

# The System Architect; Meddler or Hero?

by *Gerrit Muller* University of South-Eastern Norway-NISE

e-mail: `gaudisite@gmail.com`

`www.gaudisite.nl`

## Abstract

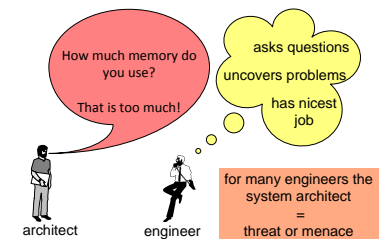
Describes architecting and the task of the architect, with emphasis on bridging the **why**, **what** and **how** of a product. The memory usage of a medical workstation is used as practical illustration.

The introduction of a system architect in an architecture unaware organisation is described. A metamorphosis takes place from a threatening meddler into an appreciated indispensable team member.

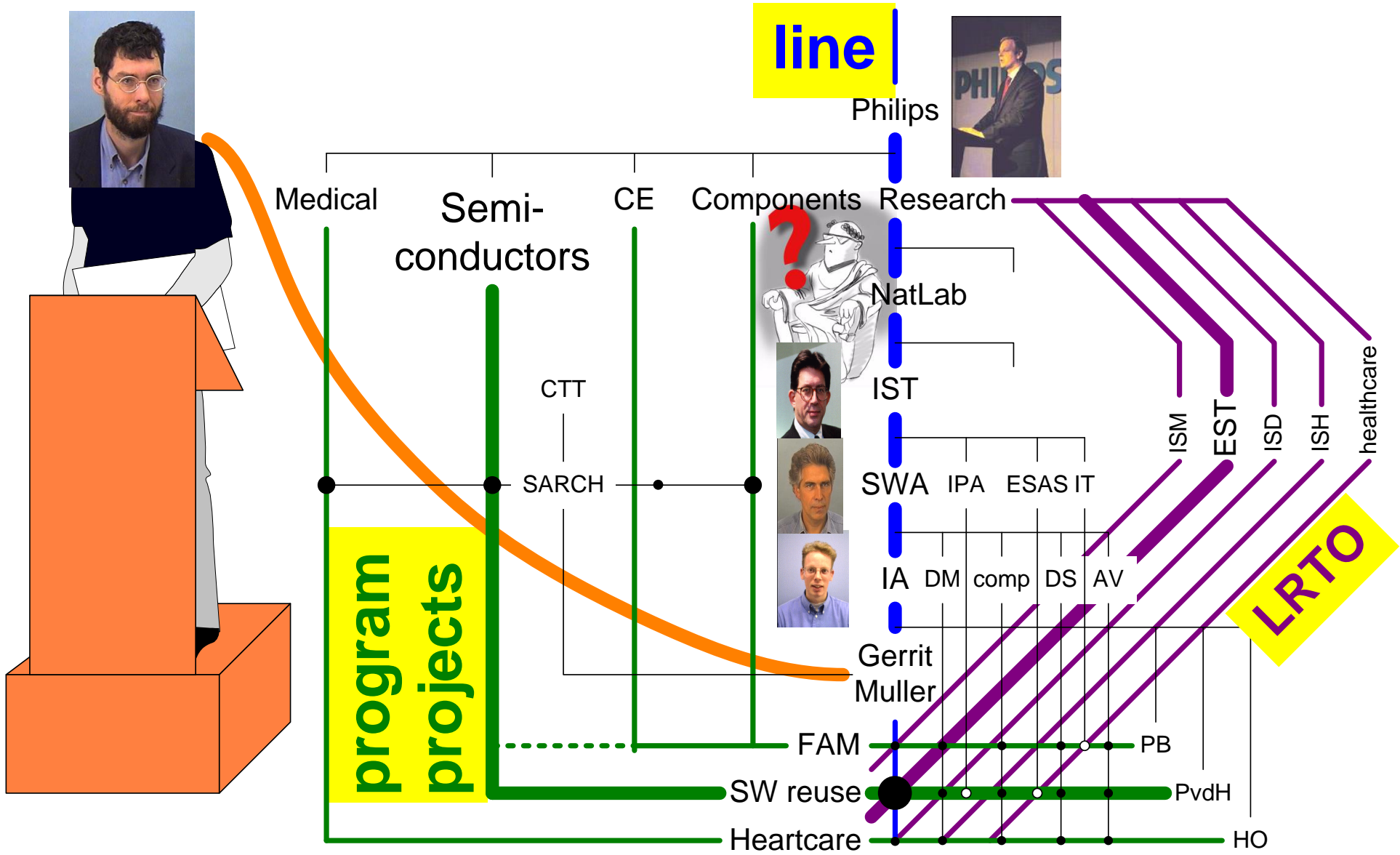
### Distribution

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.

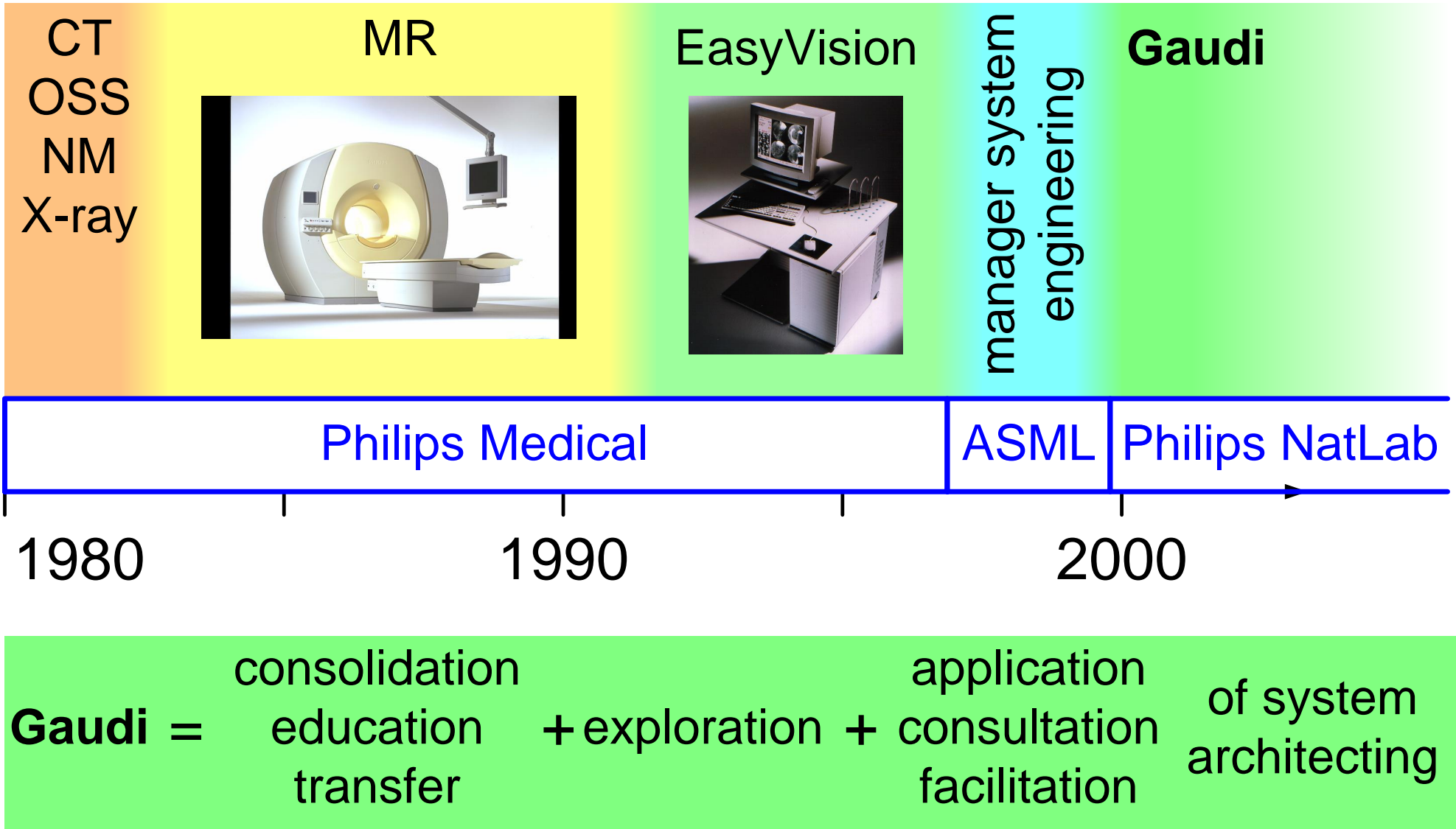
August 16, 2025  
status: finished  
version: 3.1



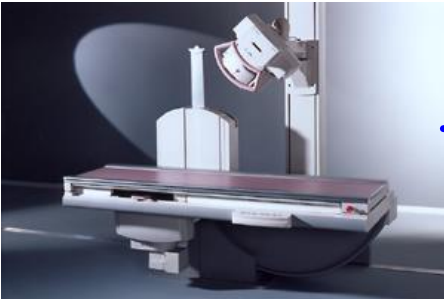
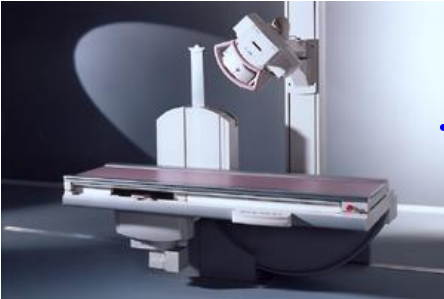
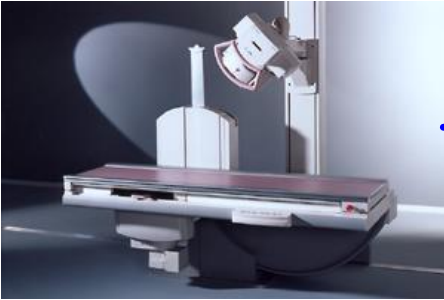
# Where is Gerrit in multi-D organisation space?



