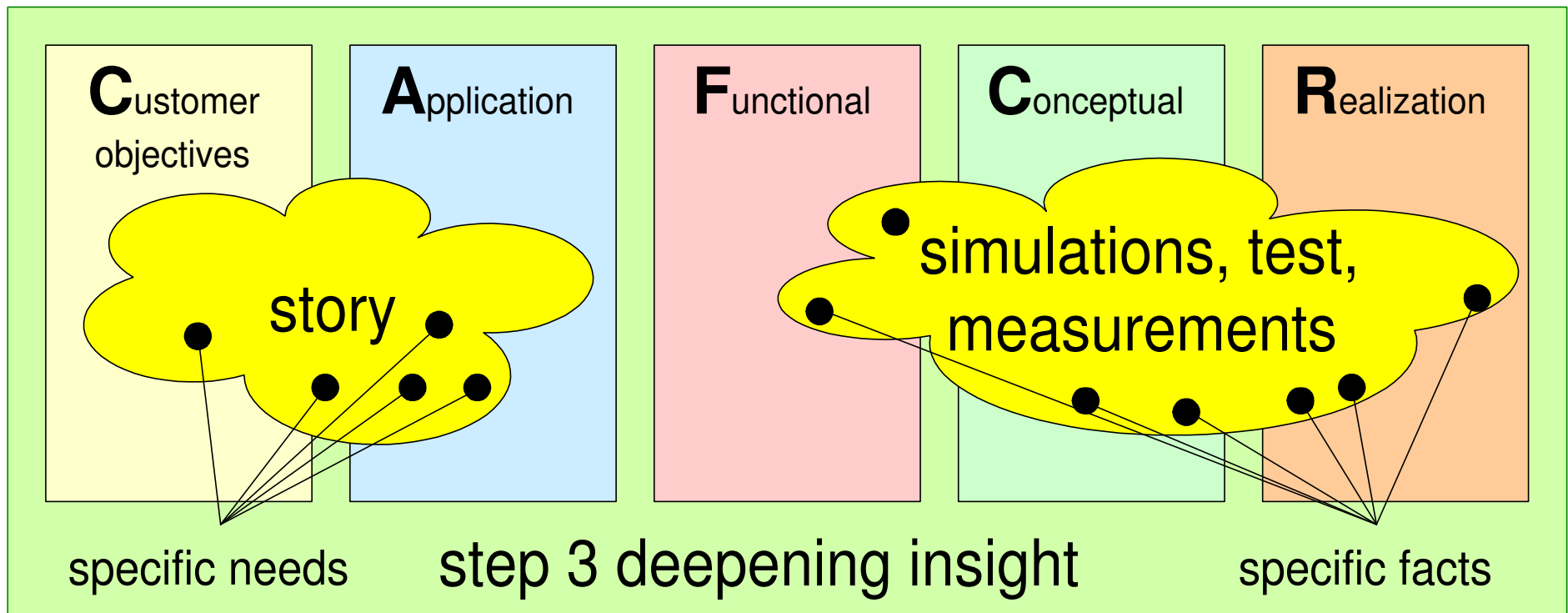
From starting point to insight



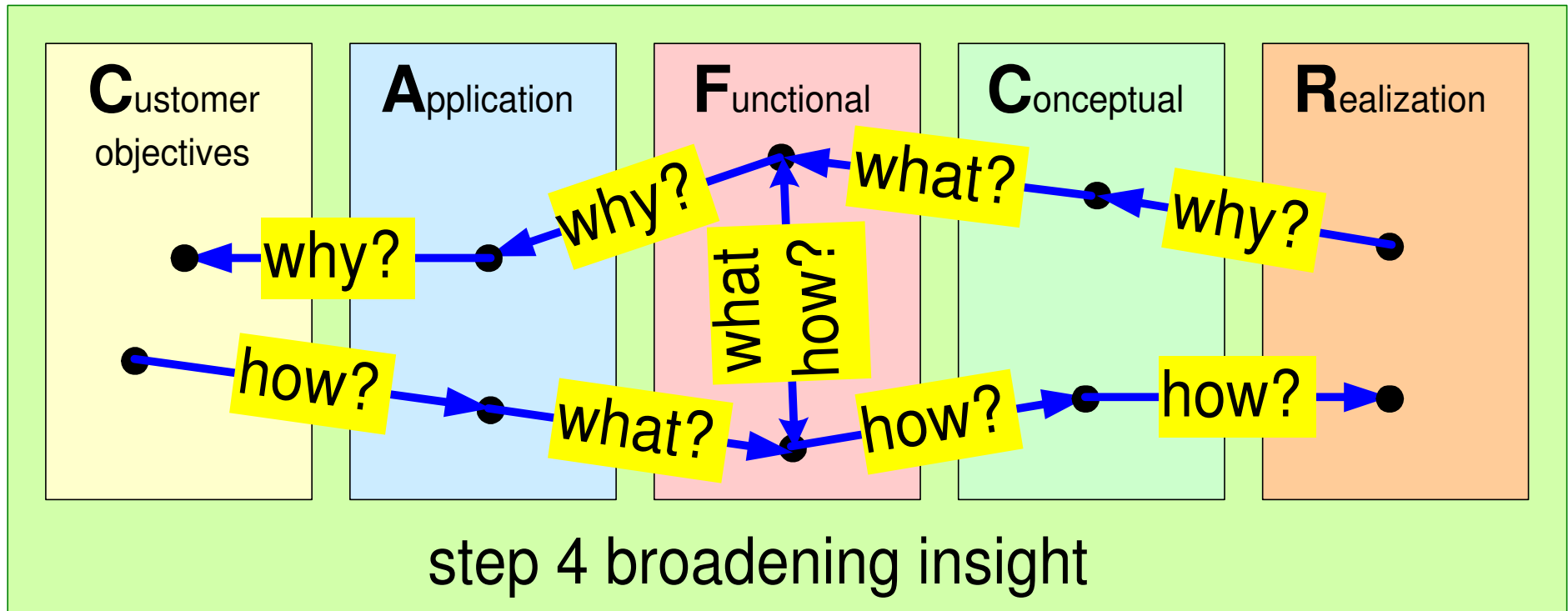
Creating Insight



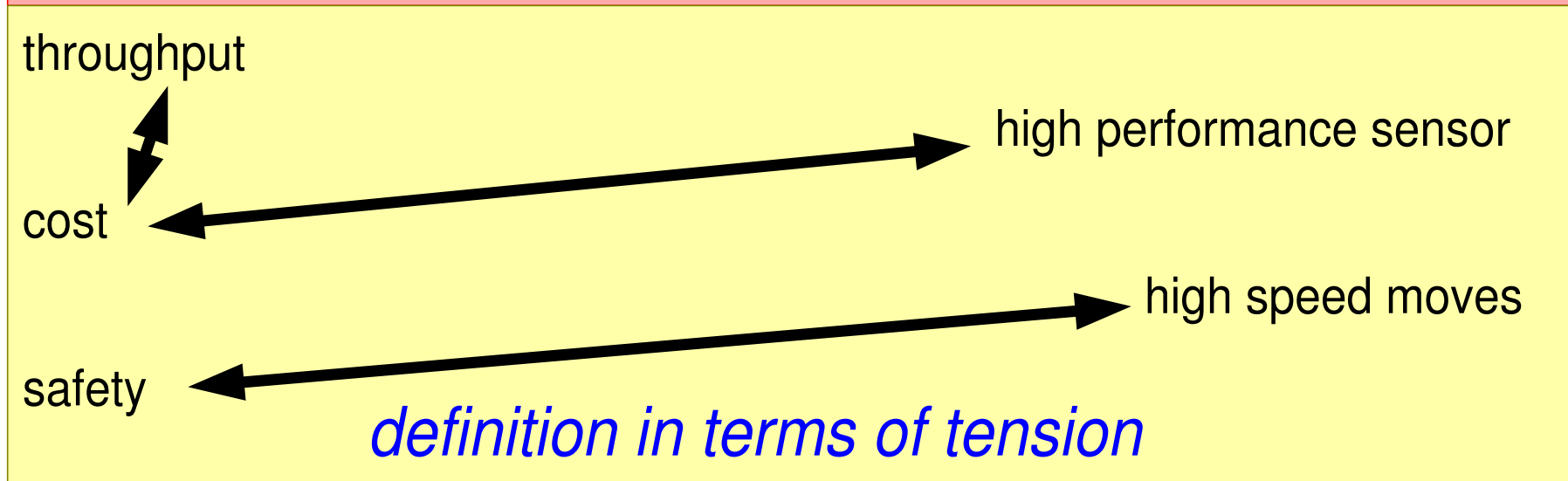
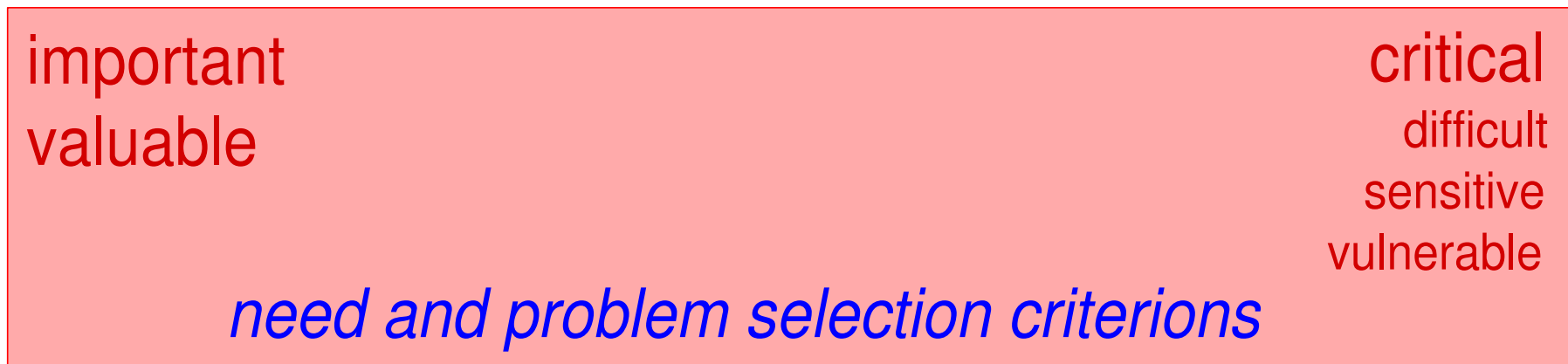
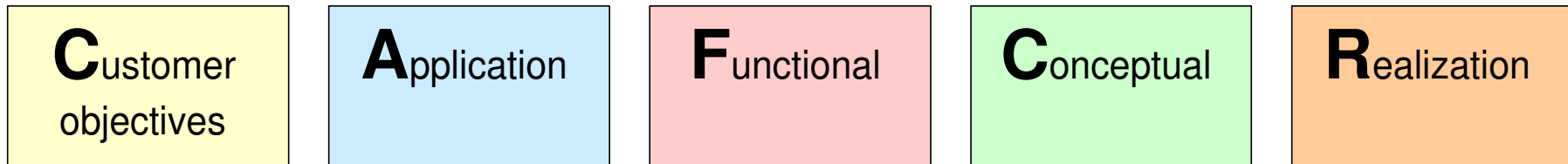
Deepening Insight