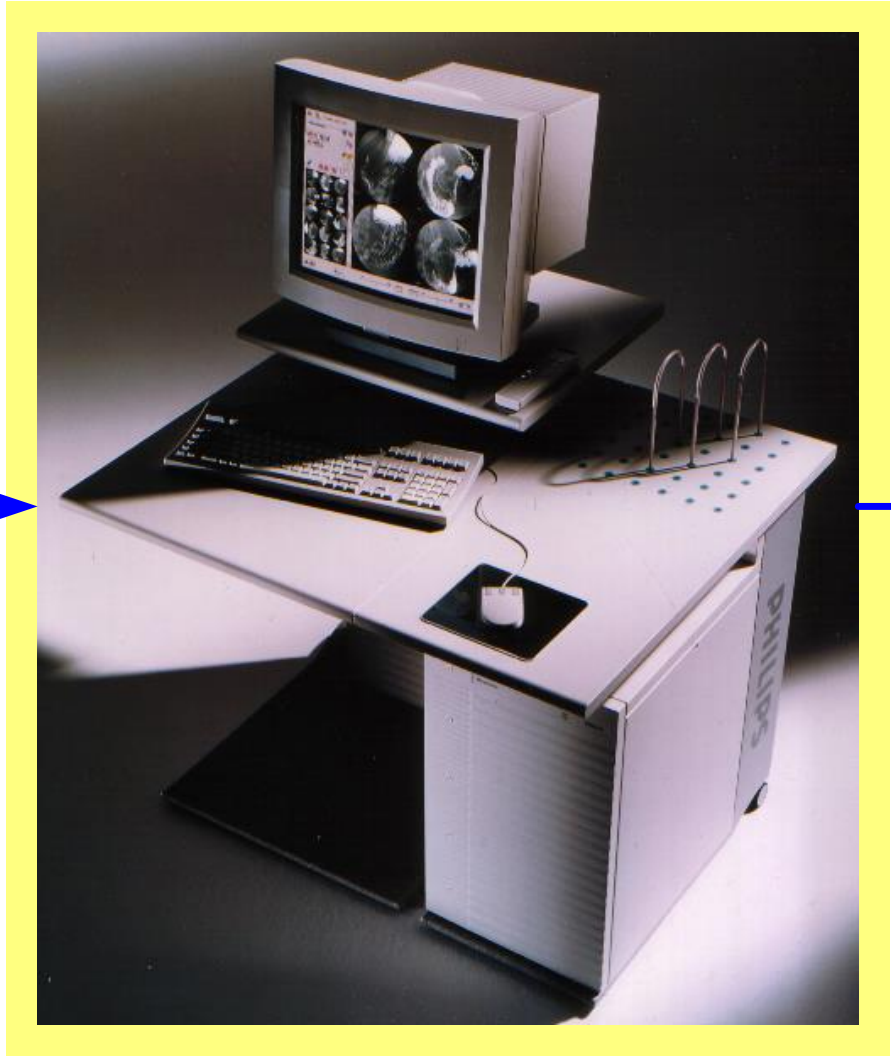
# Who is Gerrit? What is Gaudí ?



# Practical illustration; a medical imaging workstation



URF-systems

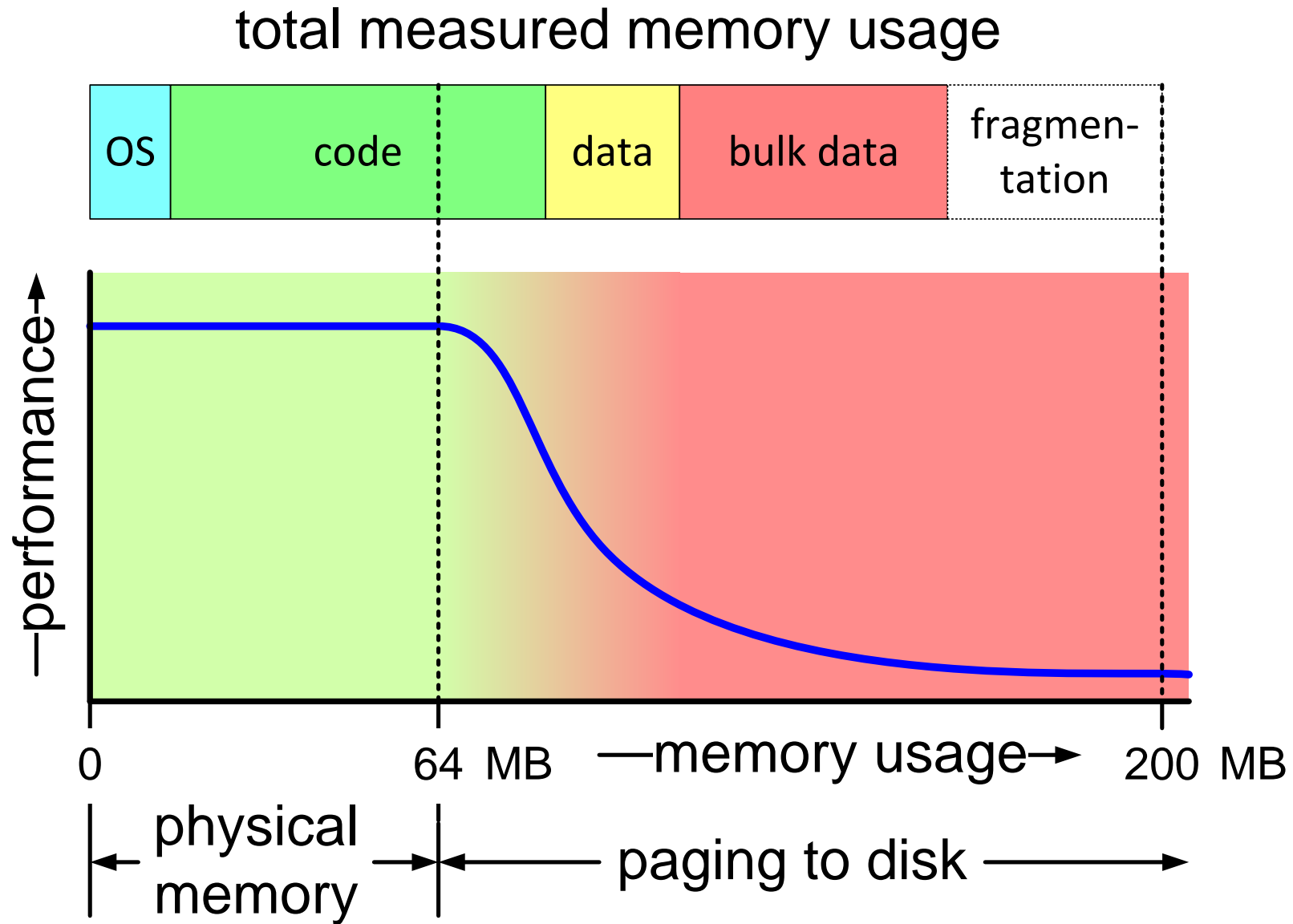


EasyVision: Medical Imaging Workstation

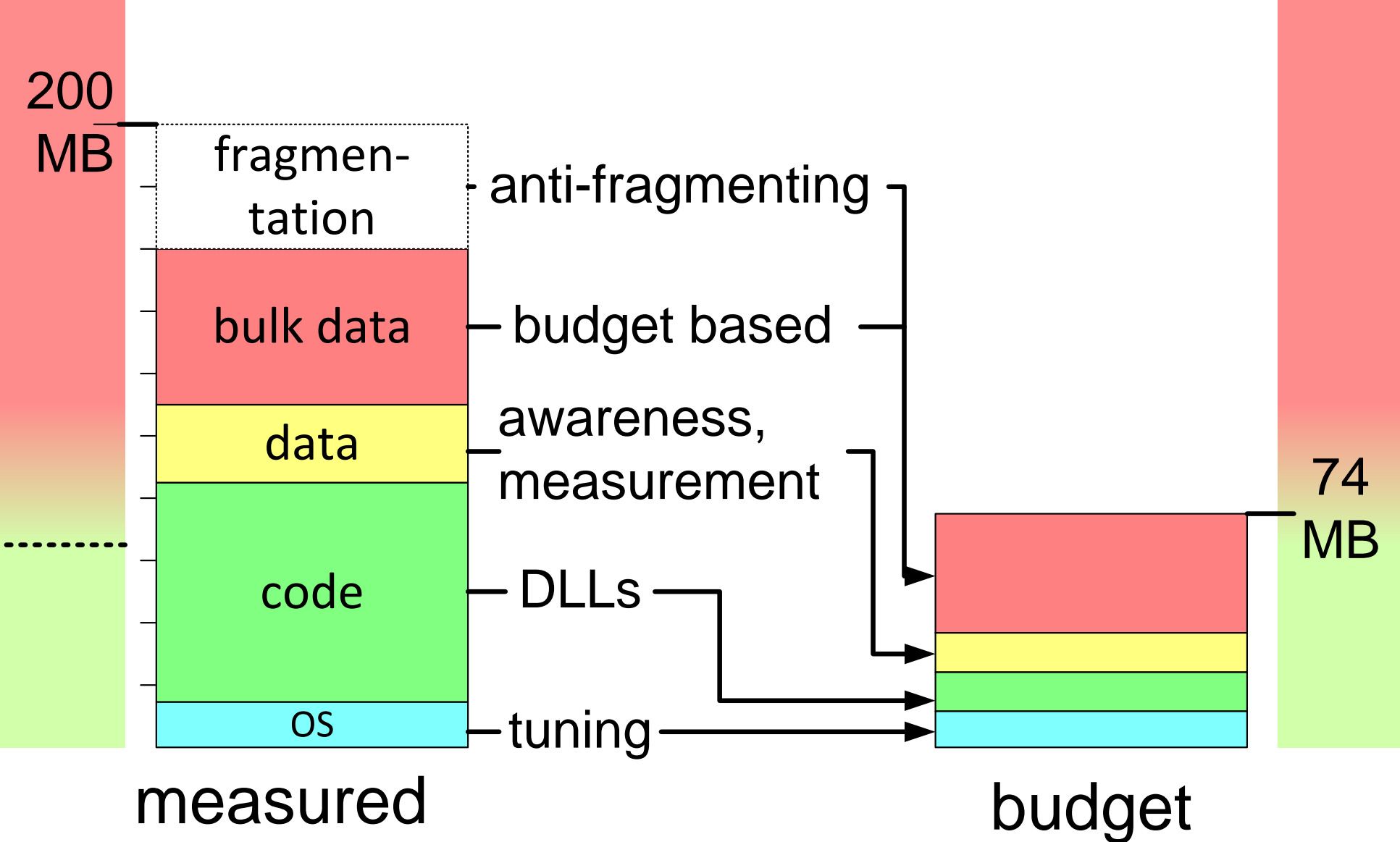


typical clinical image (intestines)