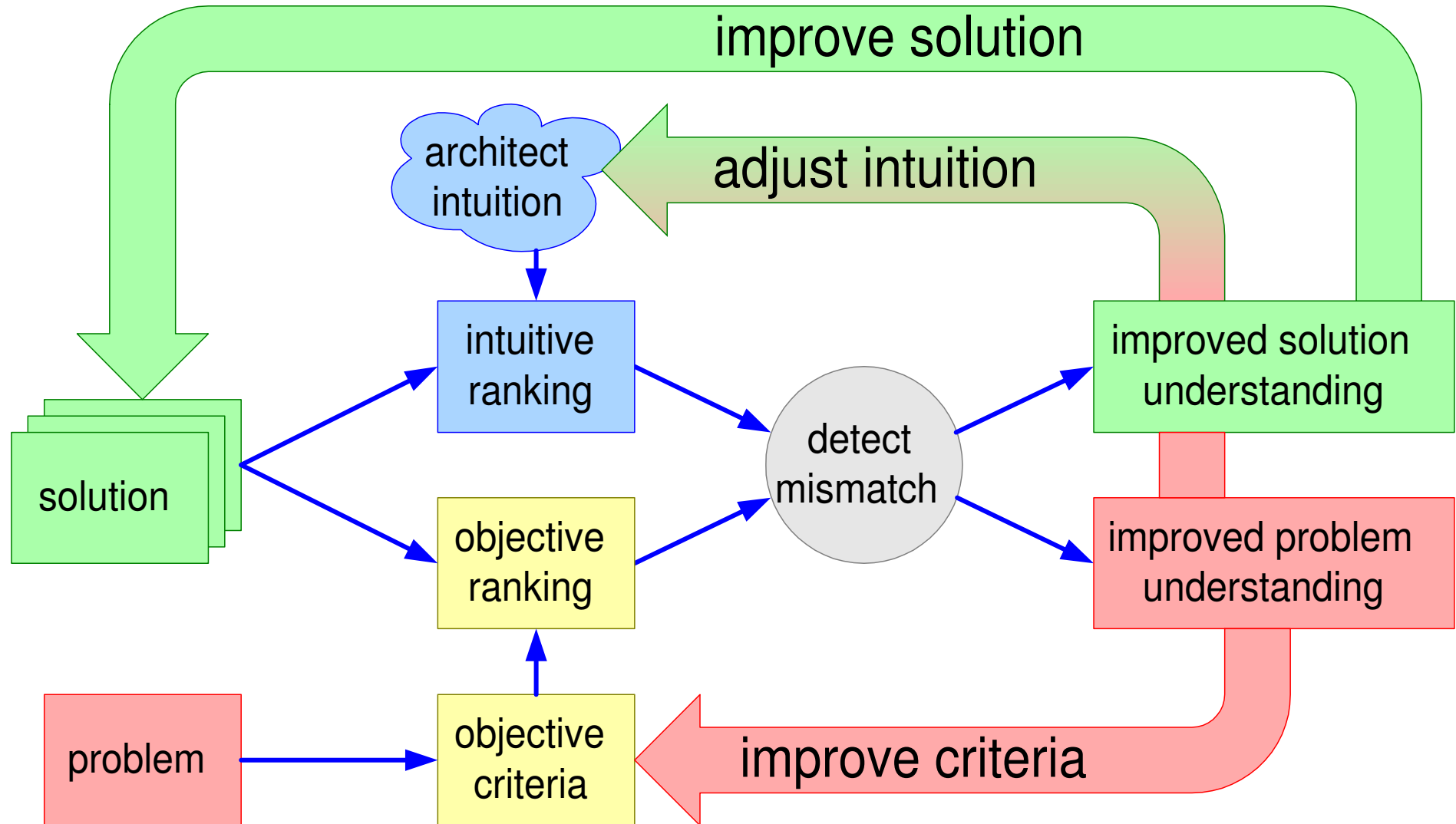
Broadening Insight



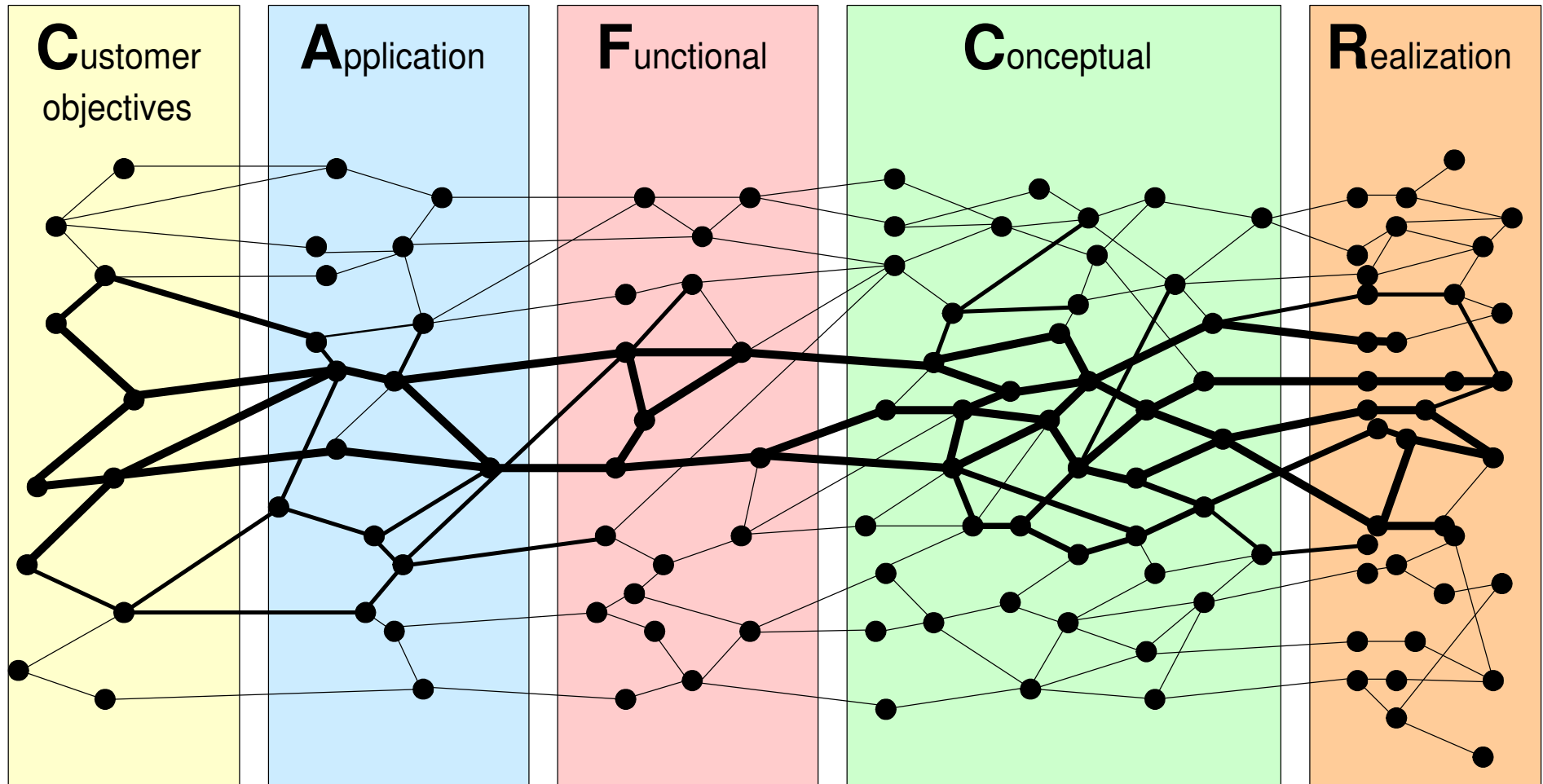
Problem identification and articulation



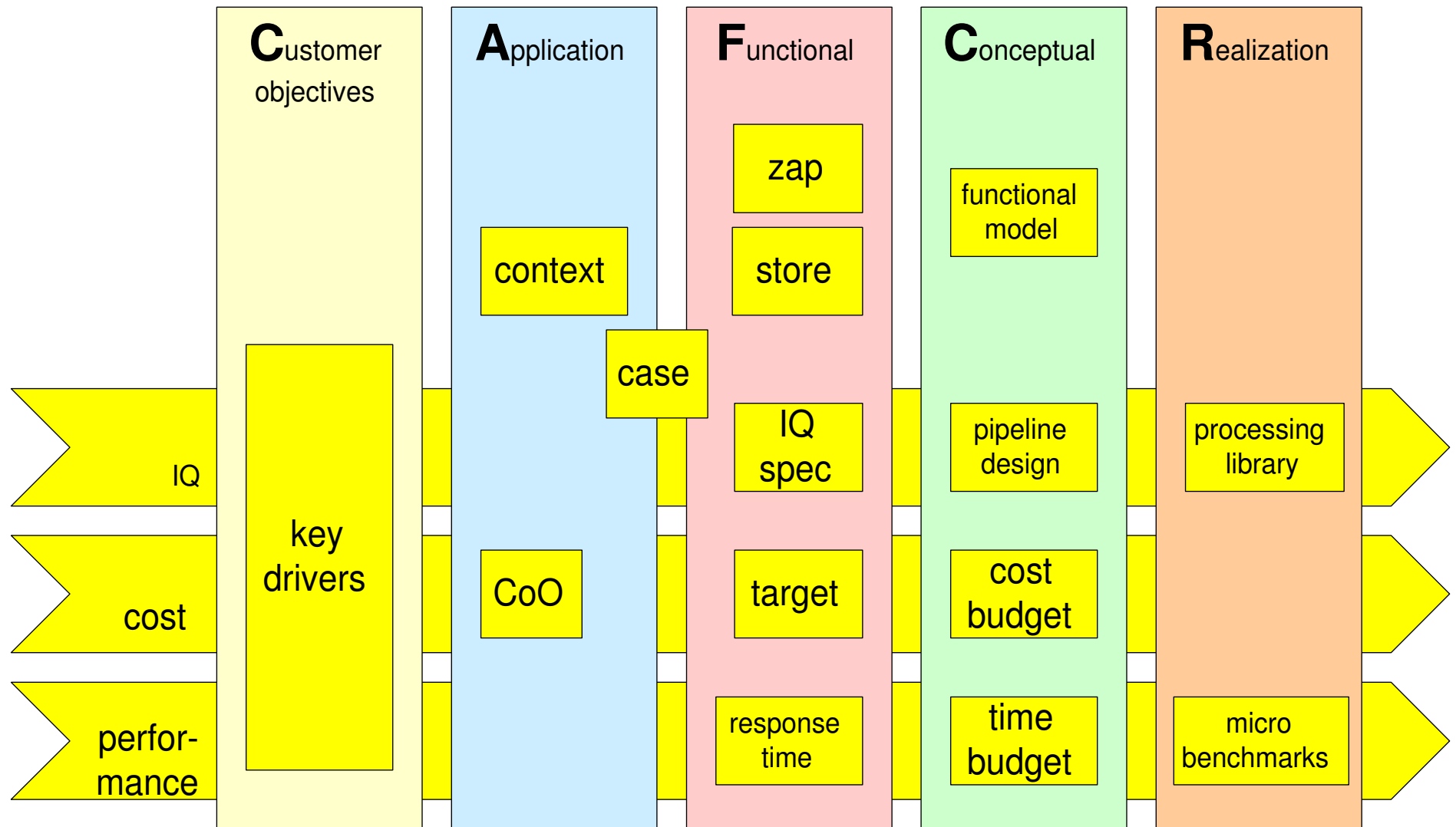
Iteration during the analysis



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:

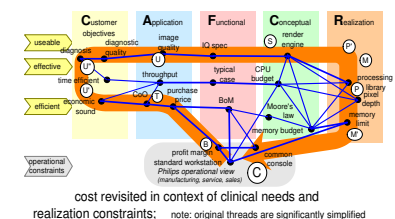
- the CAFCR viewpoints
- qualities as integrating needles
- story telling
- threads of reasoning

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

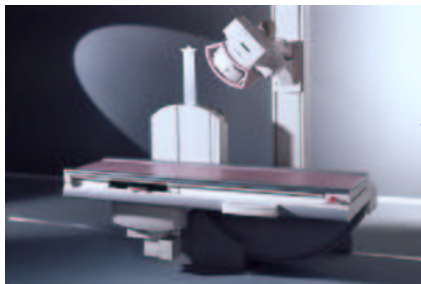
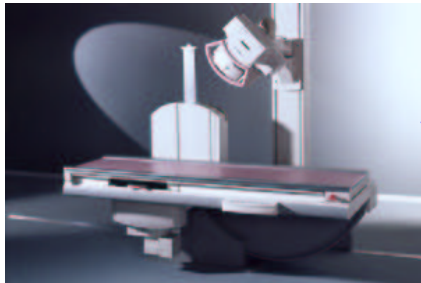
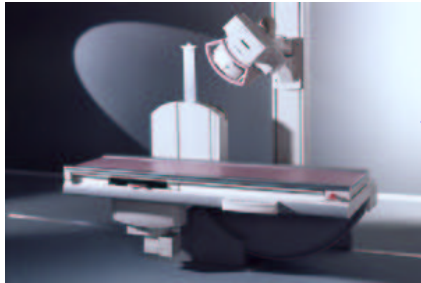
Distribution

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.

July 1, 2011
status: preliminary
draft
version: 0



Easyvision serving three URF examination rooms



URF-systems

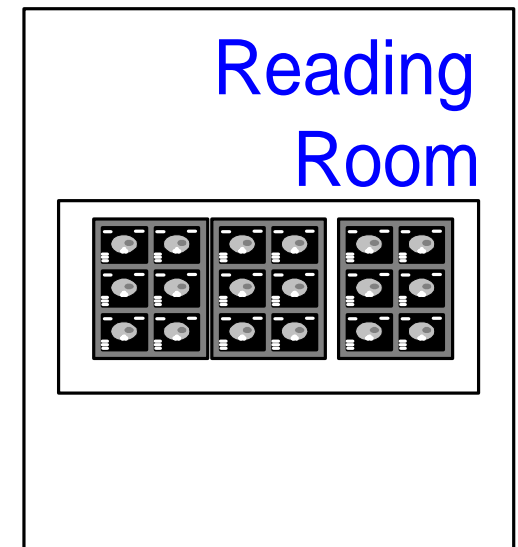
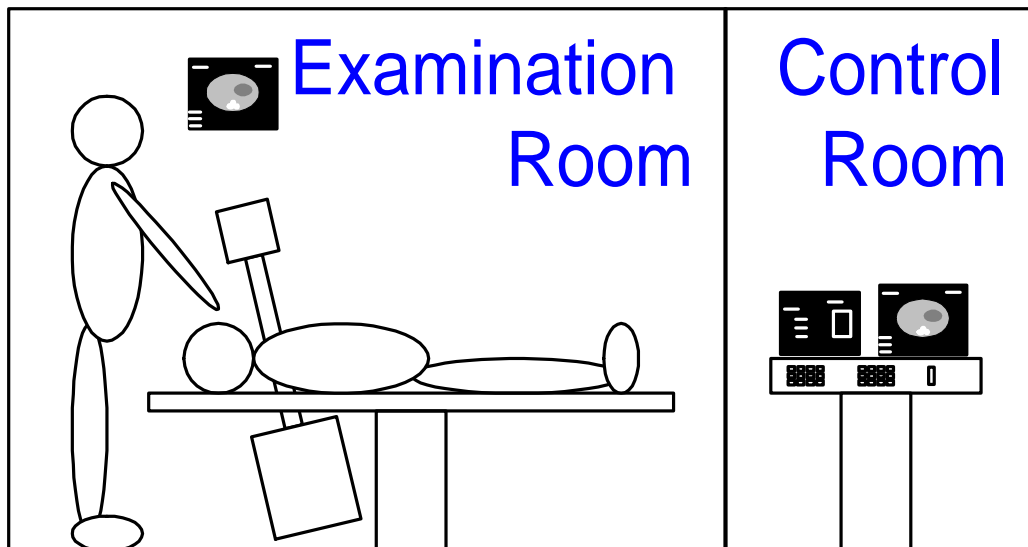
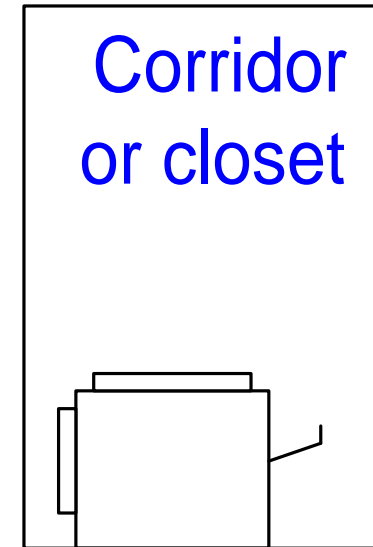
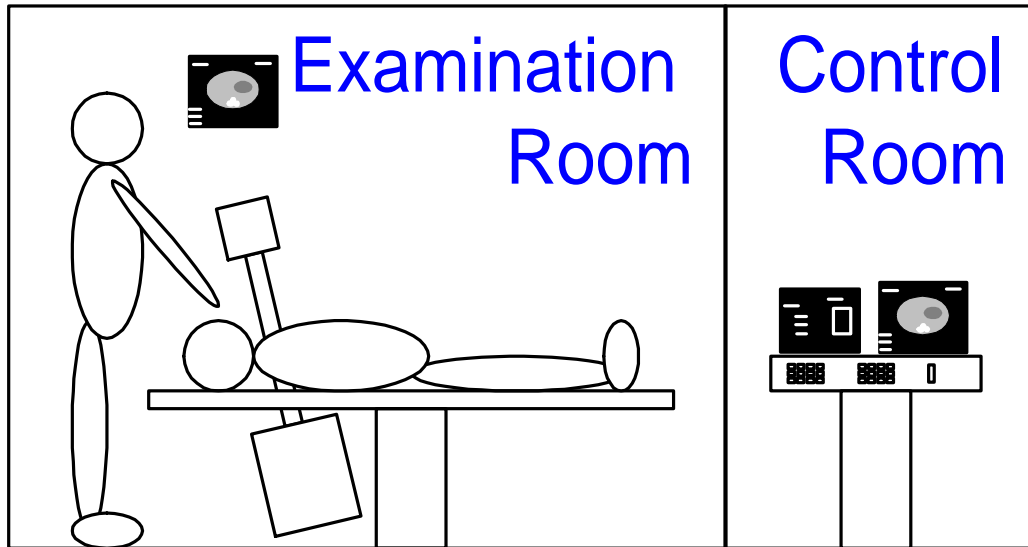


EasyVision: Medical Imaging Workstation

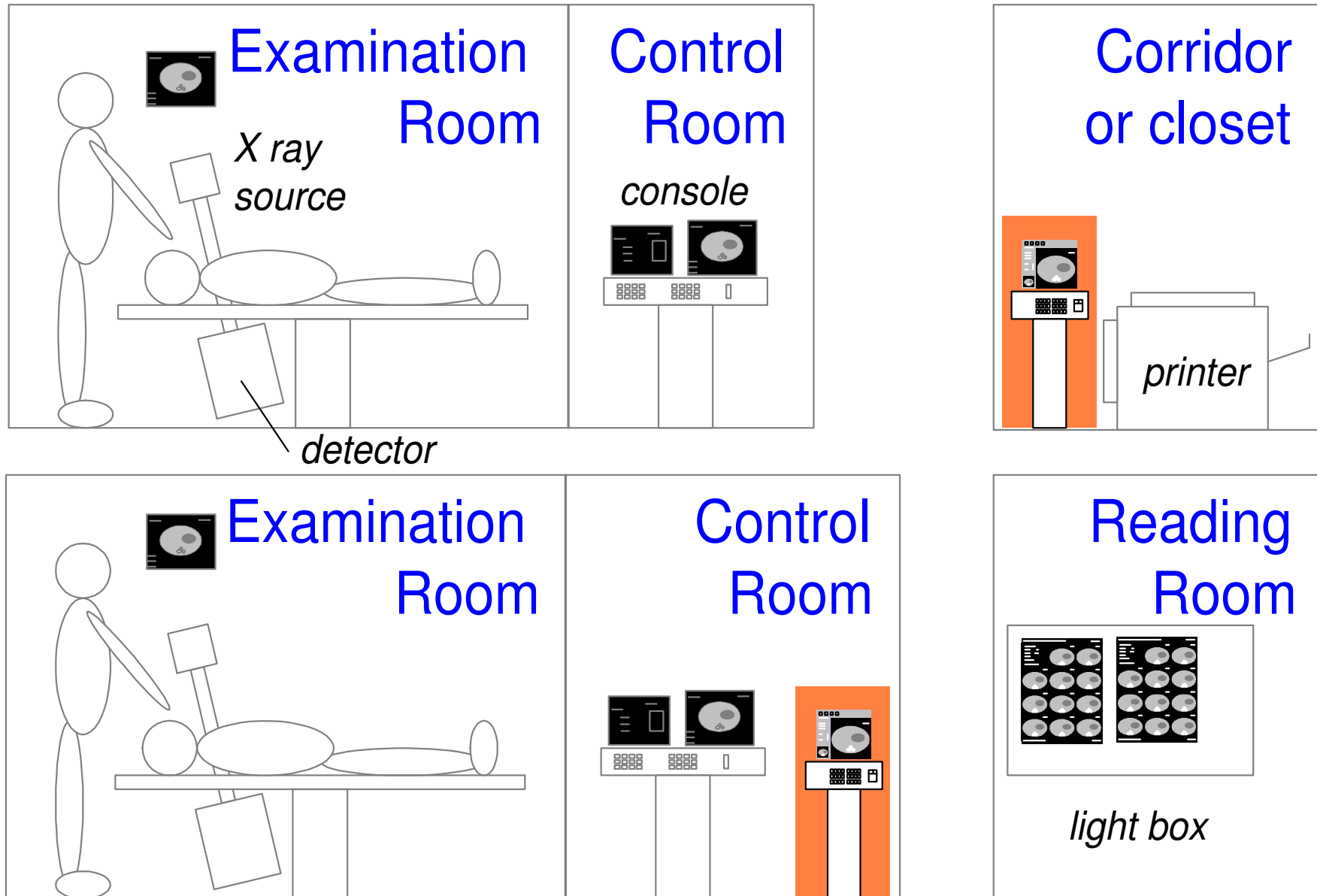


typical clinical image (intestines)

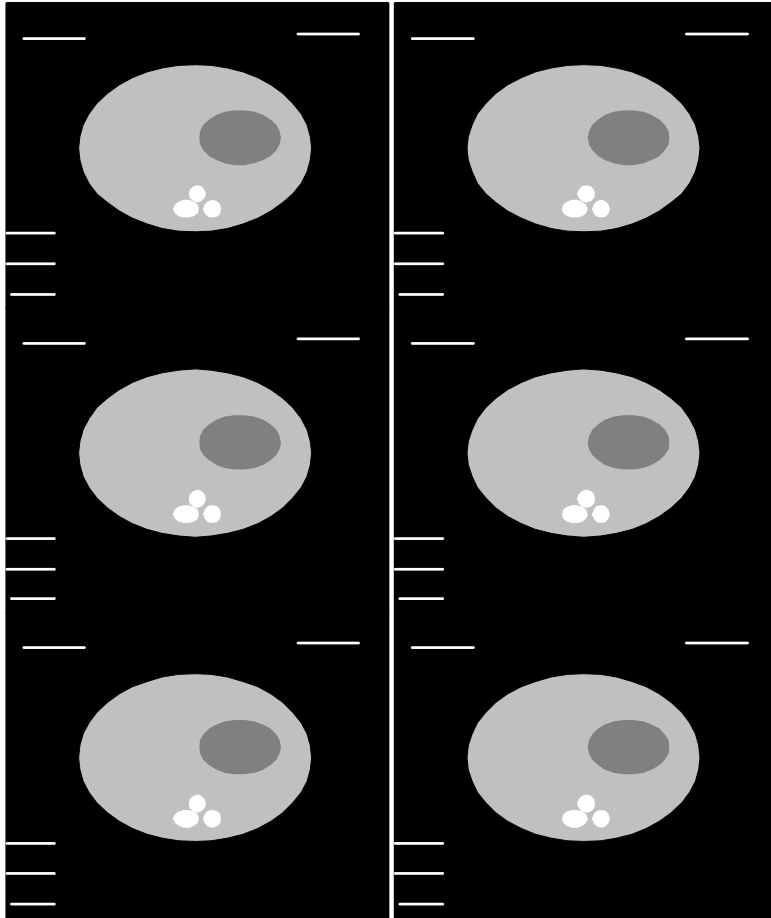
X-ray rooms from examination to reading around 1990



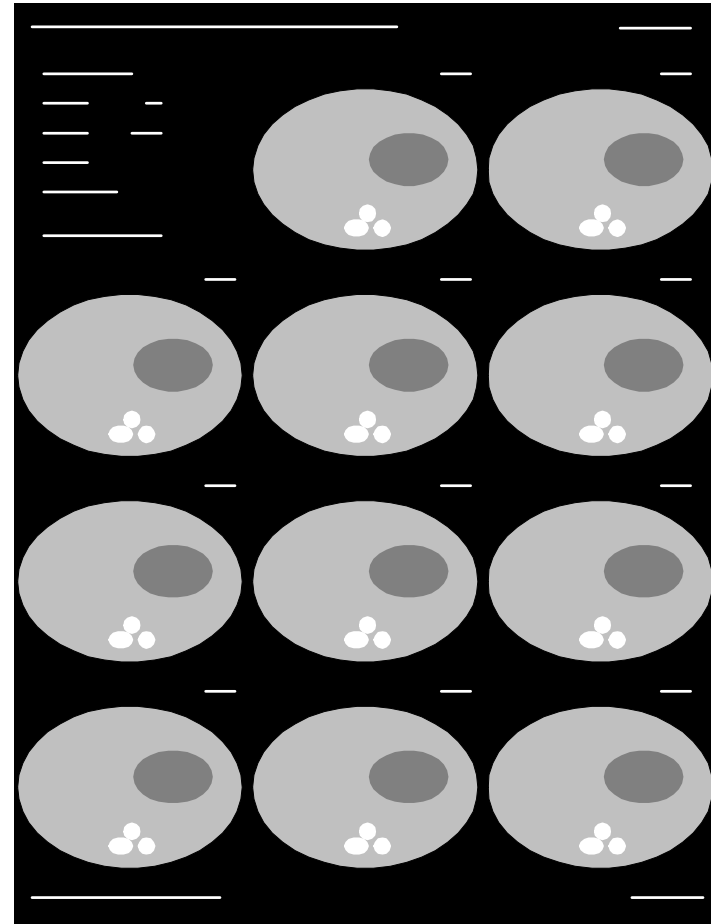
X-ray rooms with Easyvision applied as printserver