# Problem: unlimited memory consumption (1992)



# Solution: measure and iterative redesign

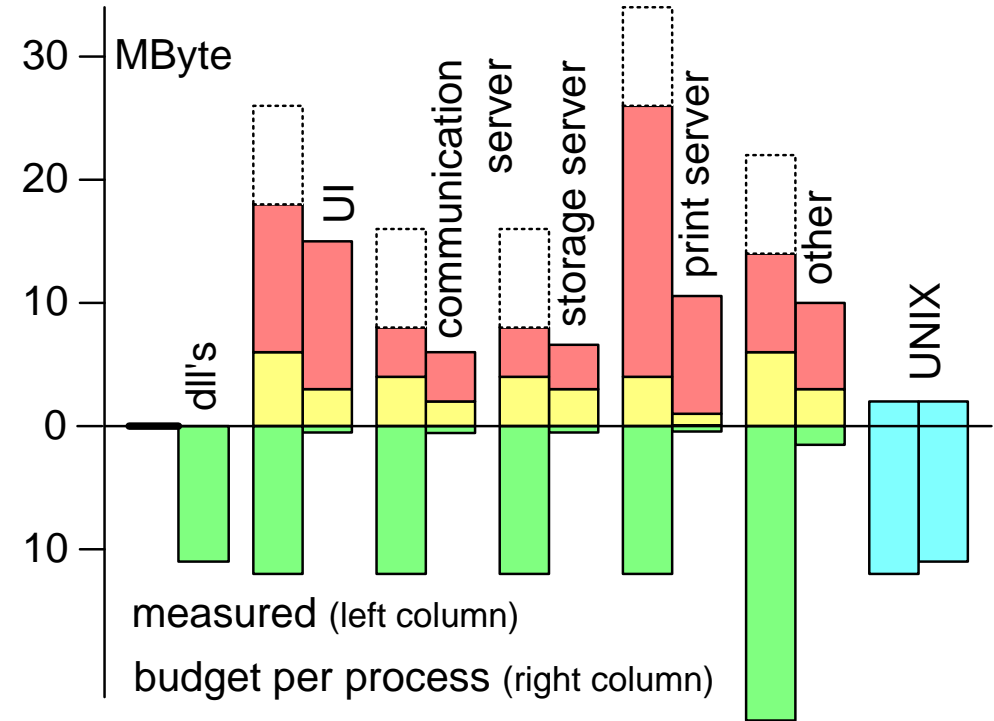


## Budget:

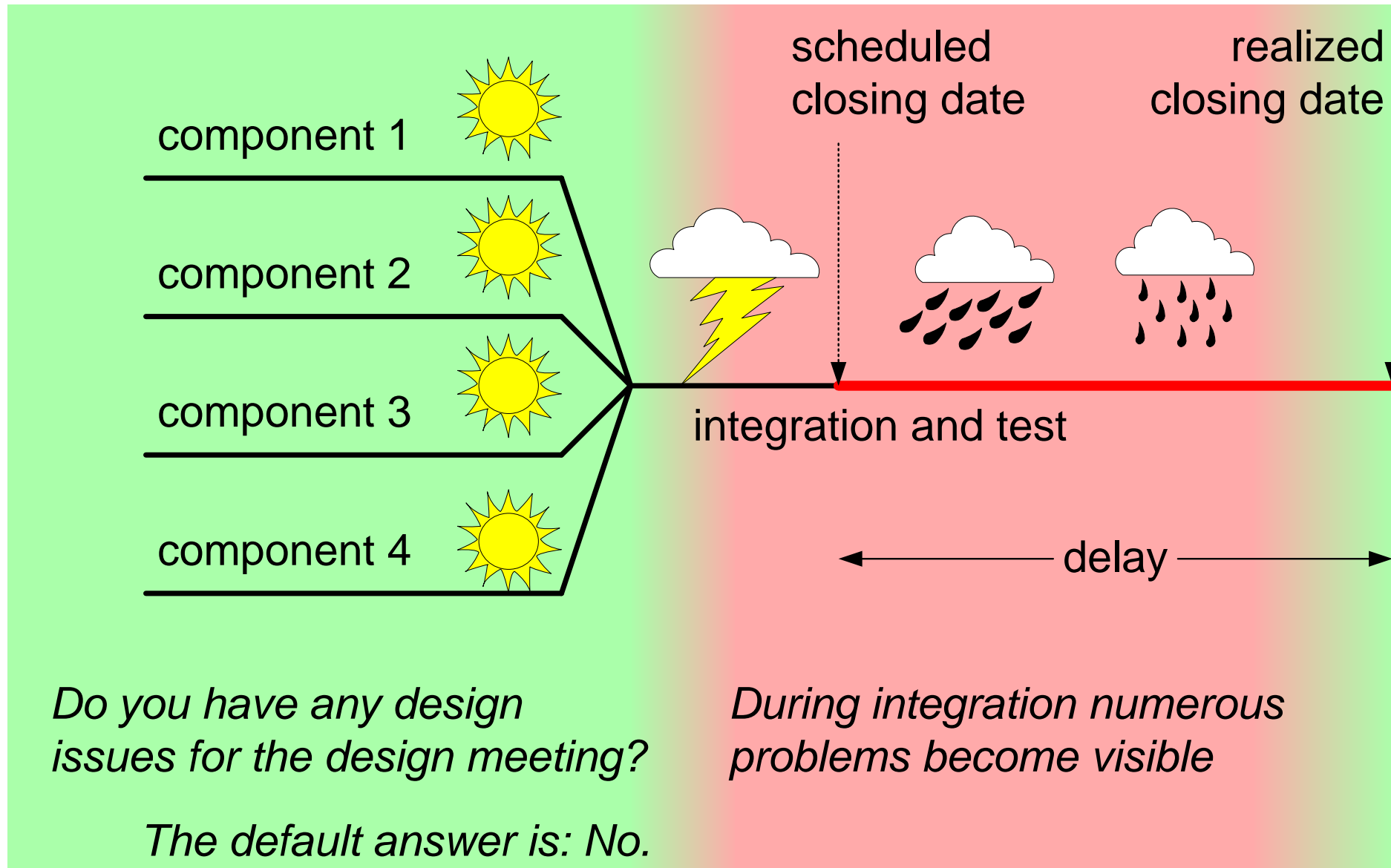
+ measurable

+ fine enough to provide direction

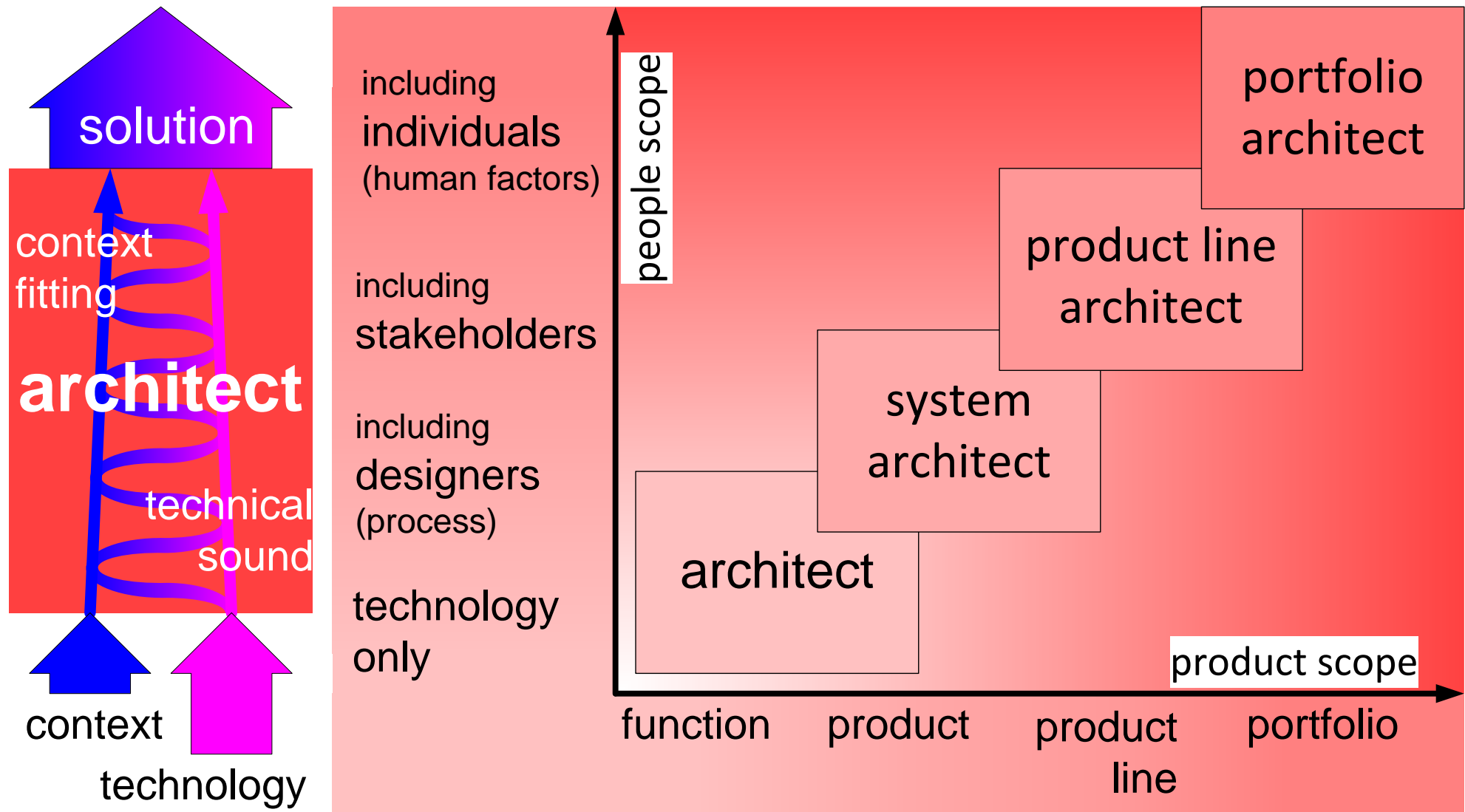
+ coarse enough to be maintainable



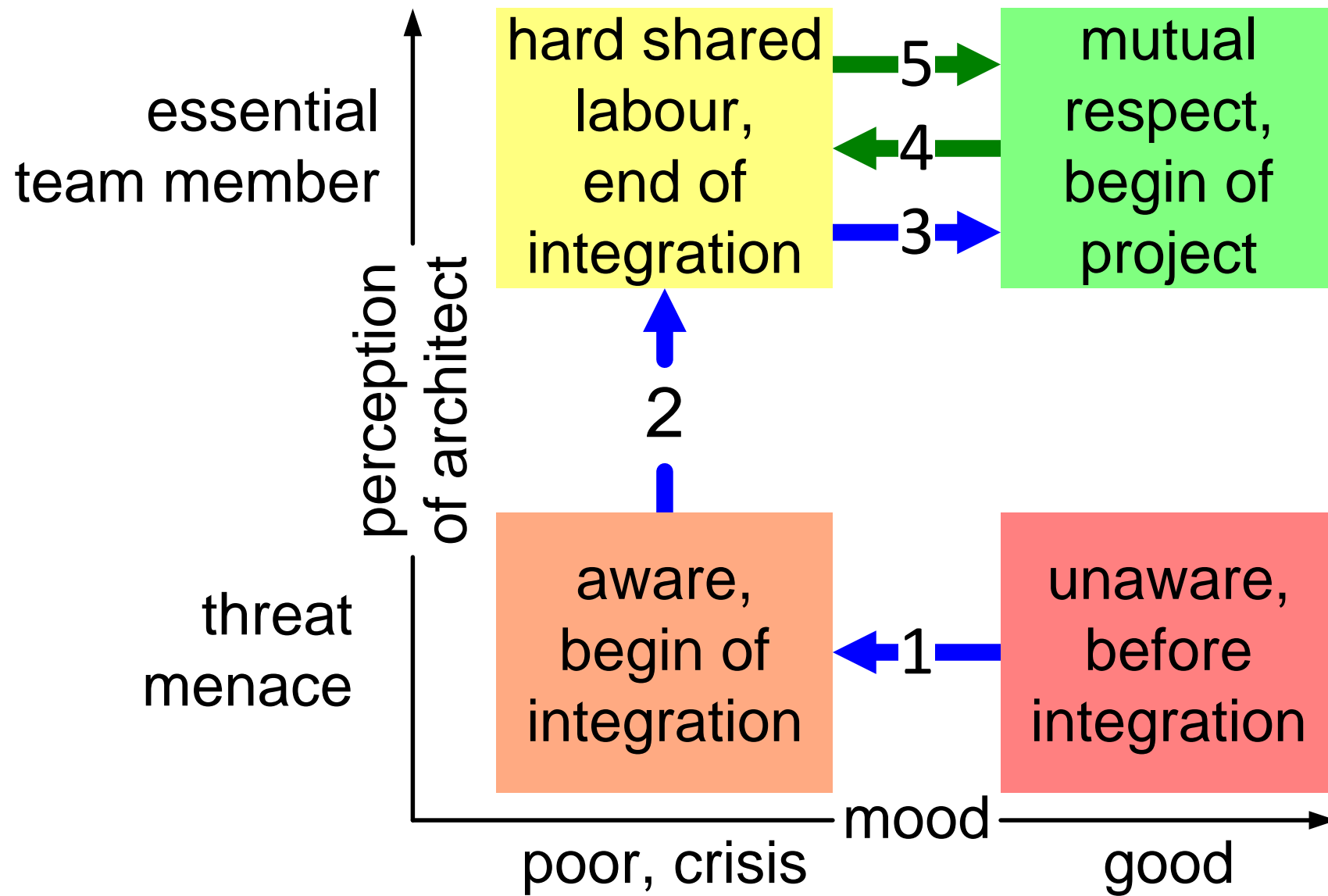
# Integration uncovers hidden problems



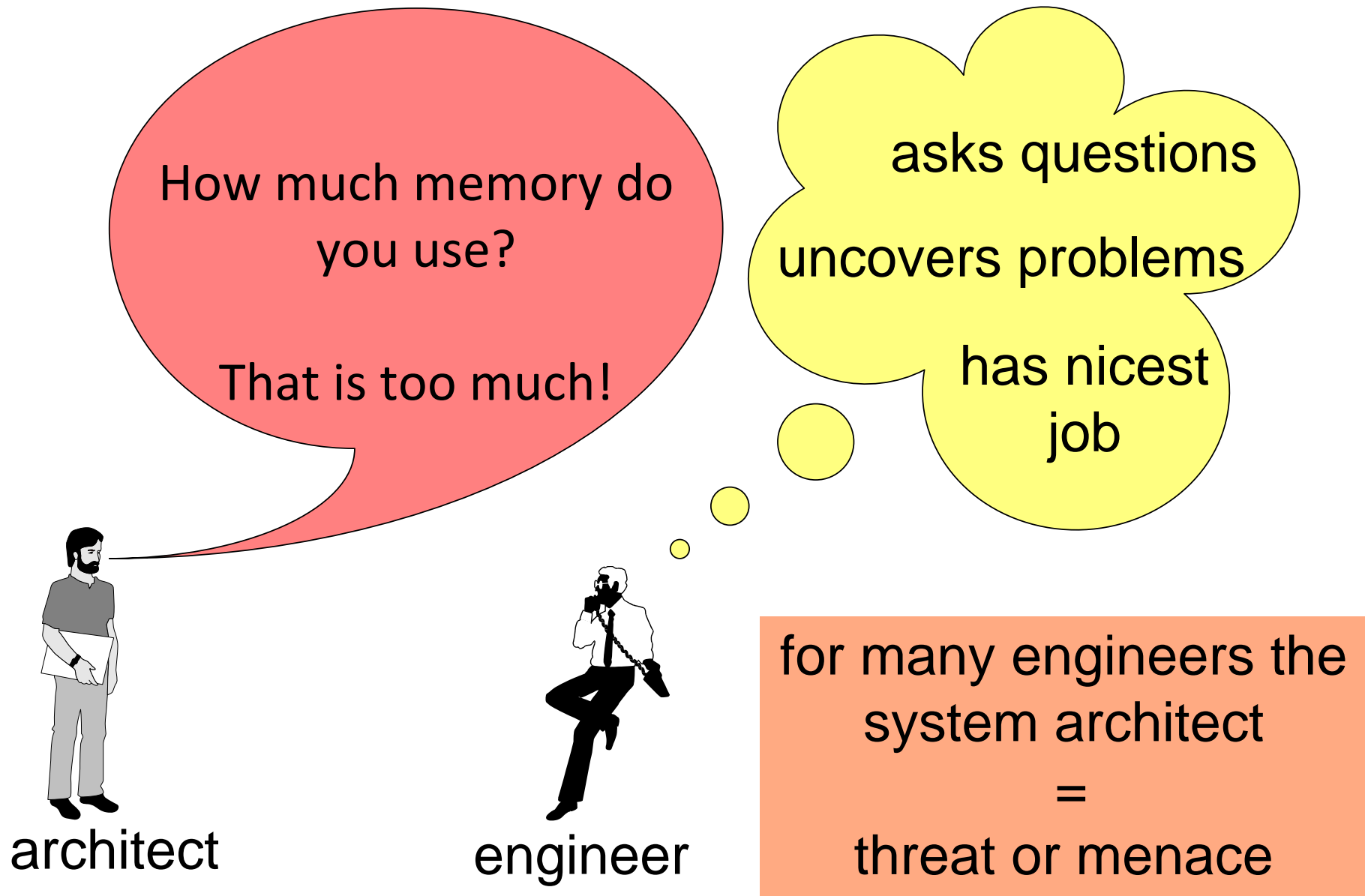
# Architecting scope



# Architecture awareness



# The engineer's perception of the architect



# Understand the customer to drive design choices

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

Product  
**How**

Customer  
**What**

Customer  
**How**

Product  
**What**

**Conceptual**

**Realisation**

**Customer objectives**

**Application**

**Functional**

**Conceptual**

**Realisation**

key drivers:

- \* throughput
- \* diagnostic quality
- \* film saving
- street price 50k\$

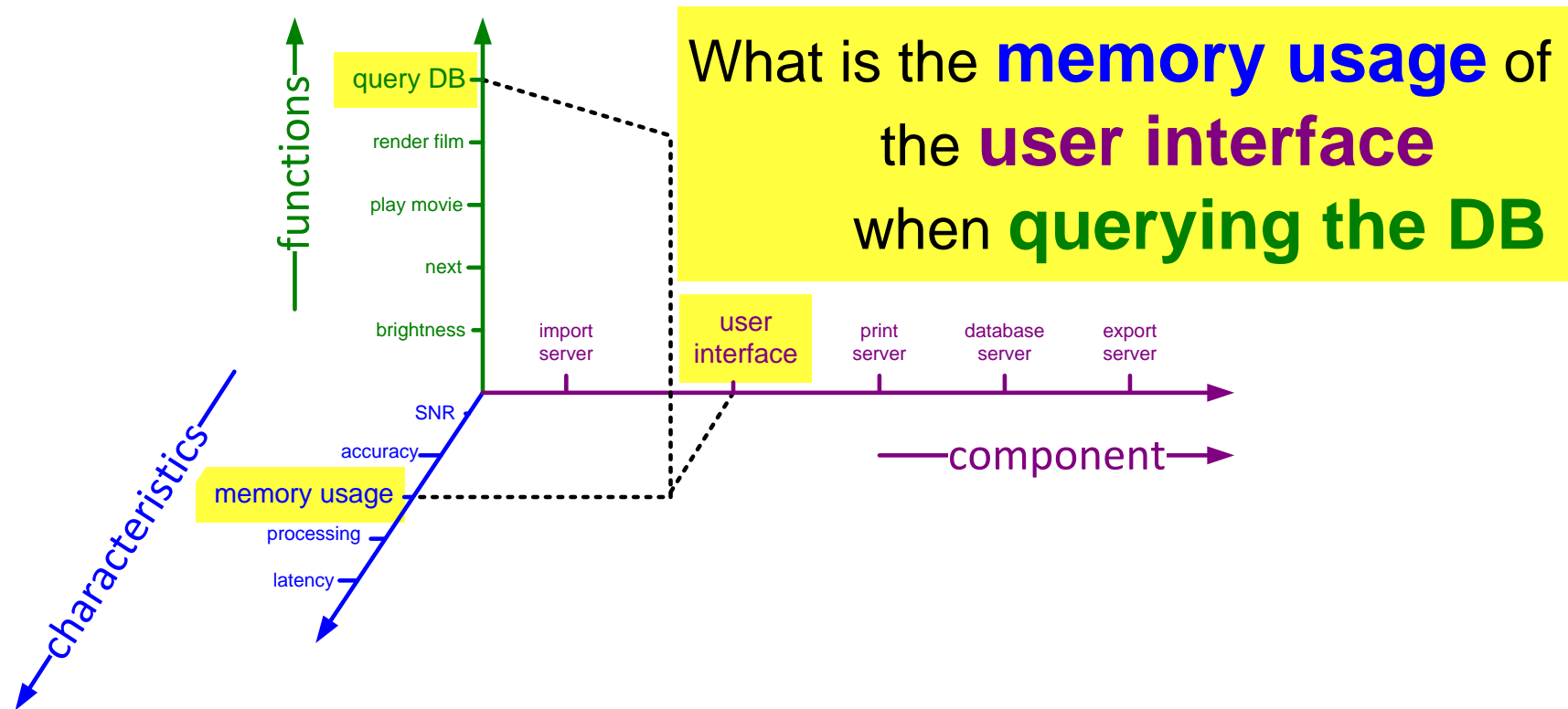
- \* 3 exam rooms
- per exam:
- \* 20 images
- \* auto-print on 3 sheets

decomposition  
anti-fragmentation  
DLL's

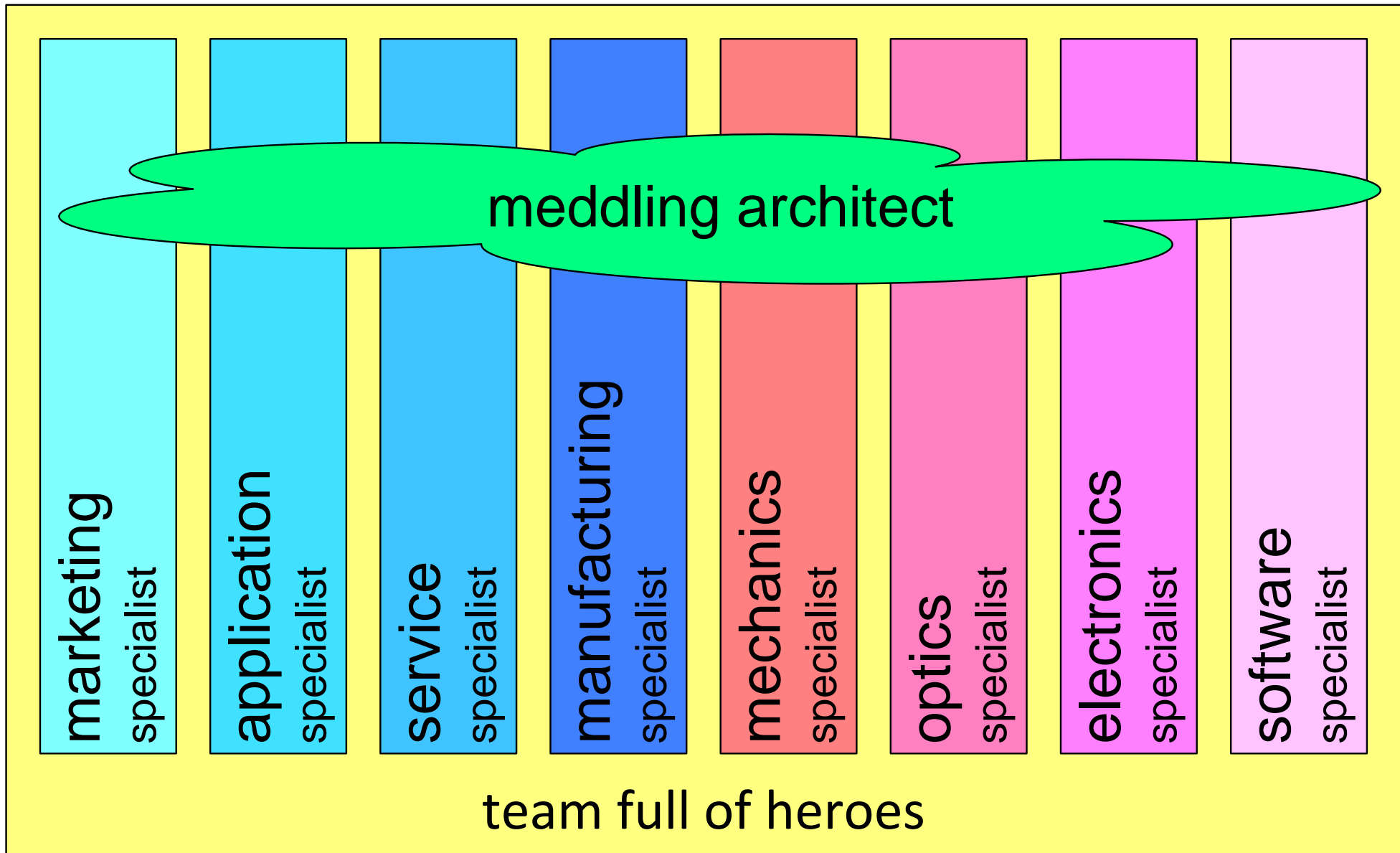
measured  
200 MB  
budget  
74 MB

# Method to detect problems: Question generator

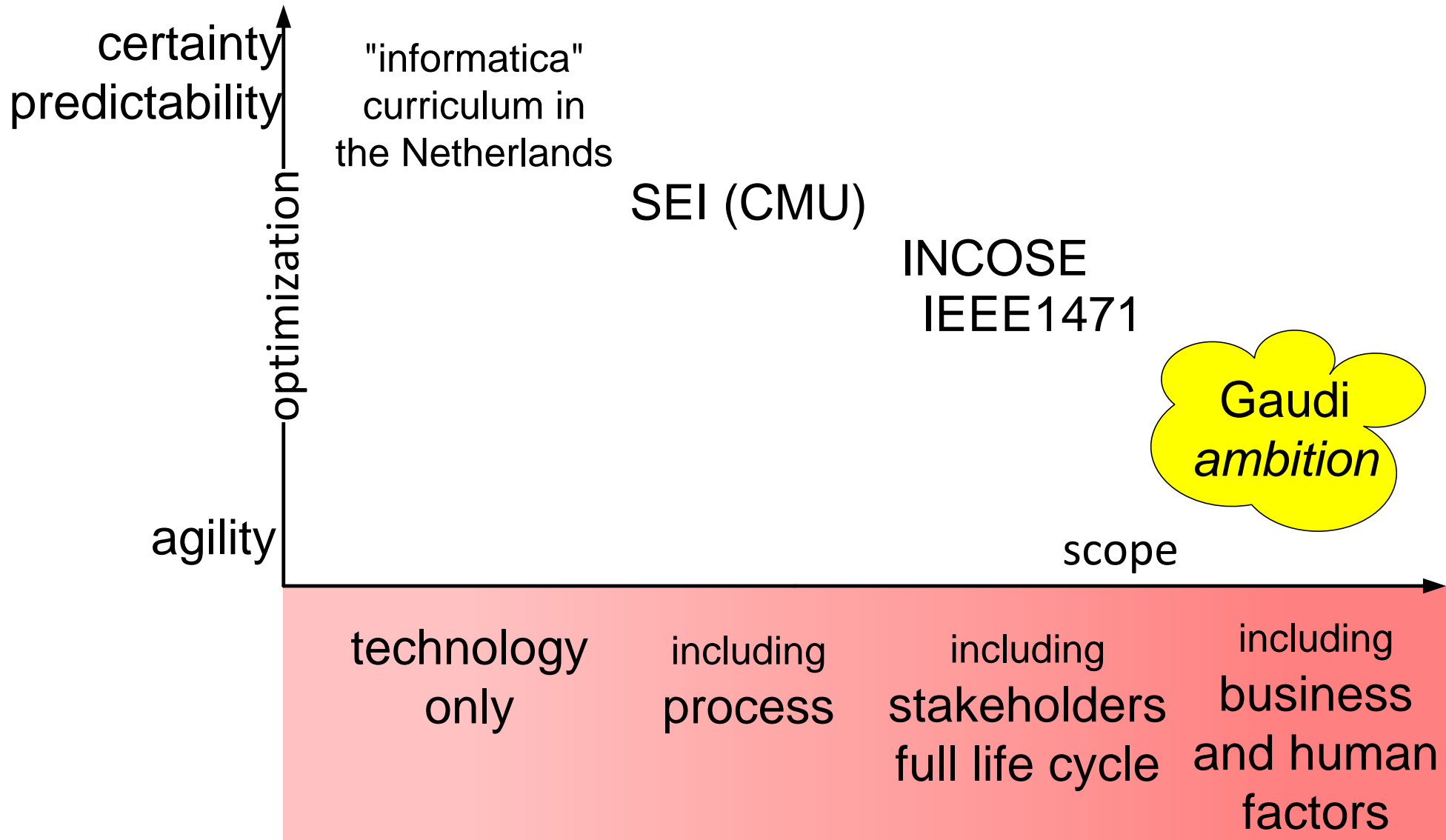
How about the **<characteristic>**  
of the **<component>**  
when performing **<function>**?