Comparison screen copy versus optimized film



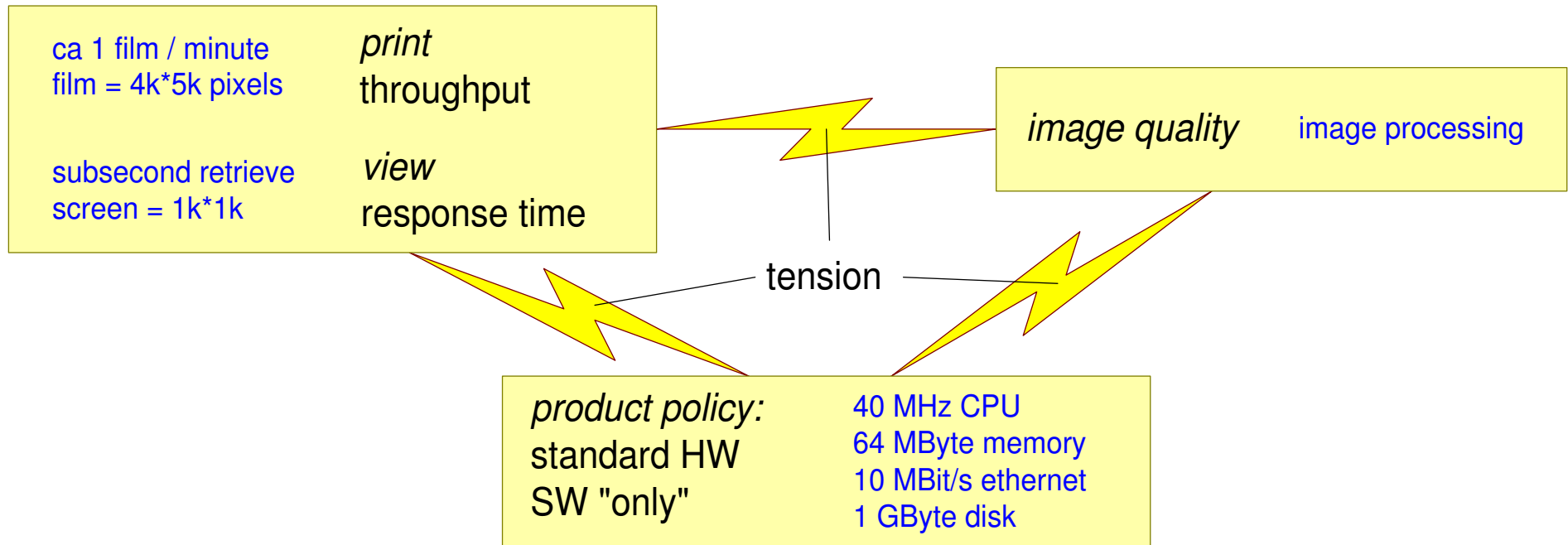
old: screen copy



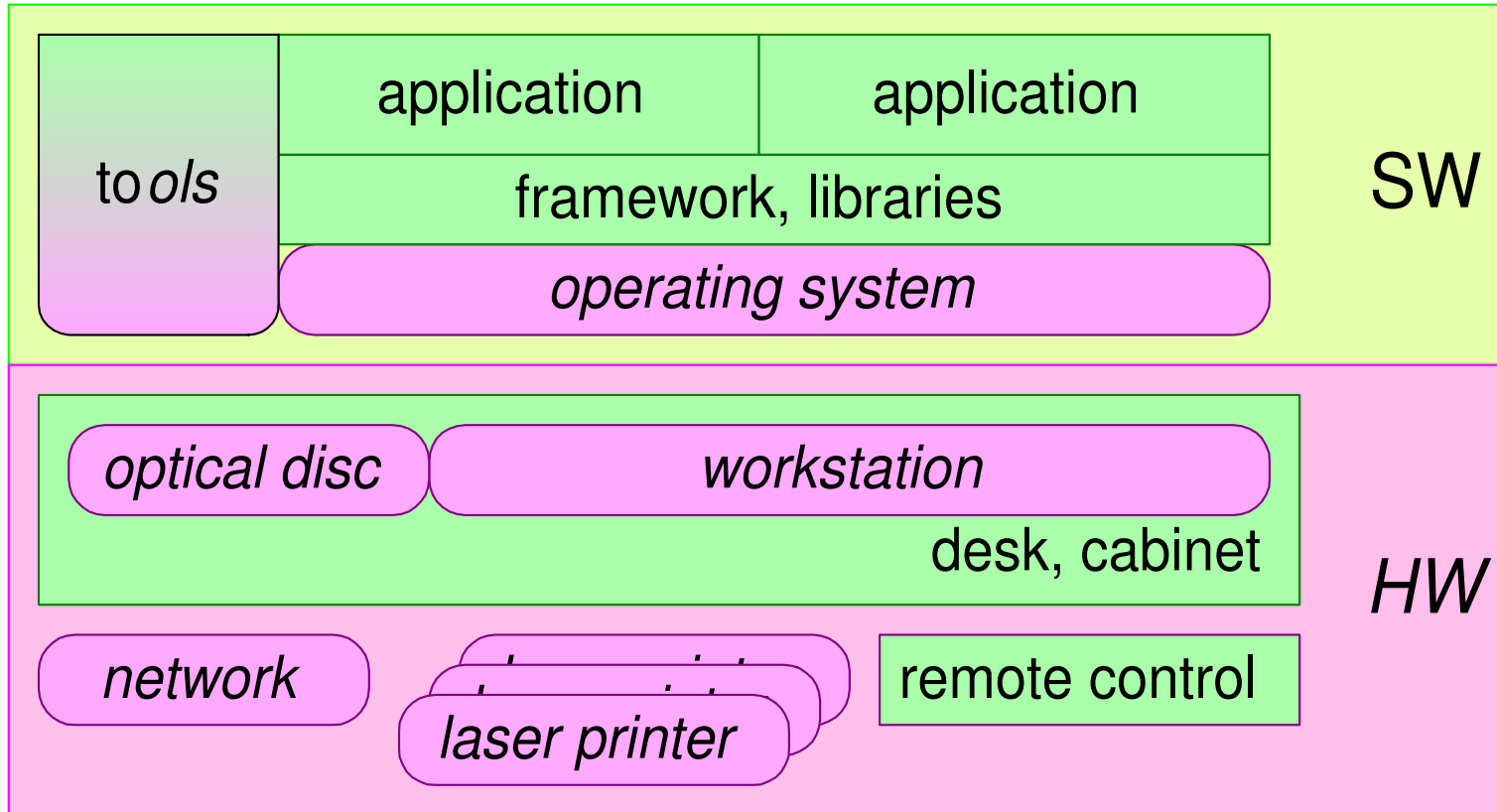
new: SW formatting

20 to 50% less film needed

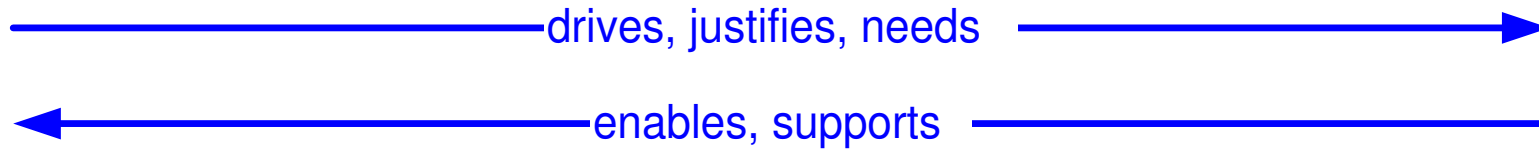
Challenges for product creation



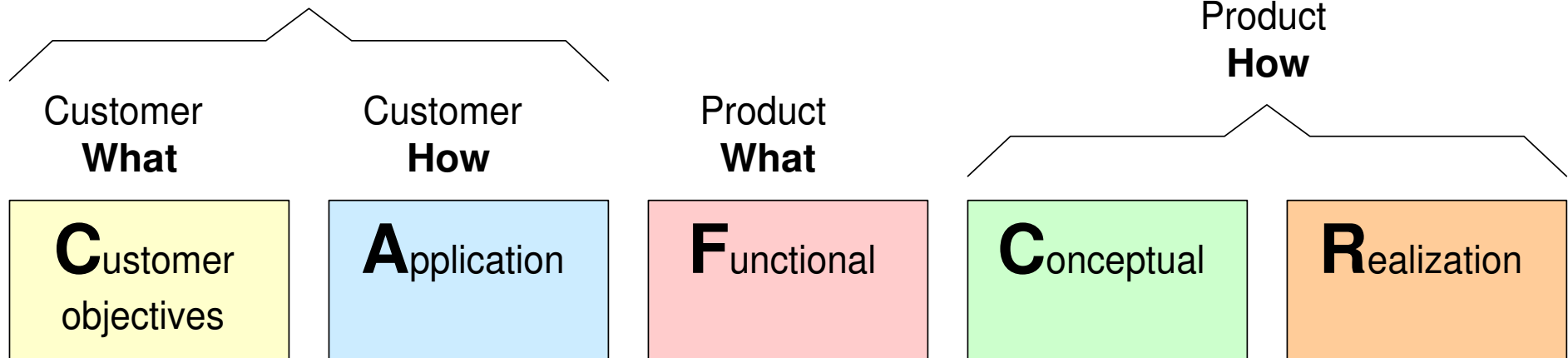
Top level decomposition



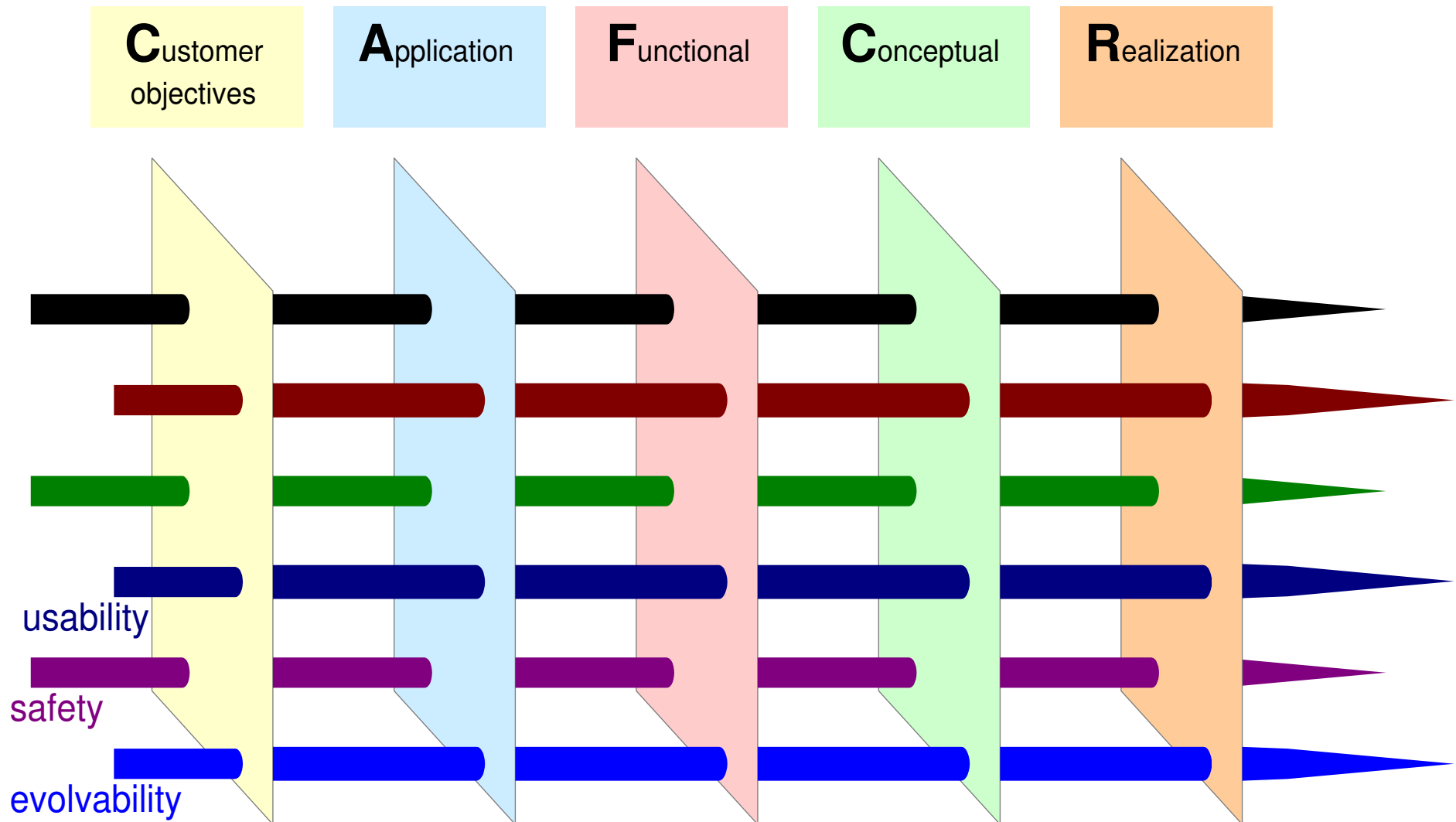
CAFCR viewpoints



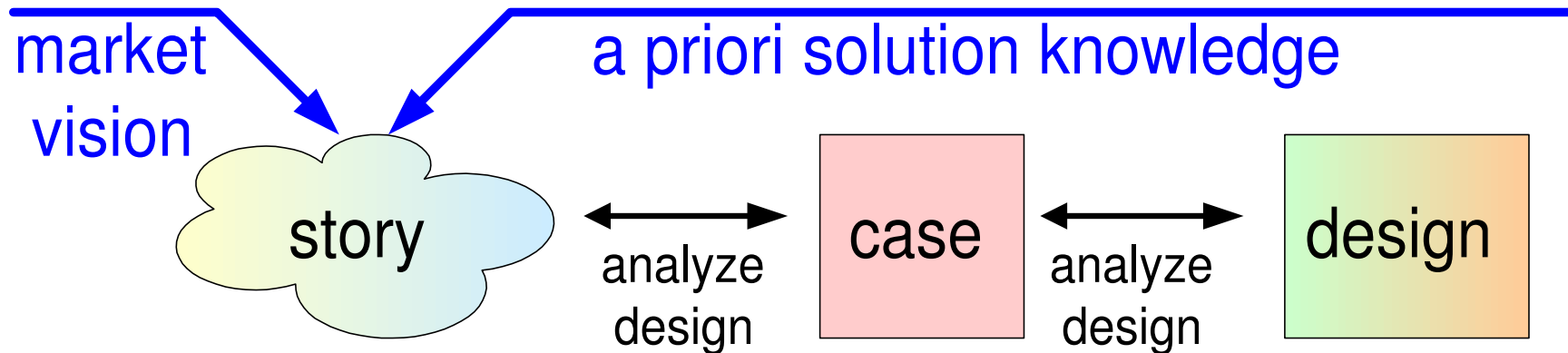
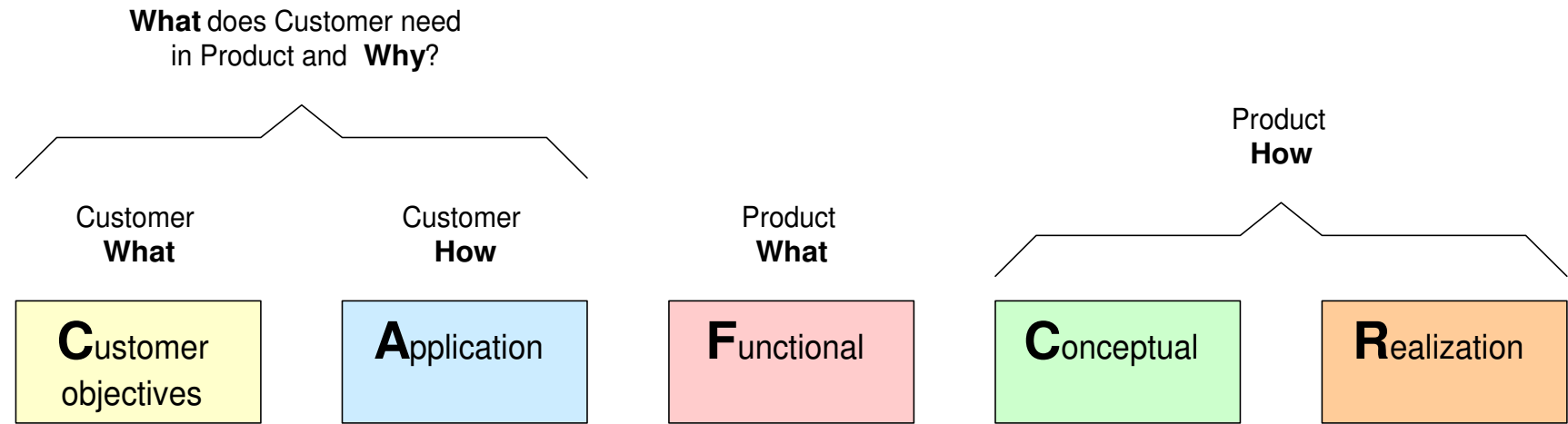
What does Customer need
in Product and **Why?**



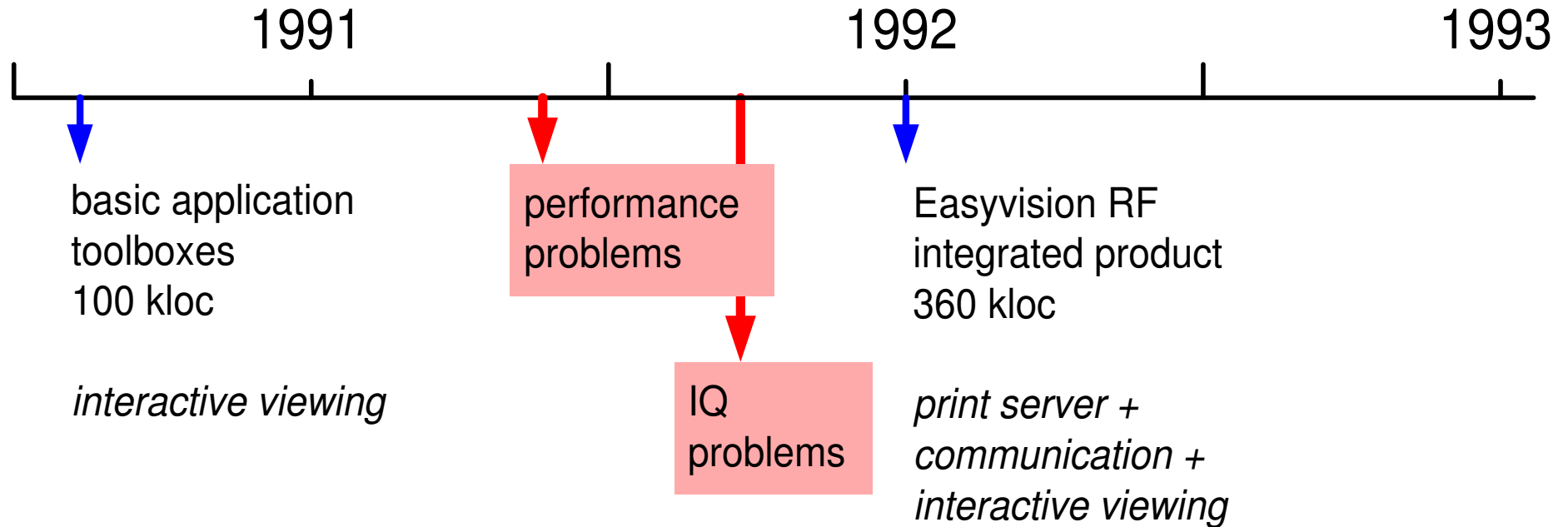
Quality needles as generic integrating concepts



From story to design

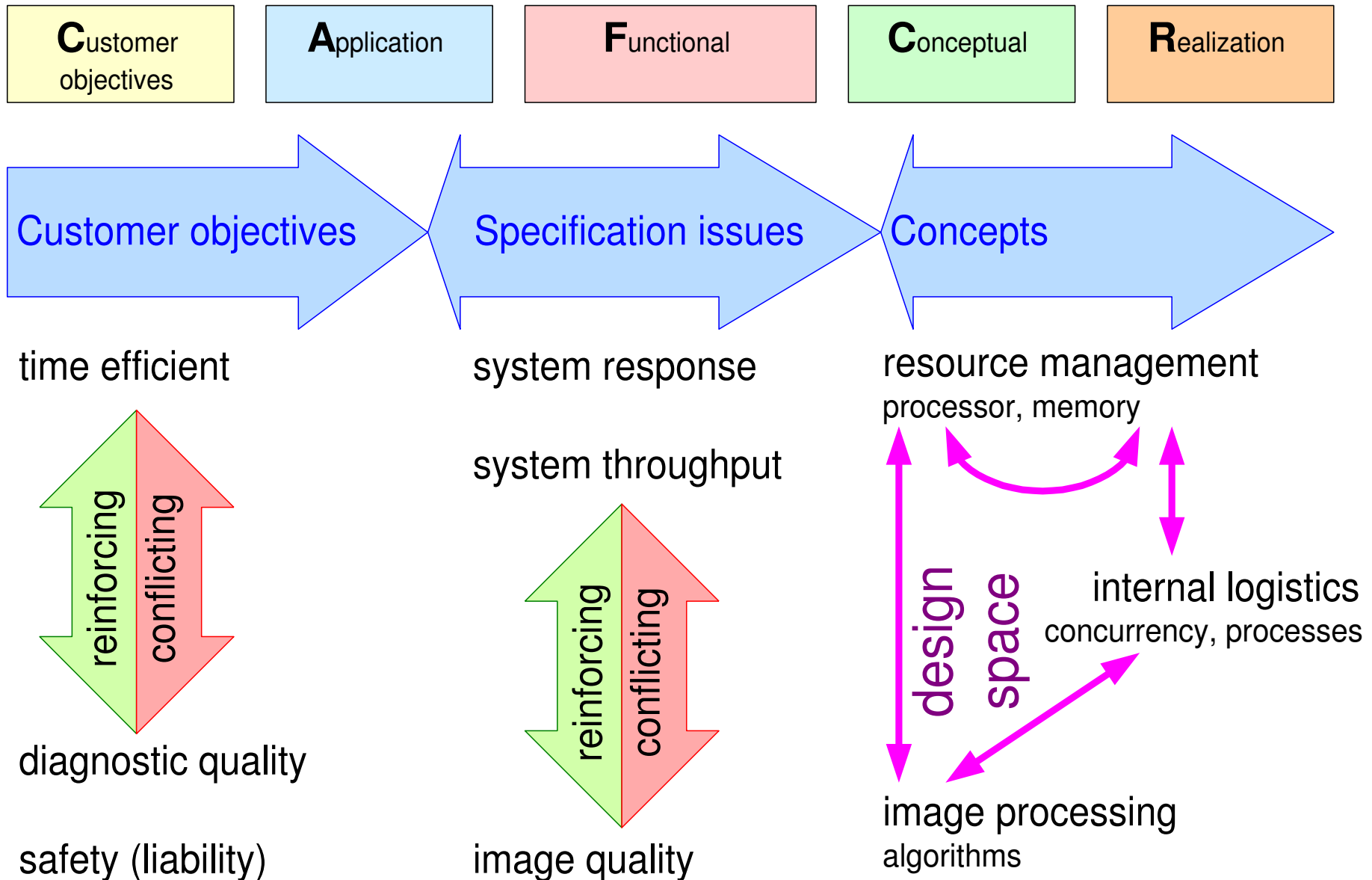


Chronology of Easyvision RF R1 development



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

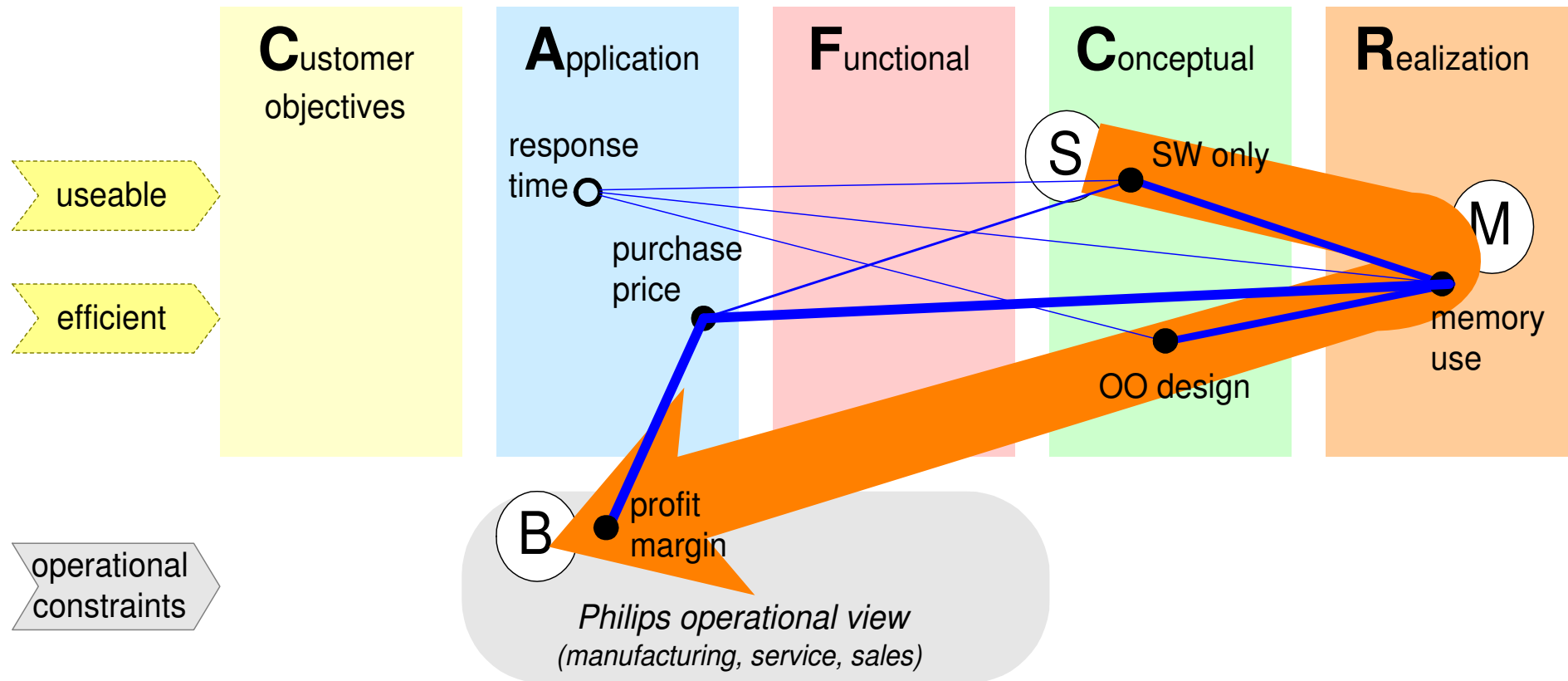
data base engine, fast, reliable and robust