# Conclusion



# Benchmarking



# Some Acknowledgements

---

Hammer, Dieter	Gijsbers, Rob	Penners, Maurice	van Gogh, Clemy	van der Sterren, William
Hoogenstraaten, Wil	Huis in 't Veld, Robert	America, Pierre	Wissink, Getty	Soede, Michiel
Mueller, Juergen	Joosten, Jan	Jaspers, Peter	Engelsma, Erwin	van Bommel, Luc
Gieles, Hans	Mulder, Alwin	Versteijlen, Joost	Stut, Wim	Krikhaar, Rene
Eggenhuisen, Huib	de Wit, Paul	Beelen, Peter	Luttikhuisen, Paul	van den Brink, Johan
Kloprogge, Raymond	Poesse, Jan	Blijd, Jarl	Bruin, Jan	Ham, Kees
Engel, Bas	Spaak, Wim	Dijkema, Marcel	Gooren, Huub	Bos, Erik
van Rijnsoever, Bart	Thus, Frank	Roelandt, Werner	den Dekker, Wim	Pijpers, Frank
Driesen, JGH	van Velden, Jeroen	Janson, Paul	van der Laak, Eric	Medema, Jeroen
Schelkers, Raymond	van Venrooy, Roland	Bandakka, Mahesh	Crins, Wim	Kaag, Bjorn
van der Heijden, Jaap	Dobbelsteen, Jan	Ledeboer, Jodie	Heerink, Lex	Giesselman, Timo
Vermeulen, Gerry	de Waal, Klaas	Geron, Nic	Schippers, Alef	Vos, Frans
Stroucken, Marc	Muijen, M	Zieringer, Peter	Schreppers, Jurgen	de Greef, Pierre
Wijnstra, Jan Gerben	Peters, Jo	Beuk, Leo	Deckers, Robert	Fischer, Stefan
Algra, Egbert	van Bommel, Pieter Jan	Koolen, Gertjan	van Balen, Auke	Pu, Xuemei
Derks, Frans	Thijssen, Henk	Koushik, Sudeendra	Huiban, Cristian	Boom, Sjirk
Faber, Albert	Boot, Gert Jan	Milosevski, Vlatko	van Loon, Gerard	ten Pierick, Henk
Aarts, Peter	Vullings, Erik	van den Broek, Ger	van den Heuvel, Patrick	Stroucken, Louis
Watabe, Yasuma	Vermeer, Ad	de Kruif, Peter	Lobo, Lincoln	Young Tai Liu
van Ouwerkerk, H	Peeters, Bob	Daenen, Steven	van de Meulenhof, Dennis	van der Steen, Marcel
Huijnen, Ton	Obbink, Henk	Soepenber, Gerben	Houtepen, Rob	Siereveld, Ad
Gonot, Mathieu	Bas, Han	Bingley, Peter	Hofsink, Robert	van Bakel, Gerian
van Splunter, Andre	Rankers, Adrian	Follon, Roel	Buurman, Hans	Engbers, Rene
van Rooijen, Joost	Akiwumi-Assani, Olu	Elzinga, Onno	Zondag, Eddy	van Wetten, Frank
Verberkt, Mark	Gopalan, Rajaraman	van den Donker, Piet	Veldmans, Ferdinand	Stevens, Frank
van der Linden, Wim	Misdorn, Han	Zwaans, Ben	Merkus, Paul	Wubben, Rob
Patrzalek, Jarek	Schatorie, John	Harmsze, Françoise	van Tuijl, Frank	Schellingerhout, Nico
Vergoossen, Theo	Boer, Richard	Jansen, Tom	Wouters, Kees	Vugts, John

[http://nlww.natlab.research.philips.com:8080/  
research/swa\\_group/muller/](http://nlww.natlab.research.philips.com:8080/research/swa_group/muller/)

*containing:*

more than 30 recent articles and or presentations  
course material SARCH, ESA stakeholders, OOTI req eng

*links on this homepage:*

this presentation

[MeddlerOrSaviorSlides.pdf](#)

annotated text

[MeddlerOrSaviorPaper.pdf](#)

background information Medical Imaging

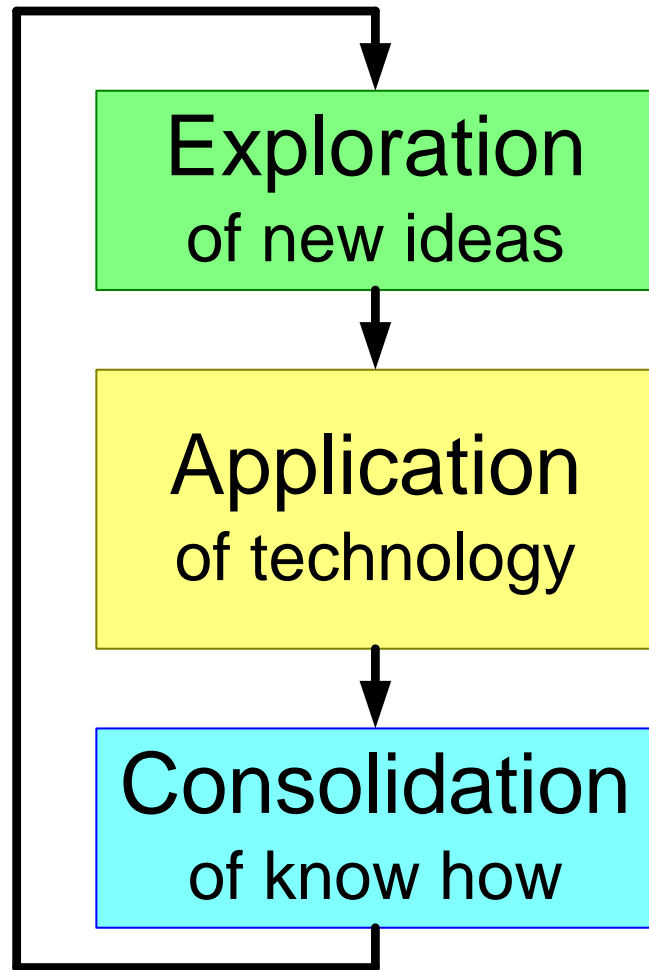
[MedicalImagingPaper.pdf](#)

original documentation Medical Imaging

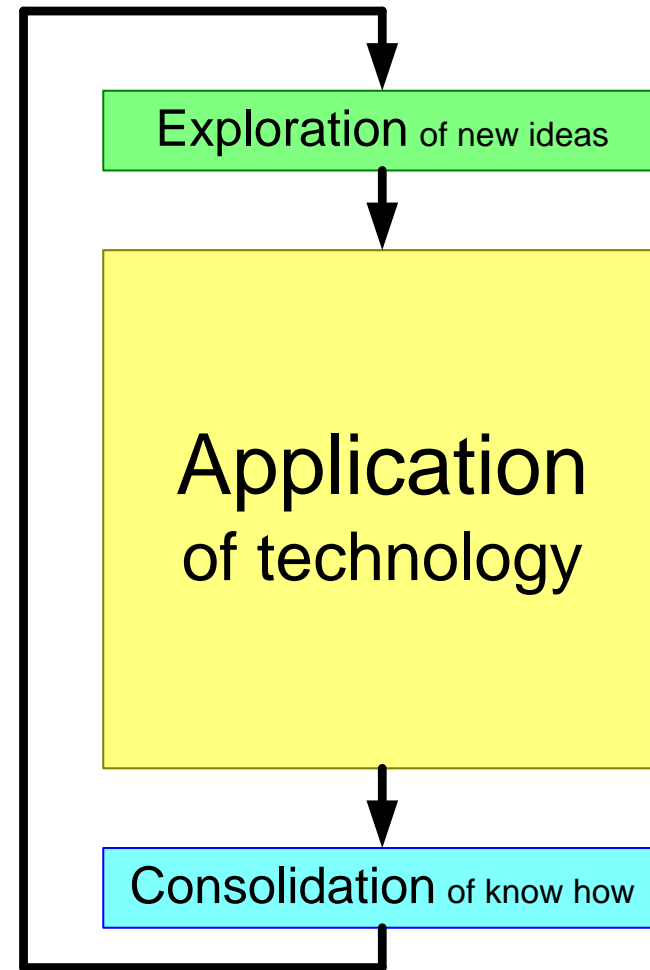
[oldPresentations.html](#)

## Answers to frequently asked questions

# Architecture and research? The technology management cycle

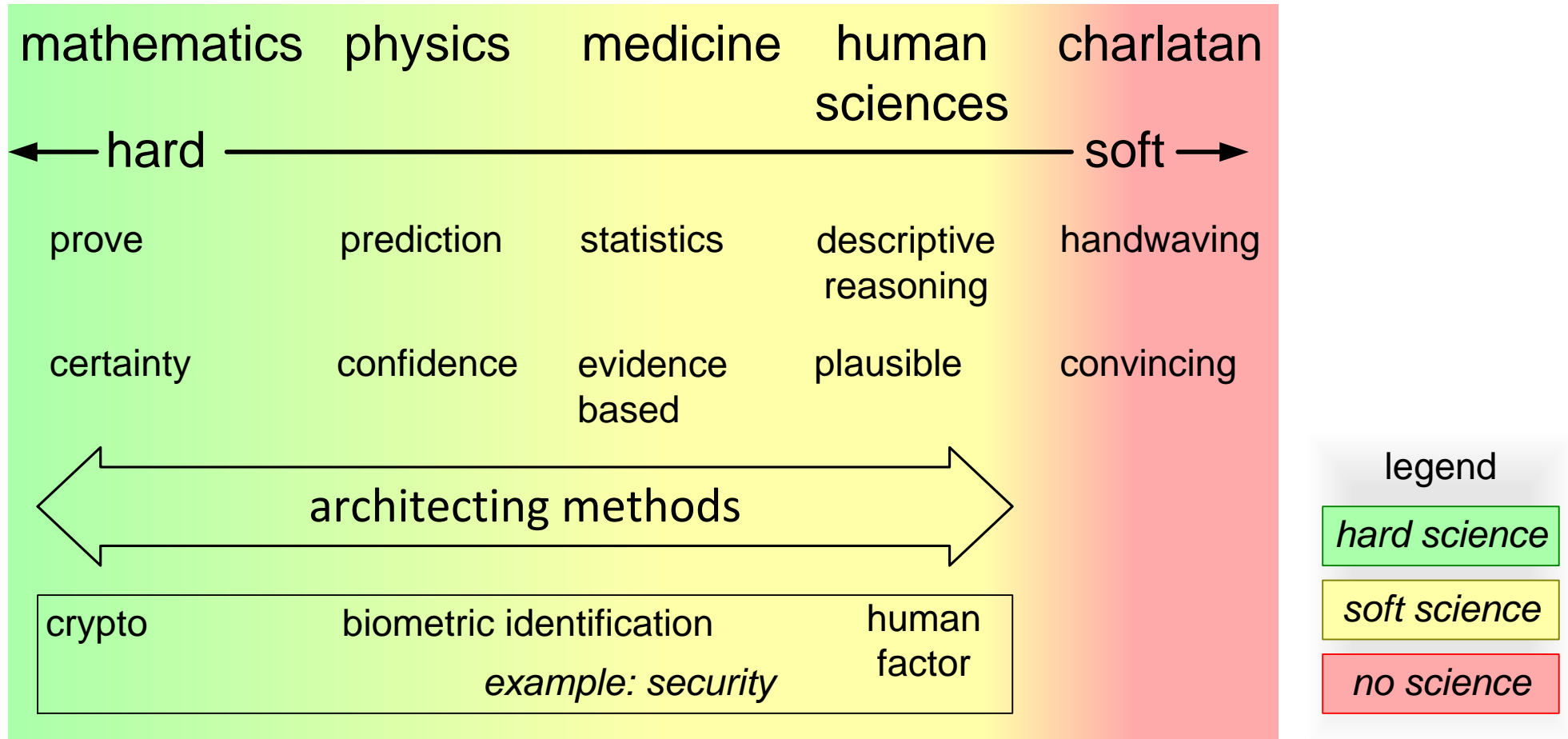


Research

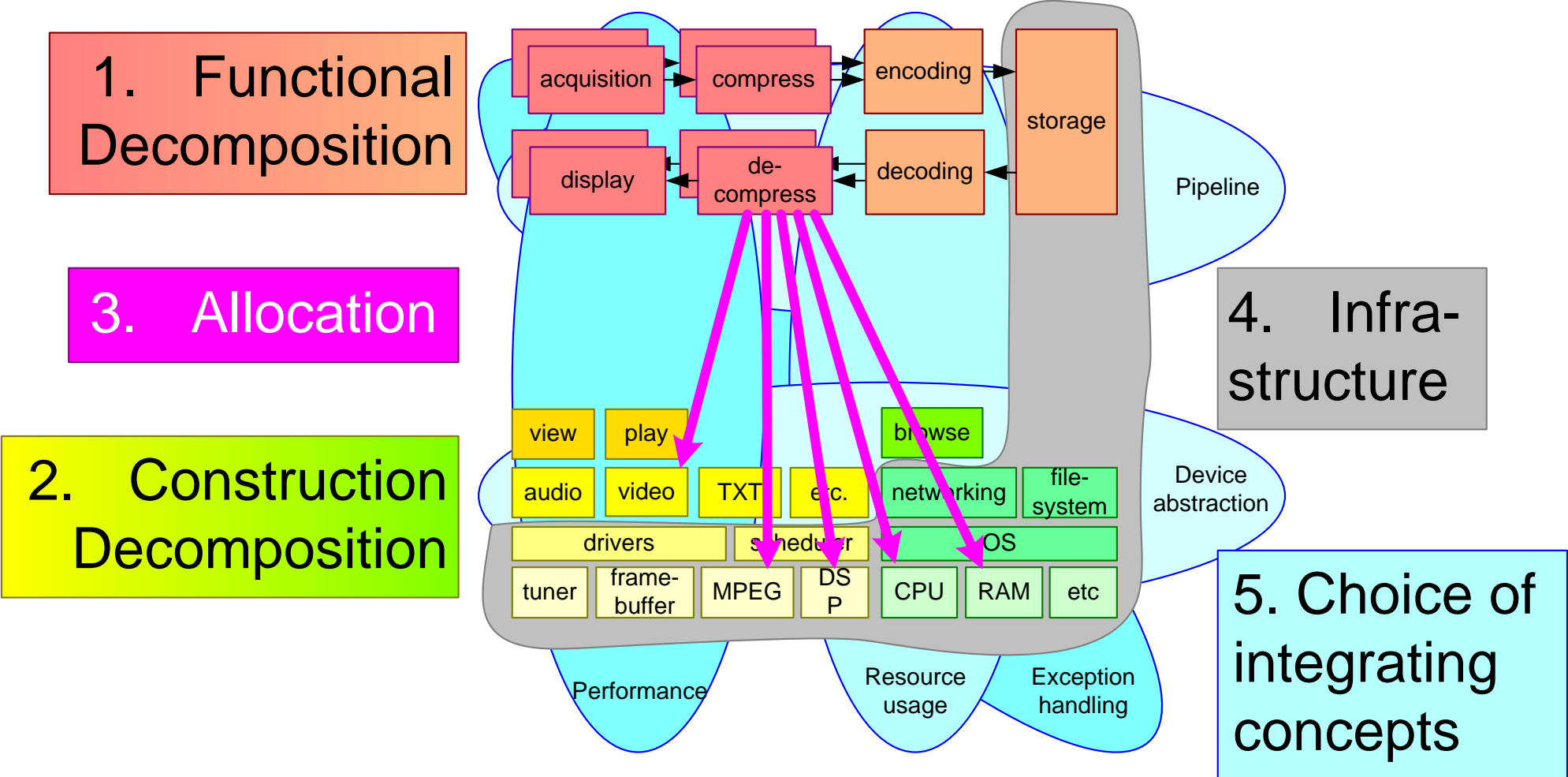


Product Division

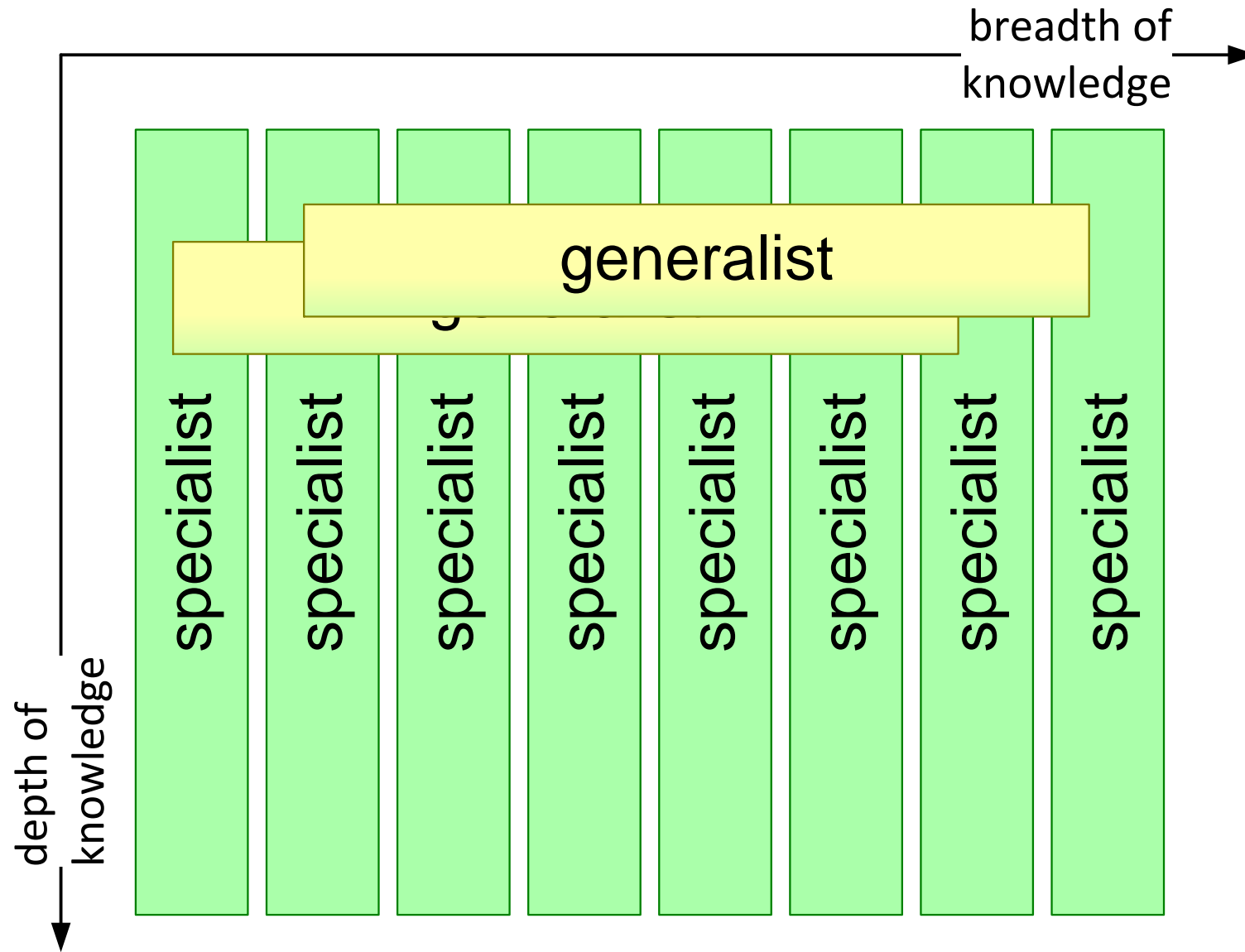
# Is architecting scientific?