extensive set of toolboxes

property based configuration

multiple coordinate spaces

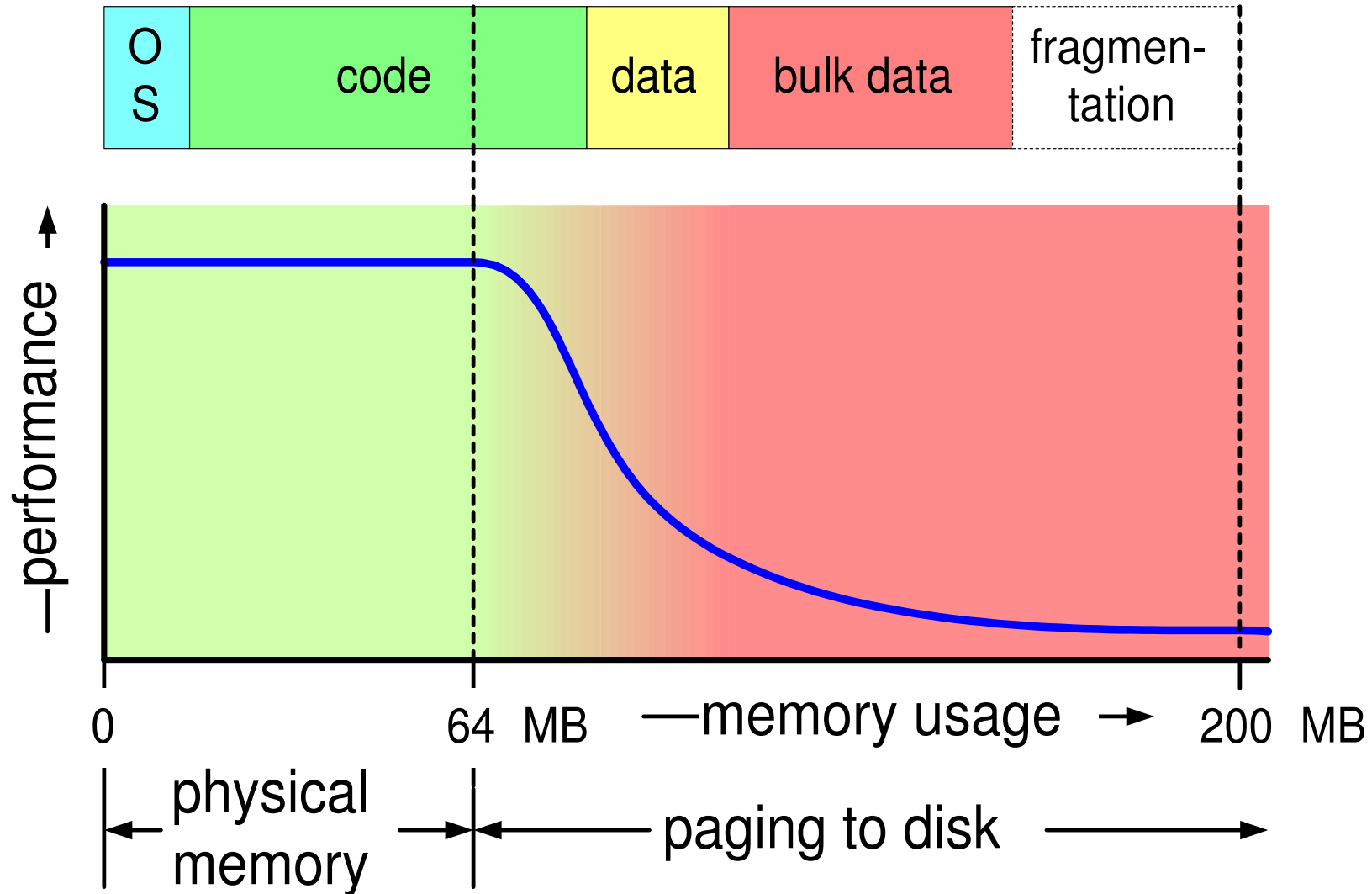
Thread of reasoning; introvert phase



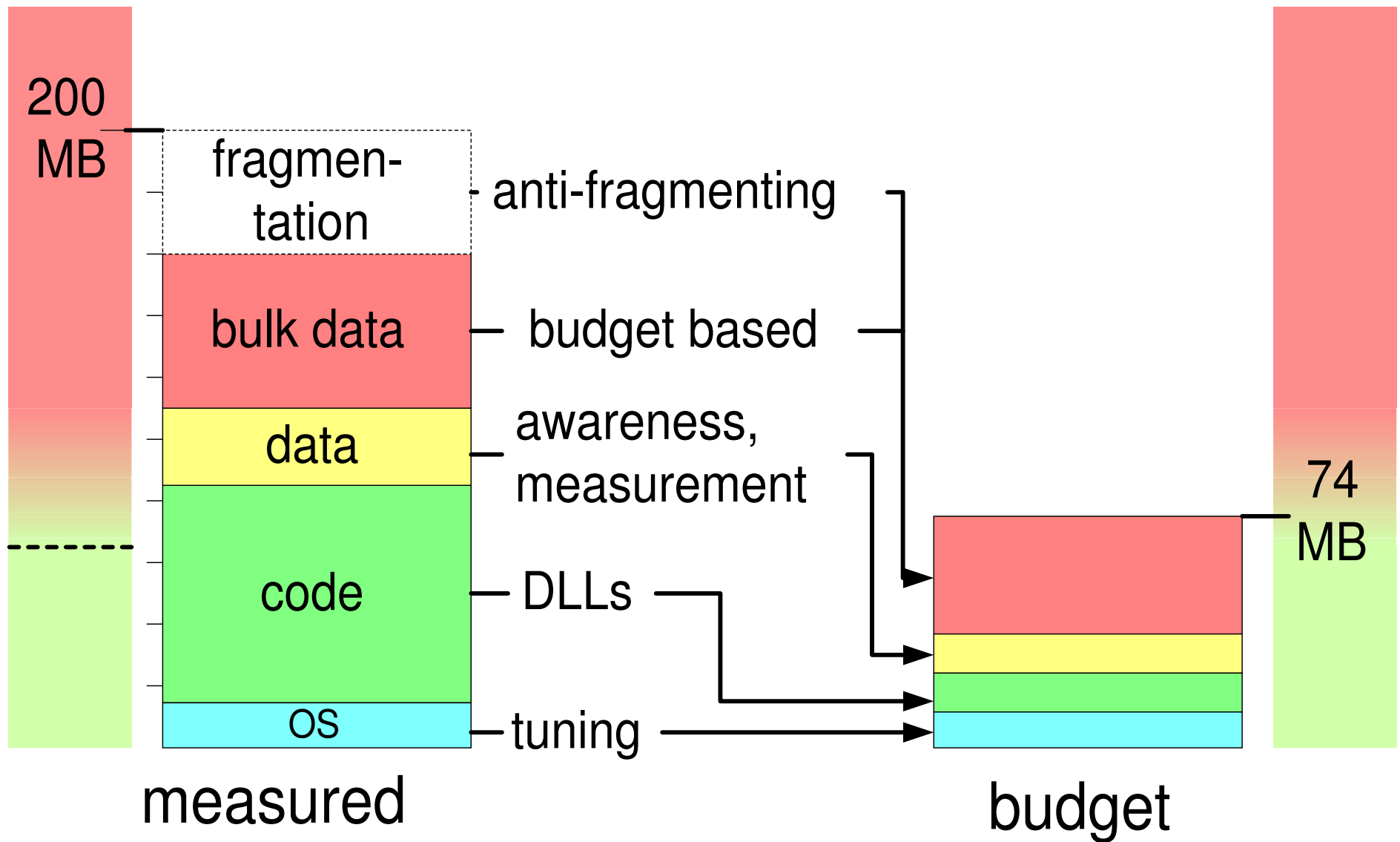
Introvert view: cost and impact of new technologies

Memory usage half way R1

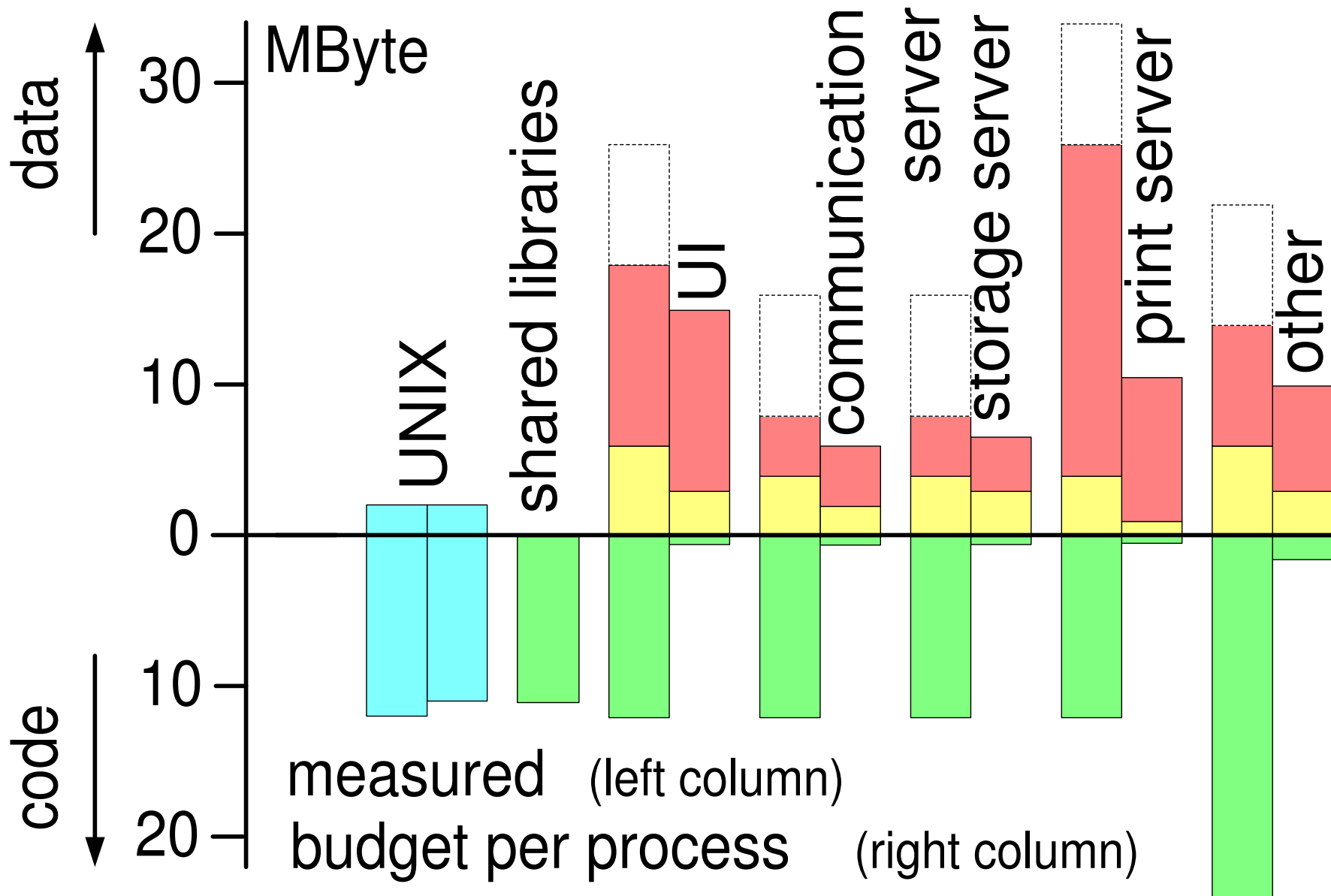
total measured memory usage



Solution of memory performance problem



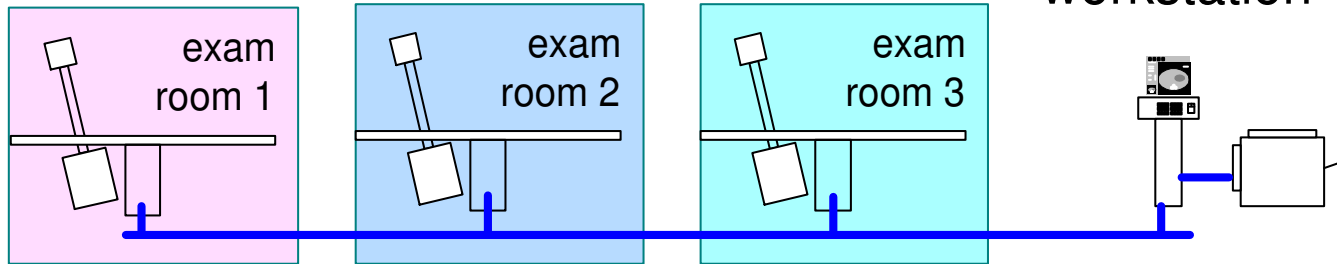
Visualization memory use per process



Typical case URF examination

3 examination rooms connected to

1 medical imaging workstation + printer

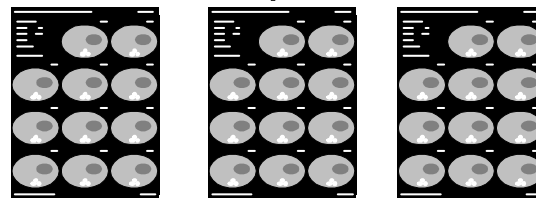


examination room: average 4 interleaved examinations / hour

image production: 20 1024² 8 bit images per examination

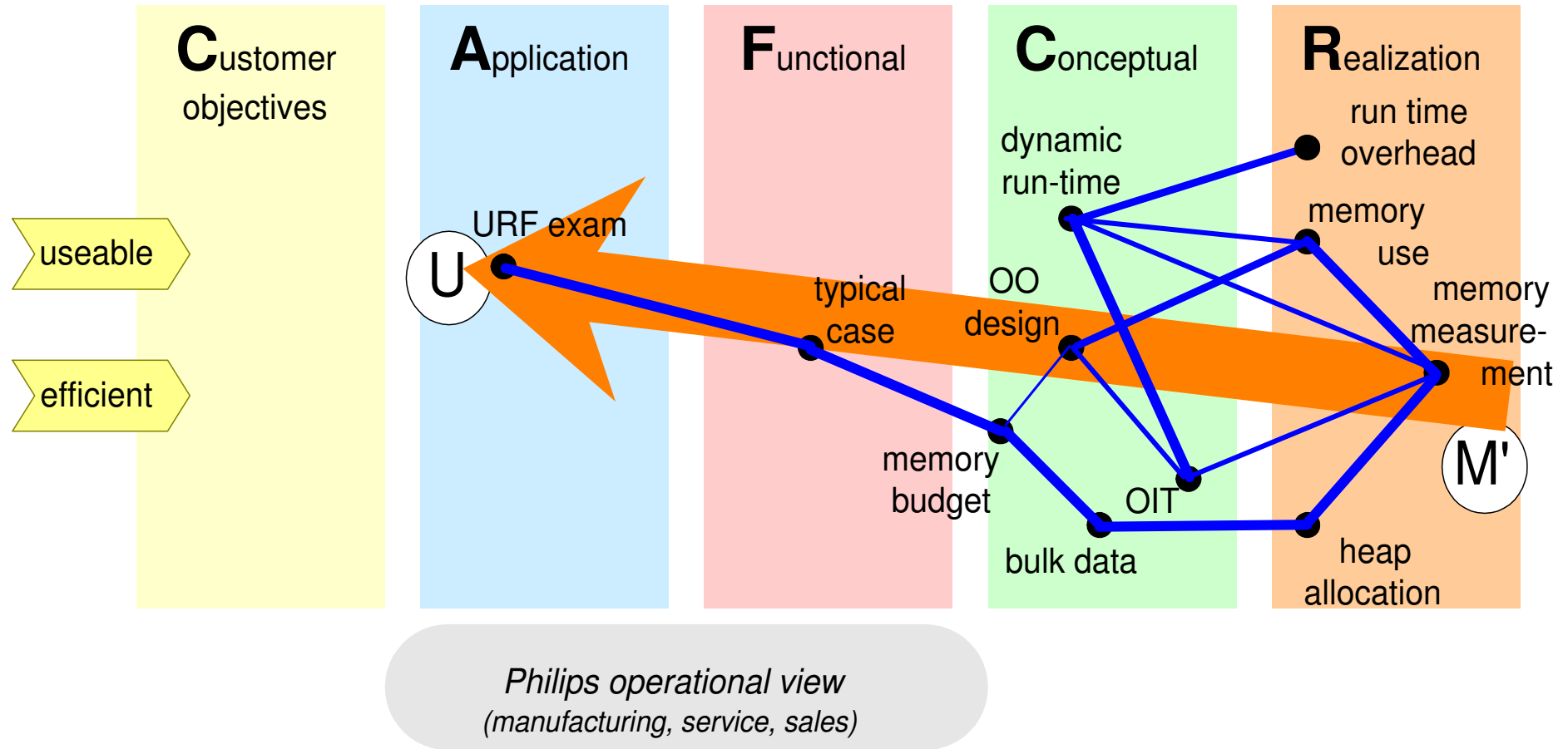


film production: 3 films of 4k*5k pixels each



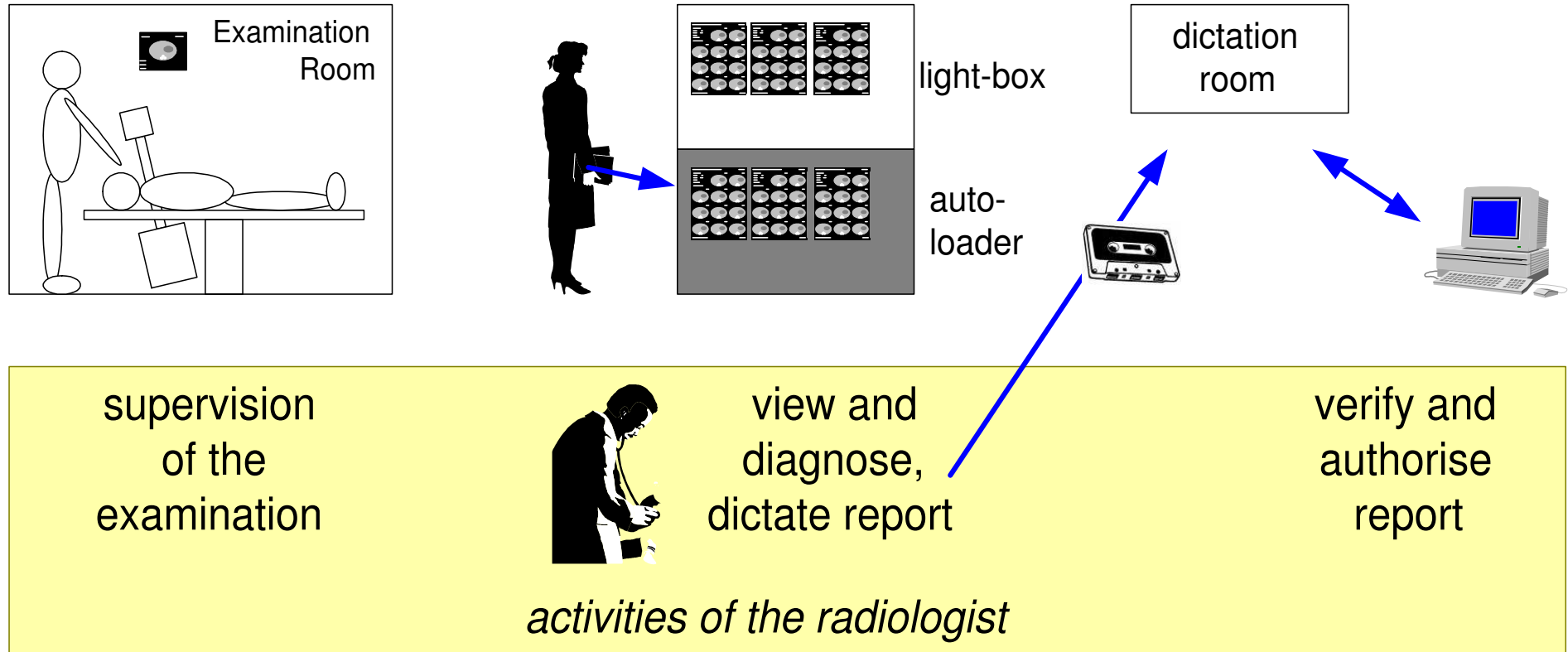
high quality output
(bi-cubic interpolation)

Thread of reasoning; phase 2

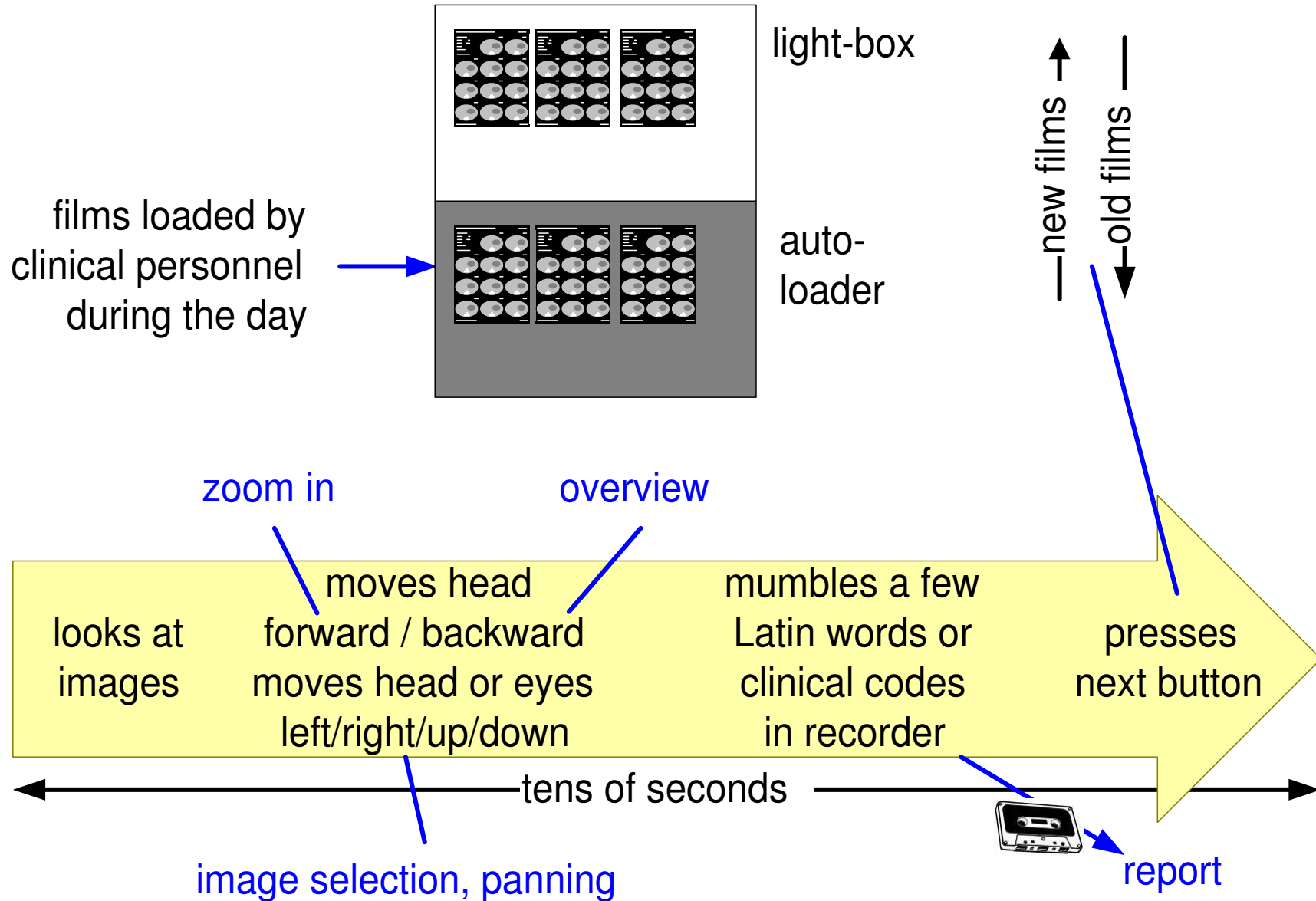


How to measure memory, how much is needed?
from introvert to extrovert

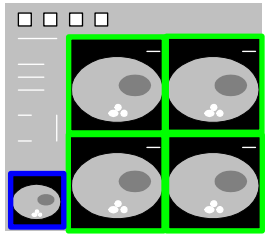
Radiologist workspots and activities



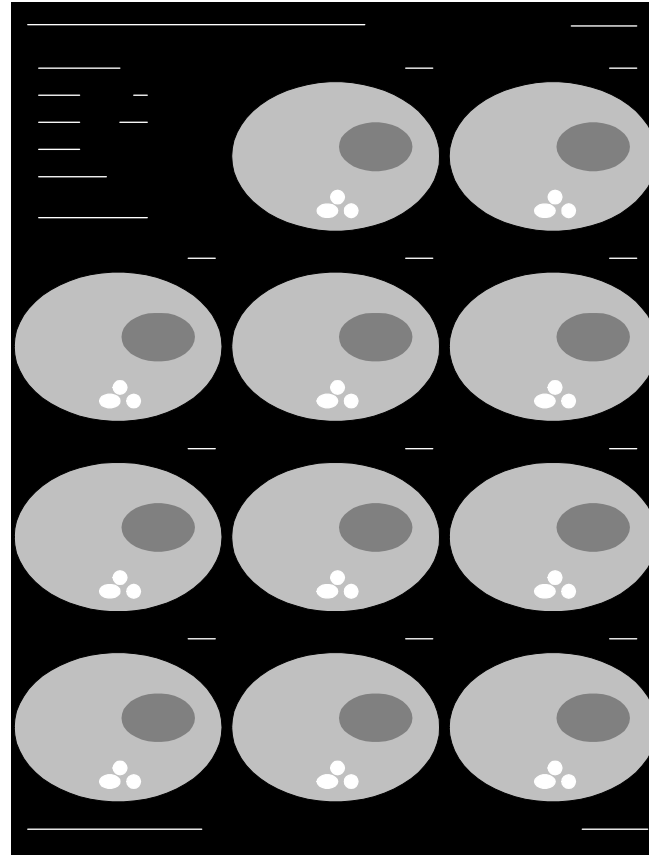
Diagnosis in tens of seconds



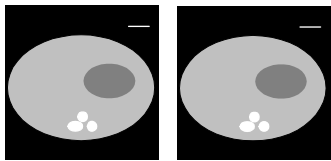
Rendered images at different destinations



Screen:
low resolution
fast response



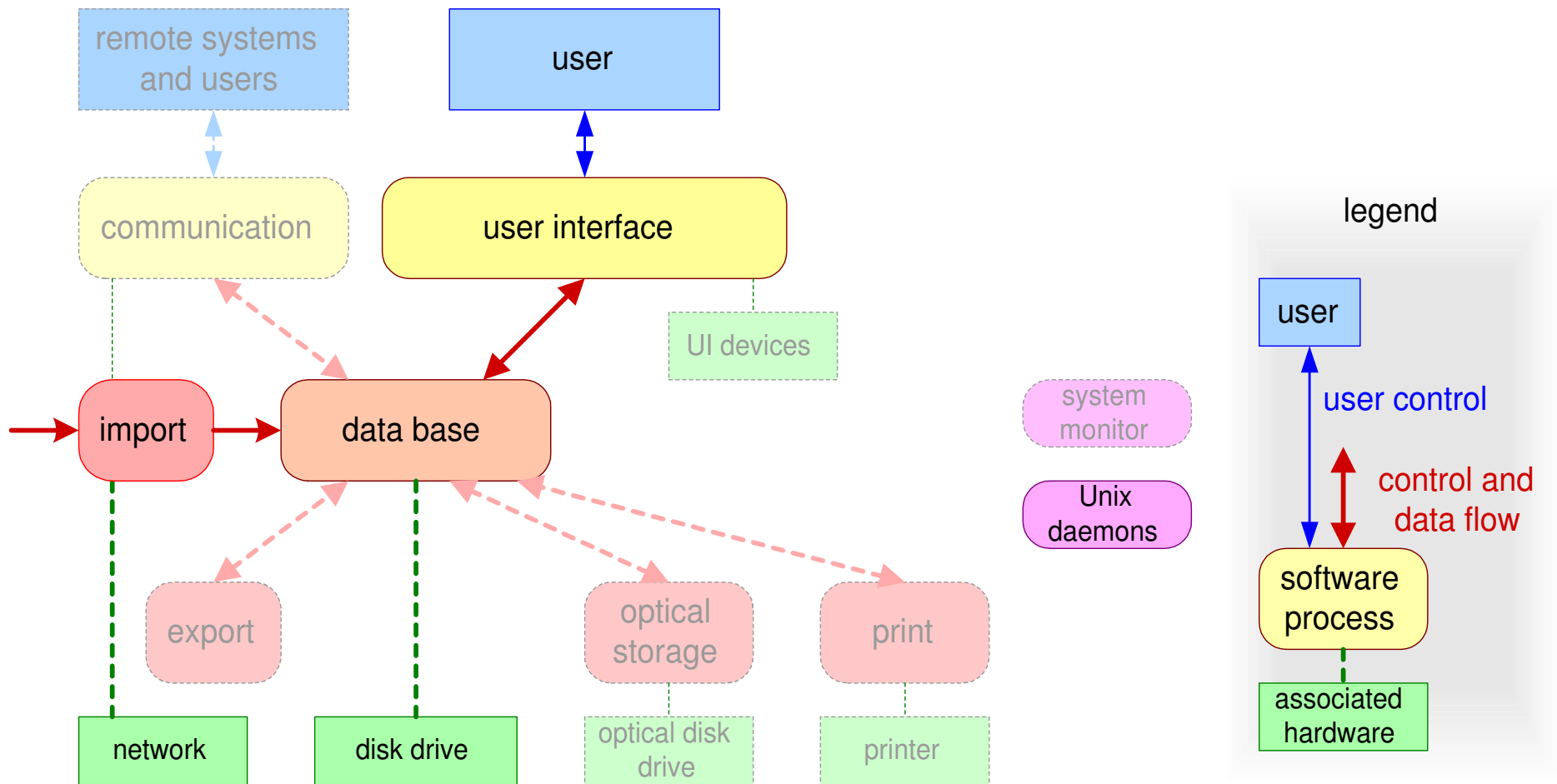
Film:
high resolution
high throughput



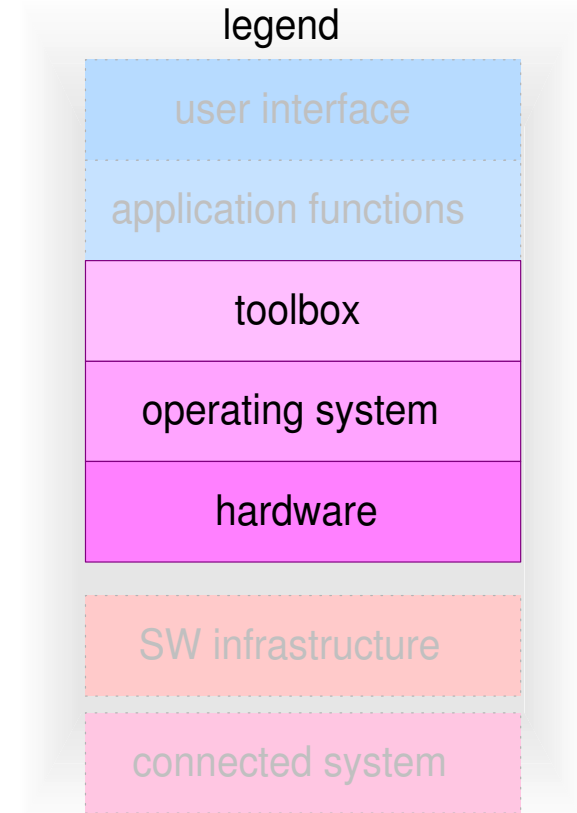
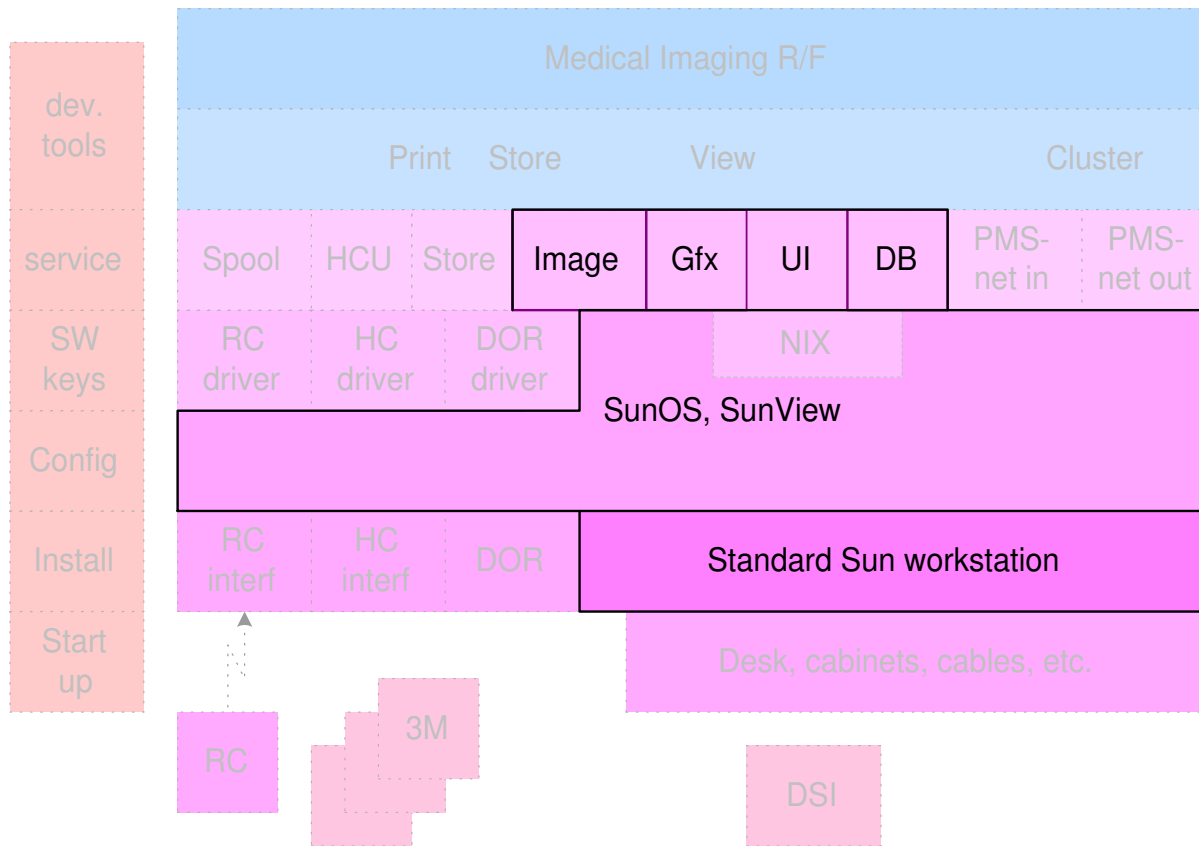
Network:
medium resolution
high throughput



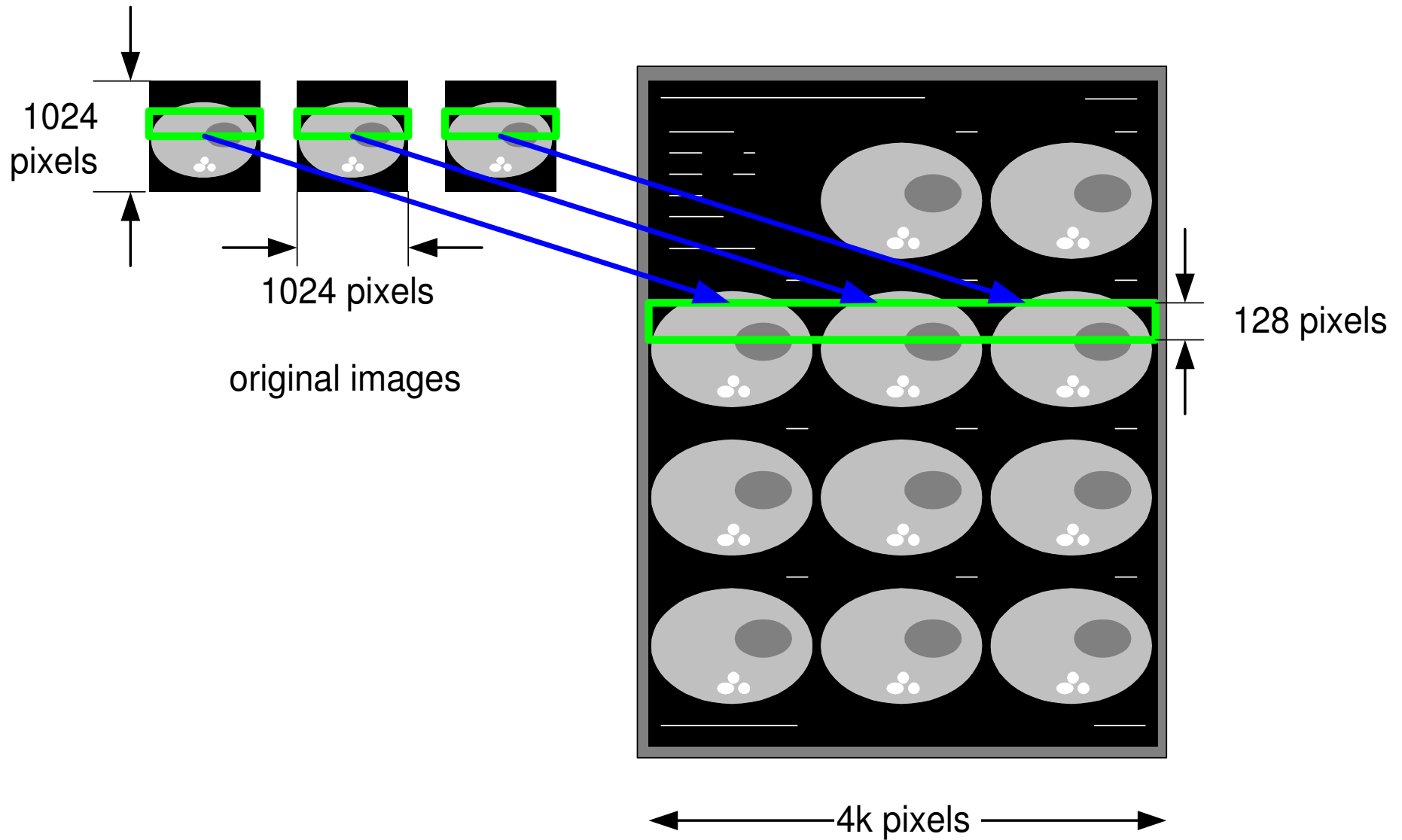
SW Process structure 1991



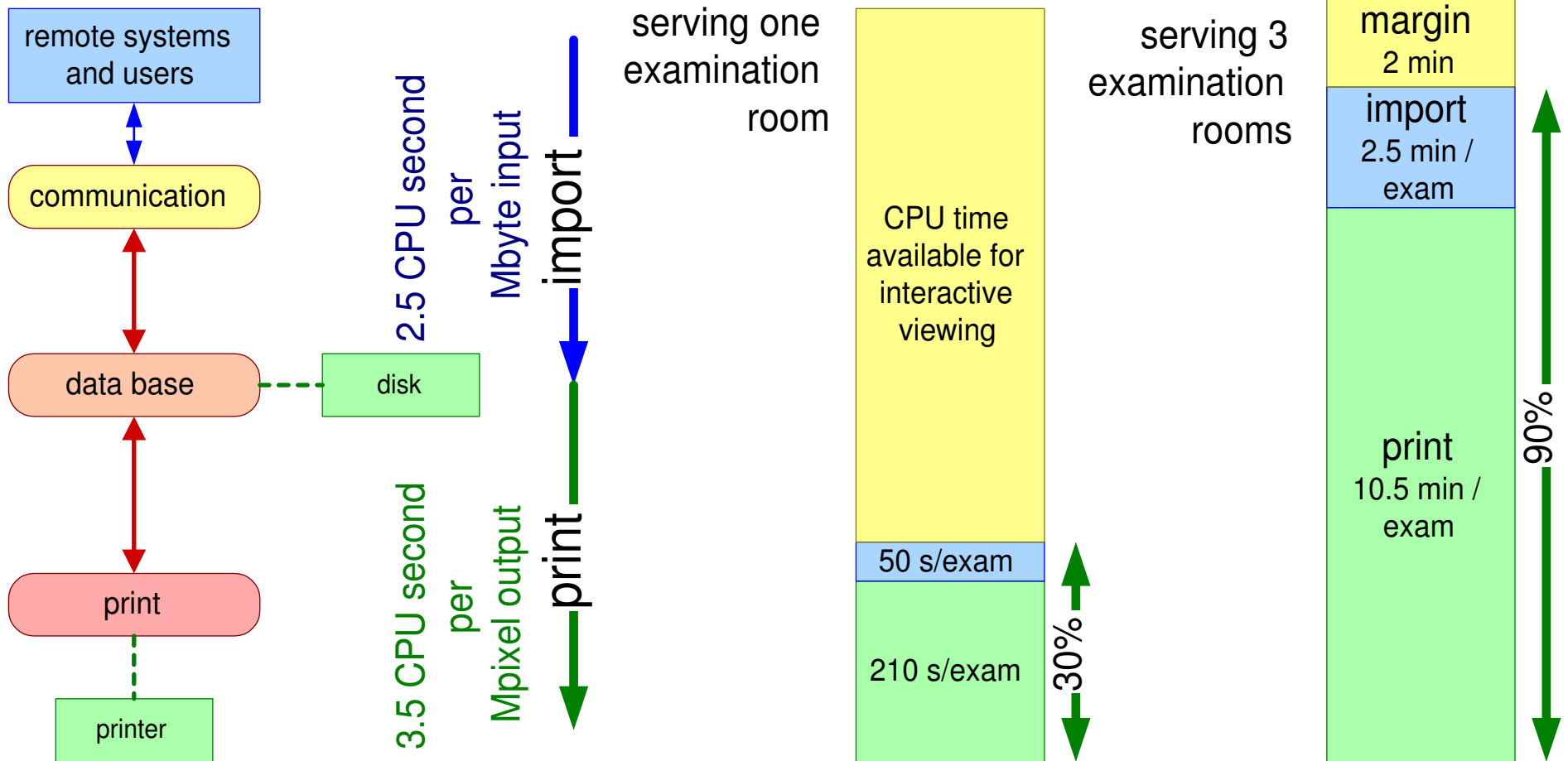
SW layers 1991



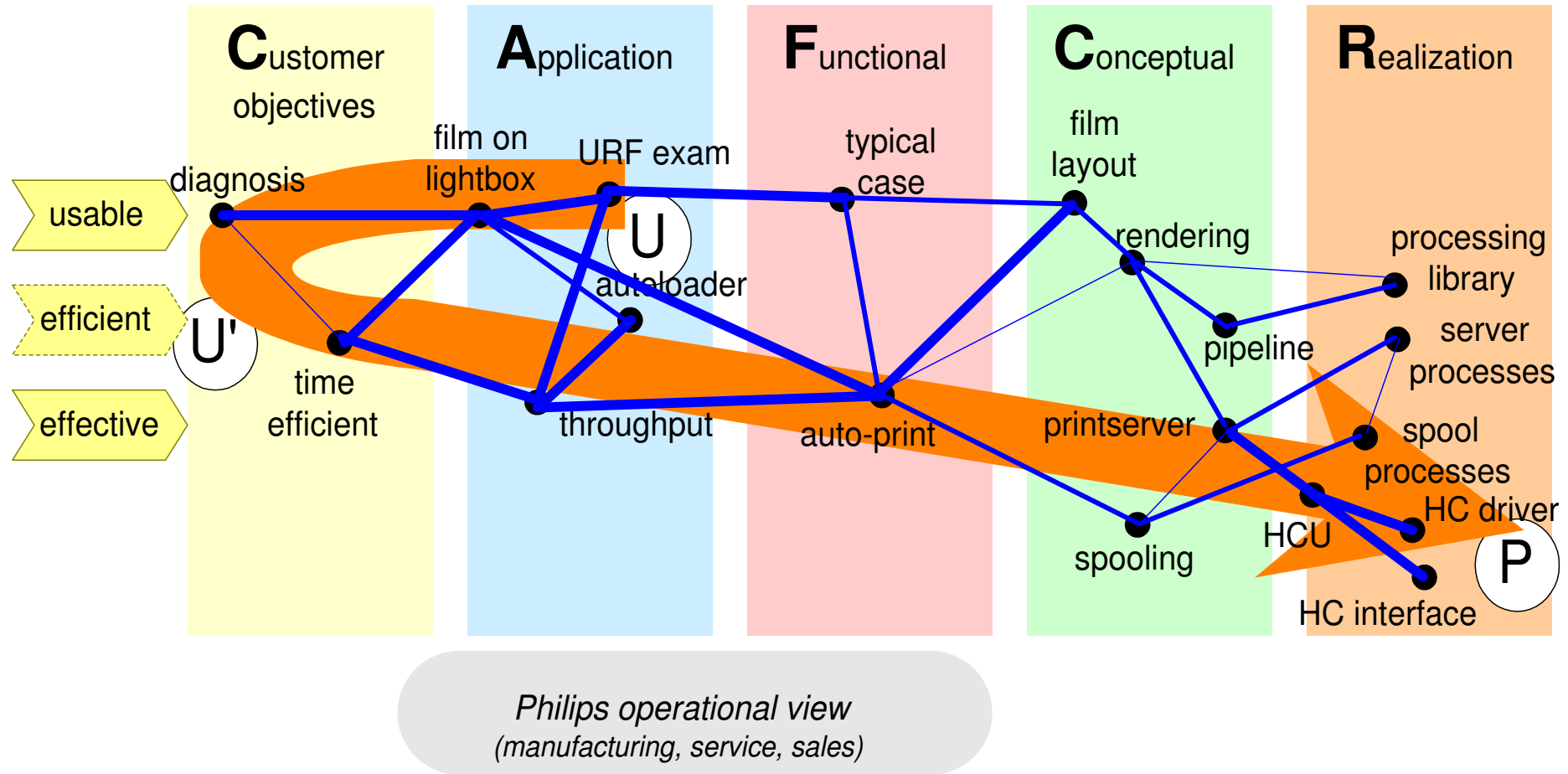
Print server is based on banding



Server CPU load

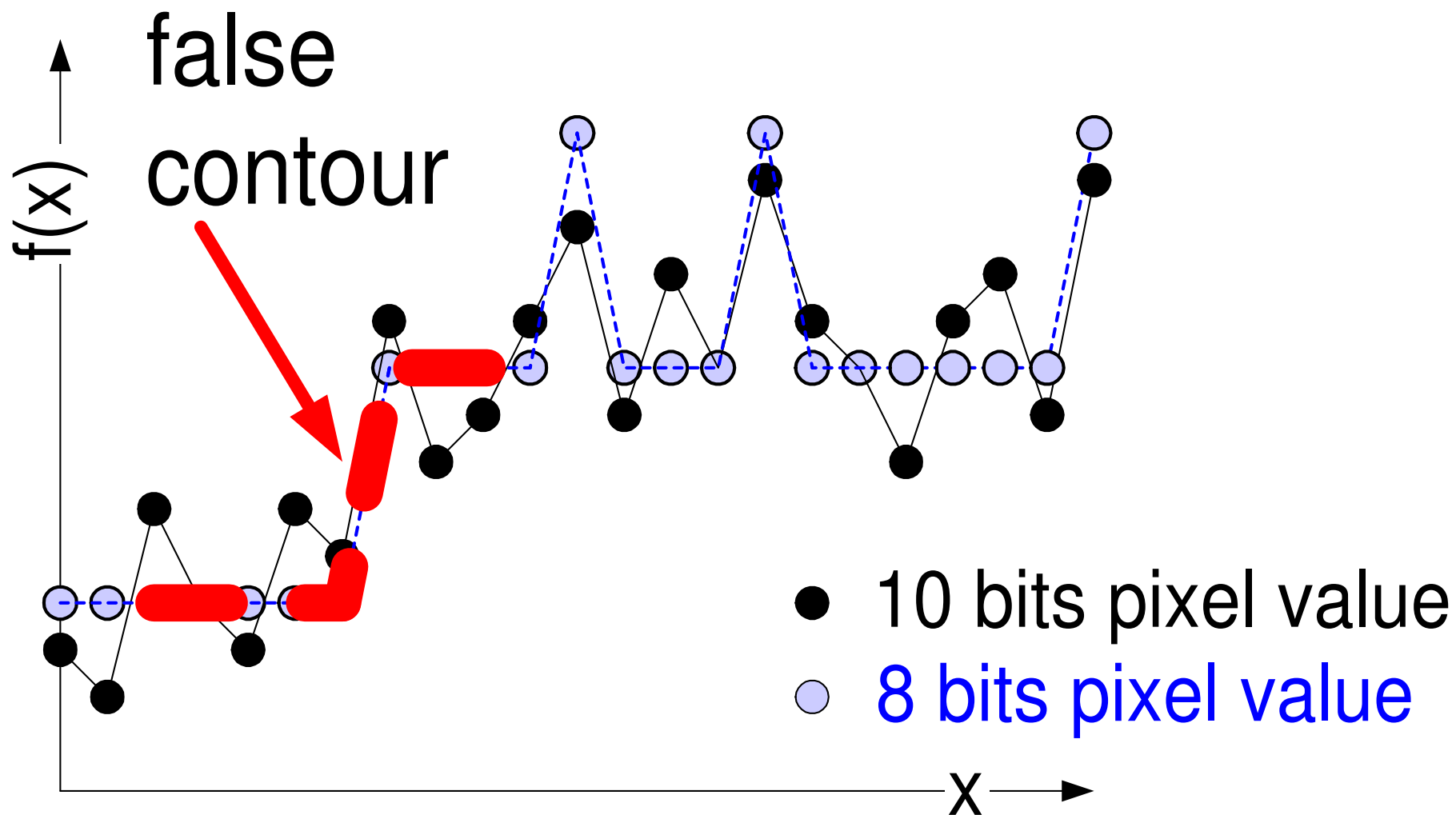


Thread of reasoning; phase 3



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

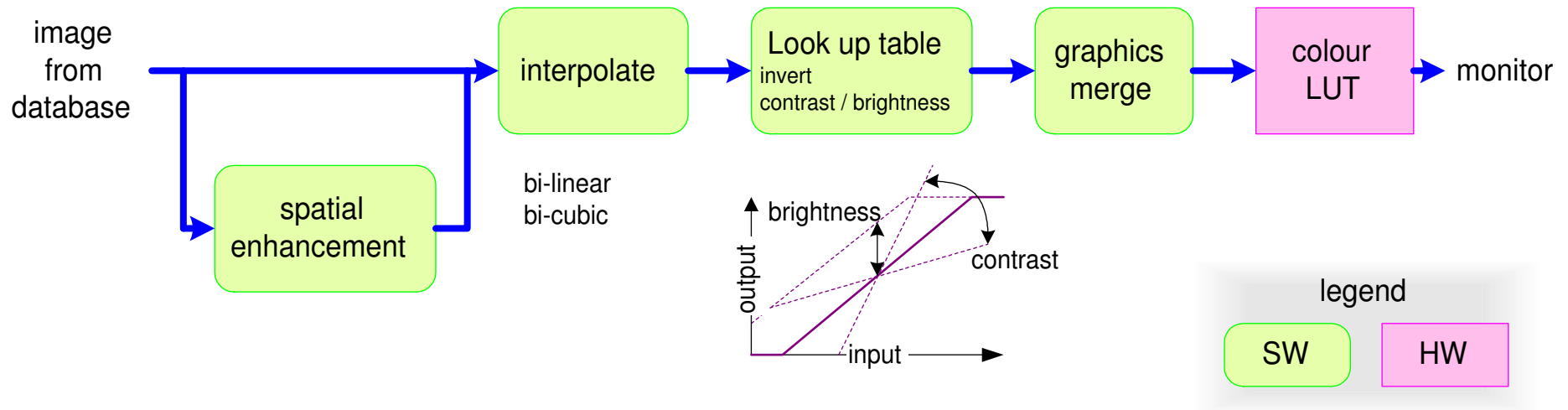
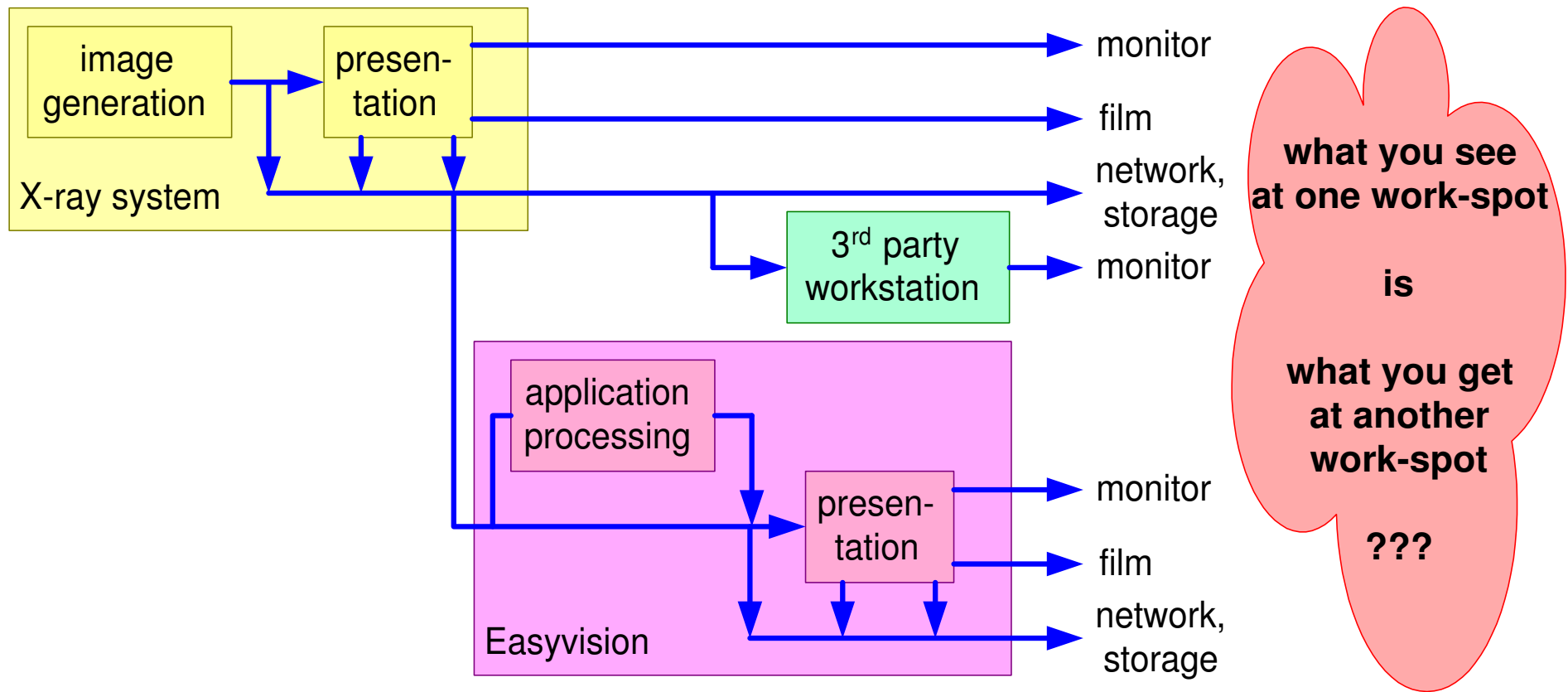
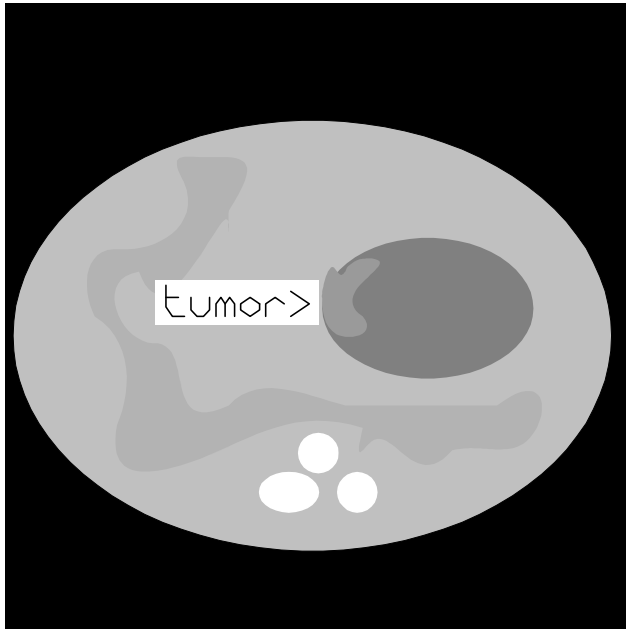


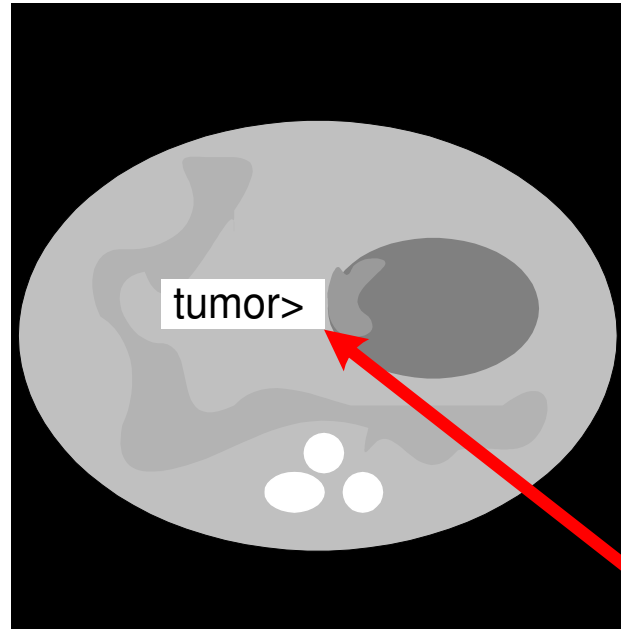
Image Quality expectation WYSIWYG



Safety problem

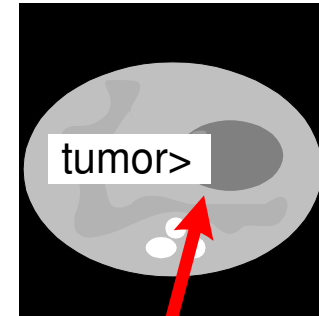


URF monitor output:
fixed size letters at fixed grid

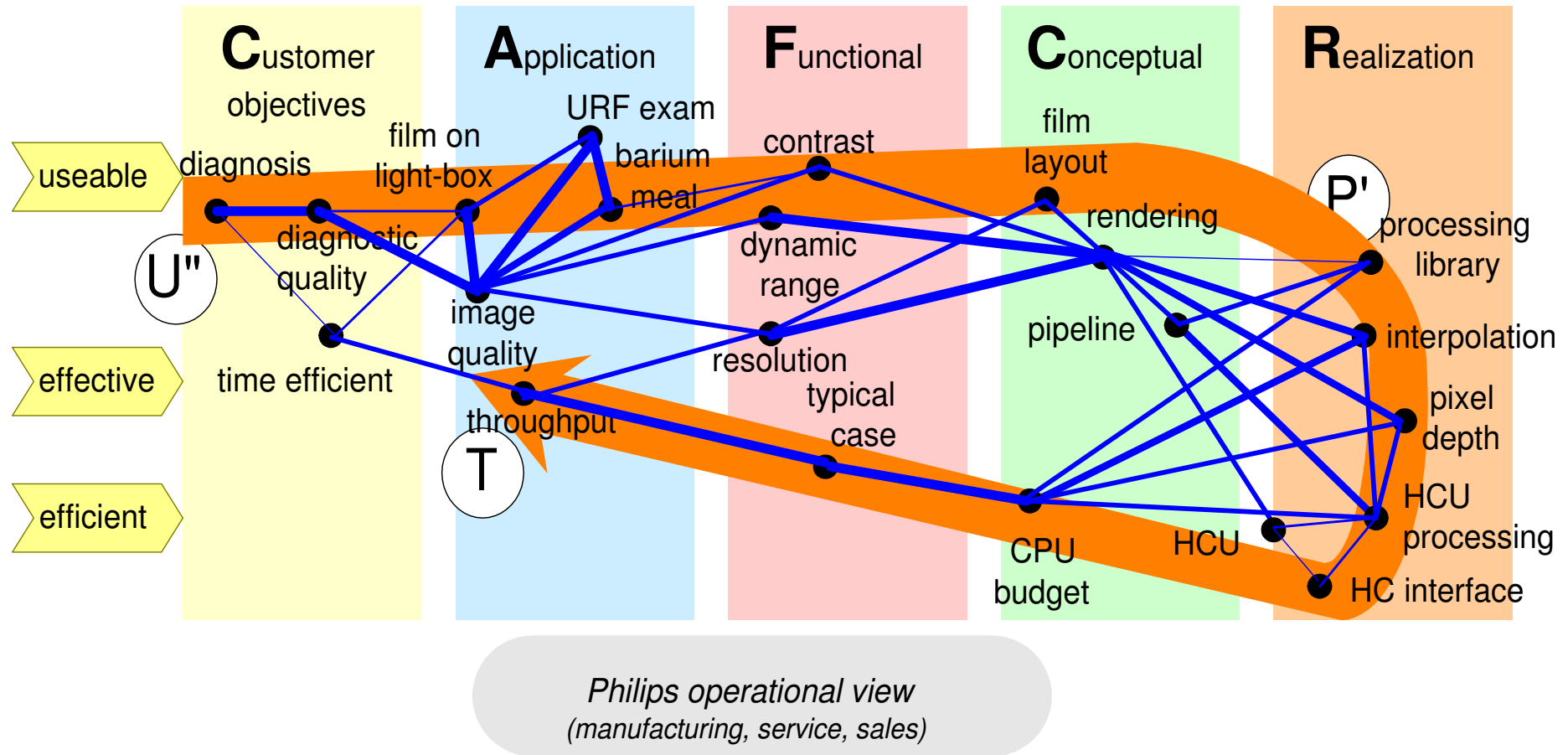


for user readability the font-size was determined "intelligently"; causing a dangerous mismatch between text and image

EV output: scaleable fonts in graphics overlay

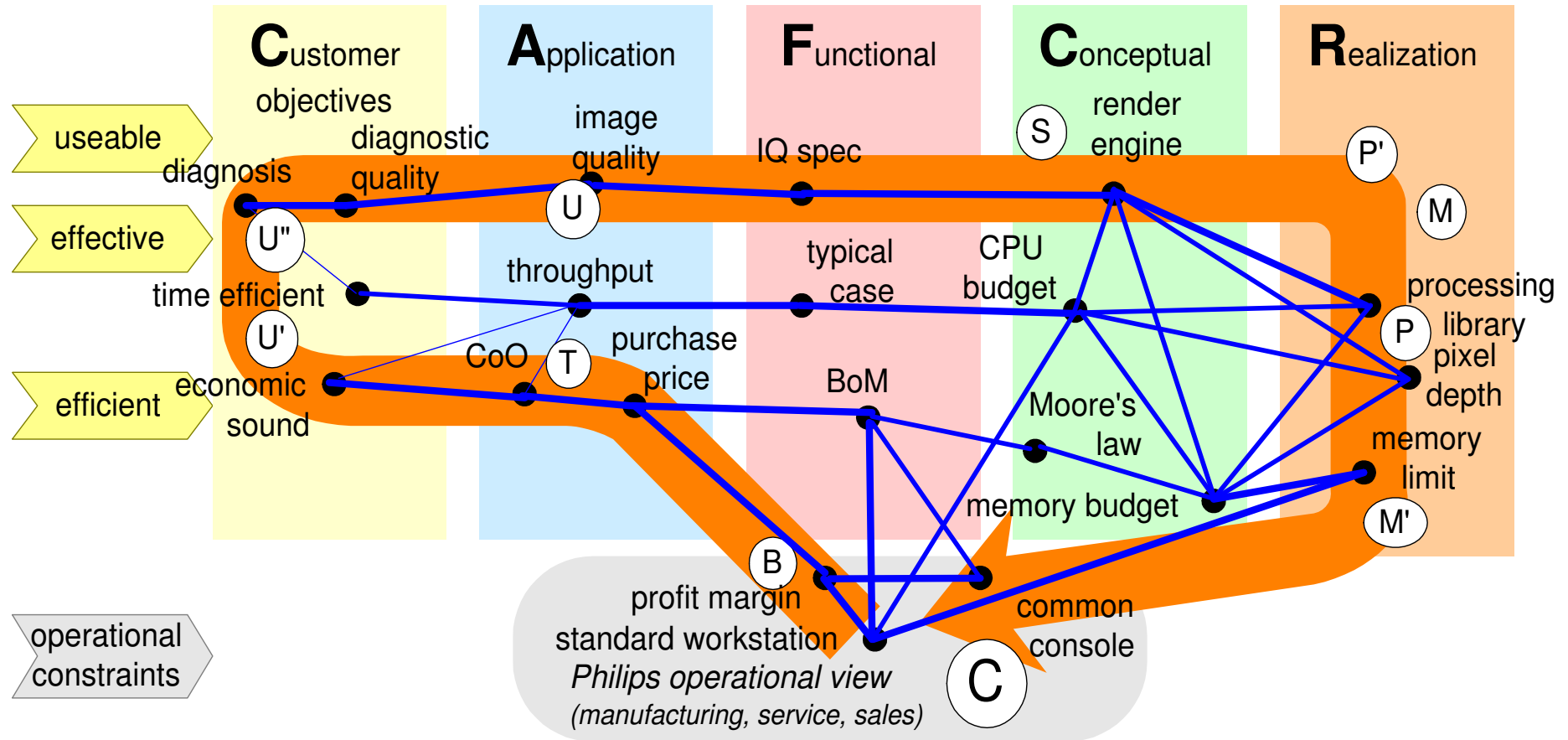


Thread of reasoning; phase 4



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

Thread of reasoning; phase 5



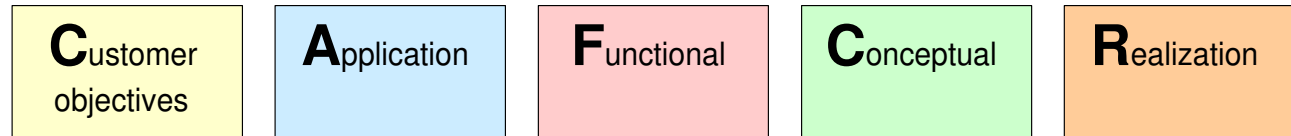
cost revisited in context of clinical needs and realization constraints; note: original threads are significantly simplified

Overview of architecting method

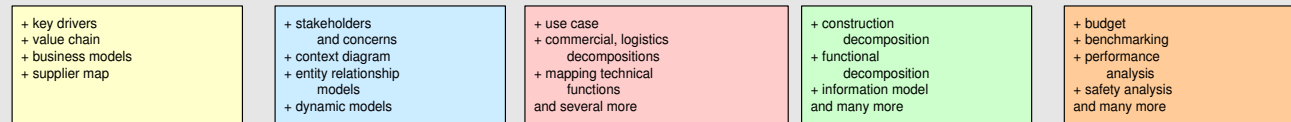
method outline

method visualization

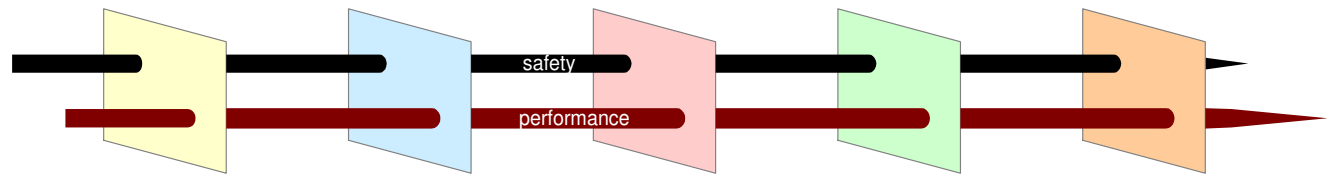
framework



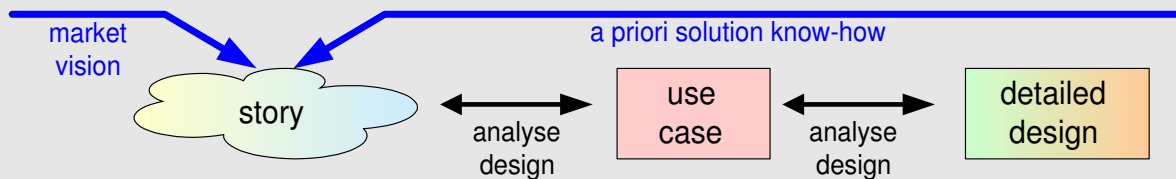
submethods



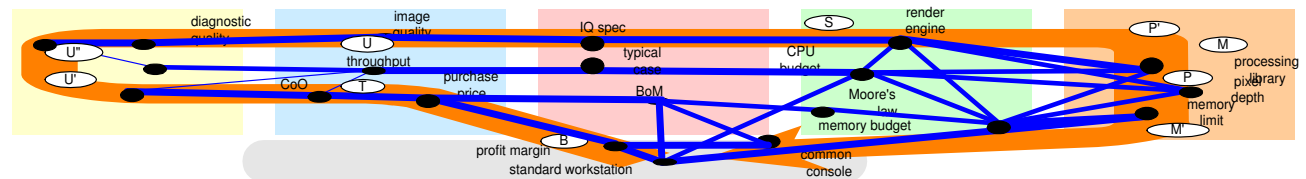
integration via qualities



explore specific details



reasoning



- Make a key driver graph

Use the key driver approach

Take the recommendations into account

- + Key drivers put requirements in broader perspective
- + Discussion creates shared understanding
- ~ The graph needs external feedback
- Are the key drivers really from the customer?
- Are the key drivers *sharp* enough?

Summary Threads of Reasoning

Conclusions

Key Driver graph connects customer objectives to system requirements

Threads of Reasoning connects Customer and Operational Objectives to design and technology choices

The overview is maintained by focusing on valuable, important, critical or sensitive aspect; Look for tensions!

Techniques, Models, Heuristics of this module

Key driver graph

Thread of reasoning

Why, What and How

Tensions

The Boderc project contributed to Key drivers and Threads of Reasoning. Especially the work of *Lou Somers, Peter van den Bosch, Zhaouri Yuan (Océ), Berry van der Wijst (Philips), Adriaan van den Brand (Centric TSolve)* , *Heico Sandee* and *Maurice Heemels* (TU/e, ESI) has been valuable.