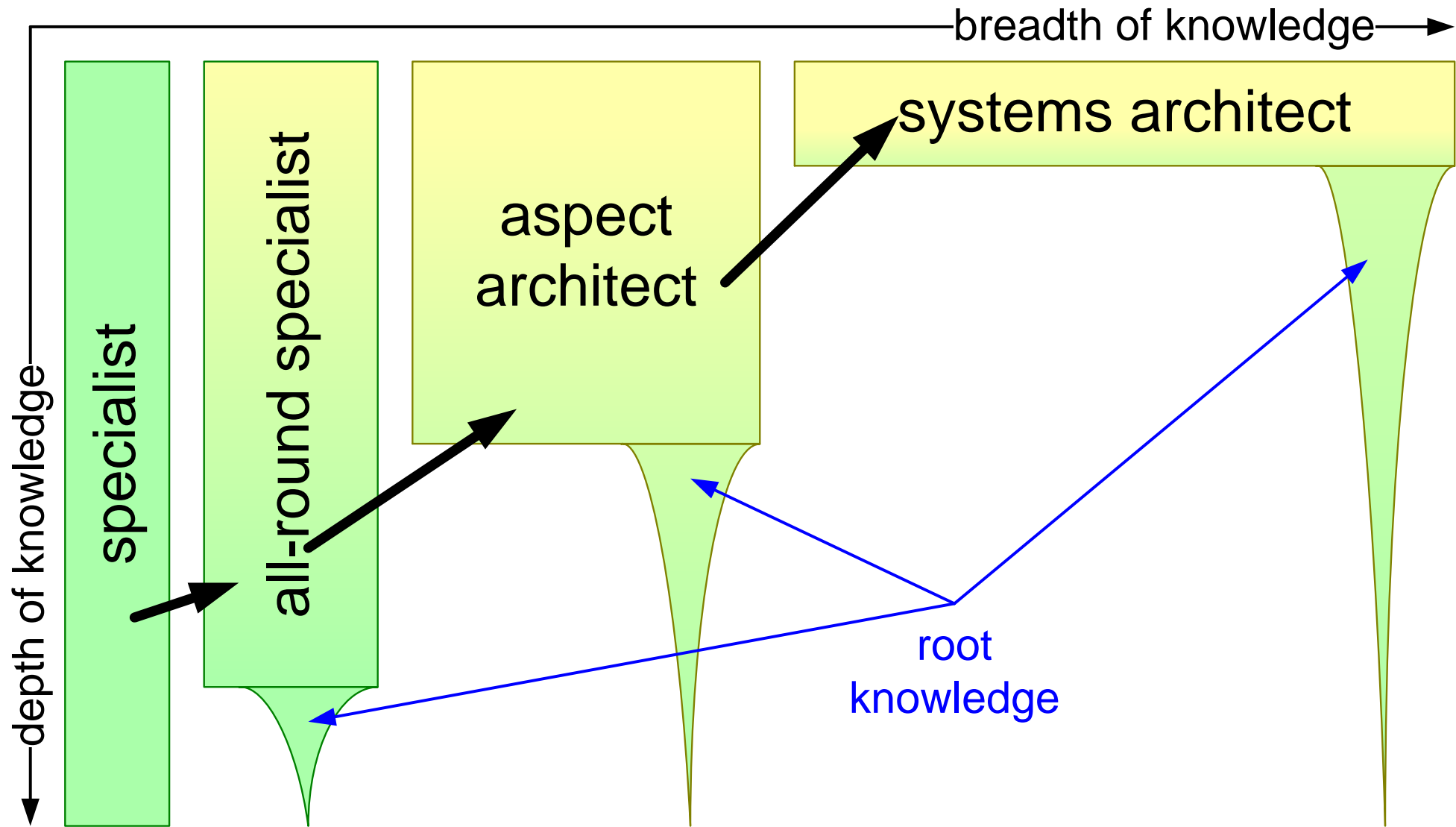
# Guiding how



# The architect as integrator



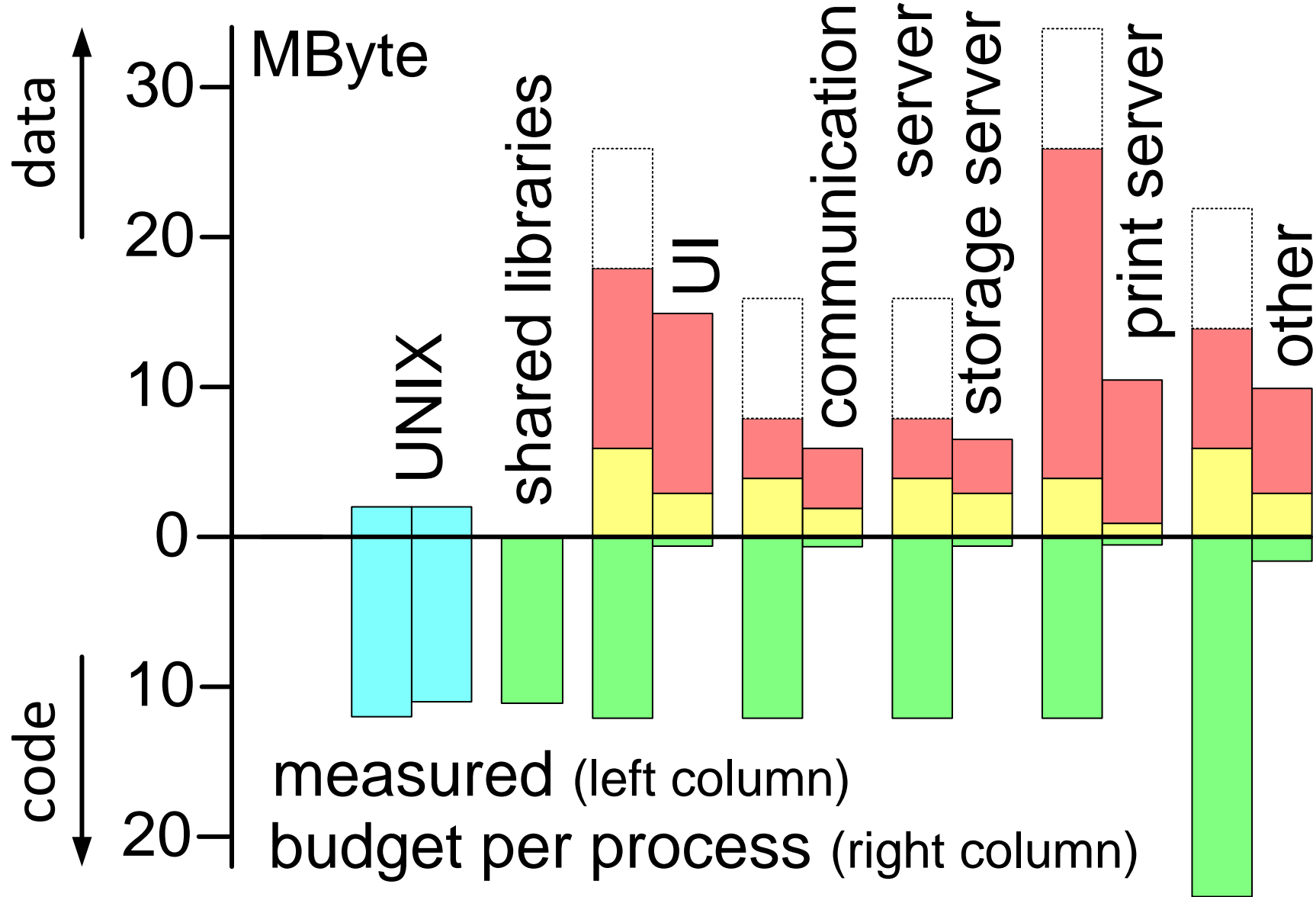
# The architect maintains technical roots



---

sheets that didn't make it into the final presentation

# Memory Budget; too detailed



# Engineers perception; too detailed, too much text

---

Engineers feeling caused by architect's activity illustrated by examples of architect behavior

Which can be nasty,  
if you don't have any answer

- Asks questions

How much memory do you use ?

Or worse, a problem is hiding  
underneath

- Identifies risks and problems

Your functions together  
use more memory than provided

A judgement is given

- Voices opinions

That is too much

Meddling in my backyard,  
without any substantial  
know how

- Does have **holes** in  
know how

Takes decisions without  
knowing how memory intensive  
a hashed dictionary is

While we are busy with  
detailed bug fixing and  
maintenance

- Does the nice work

Makes a toplevel design,  
for instance a memory budget

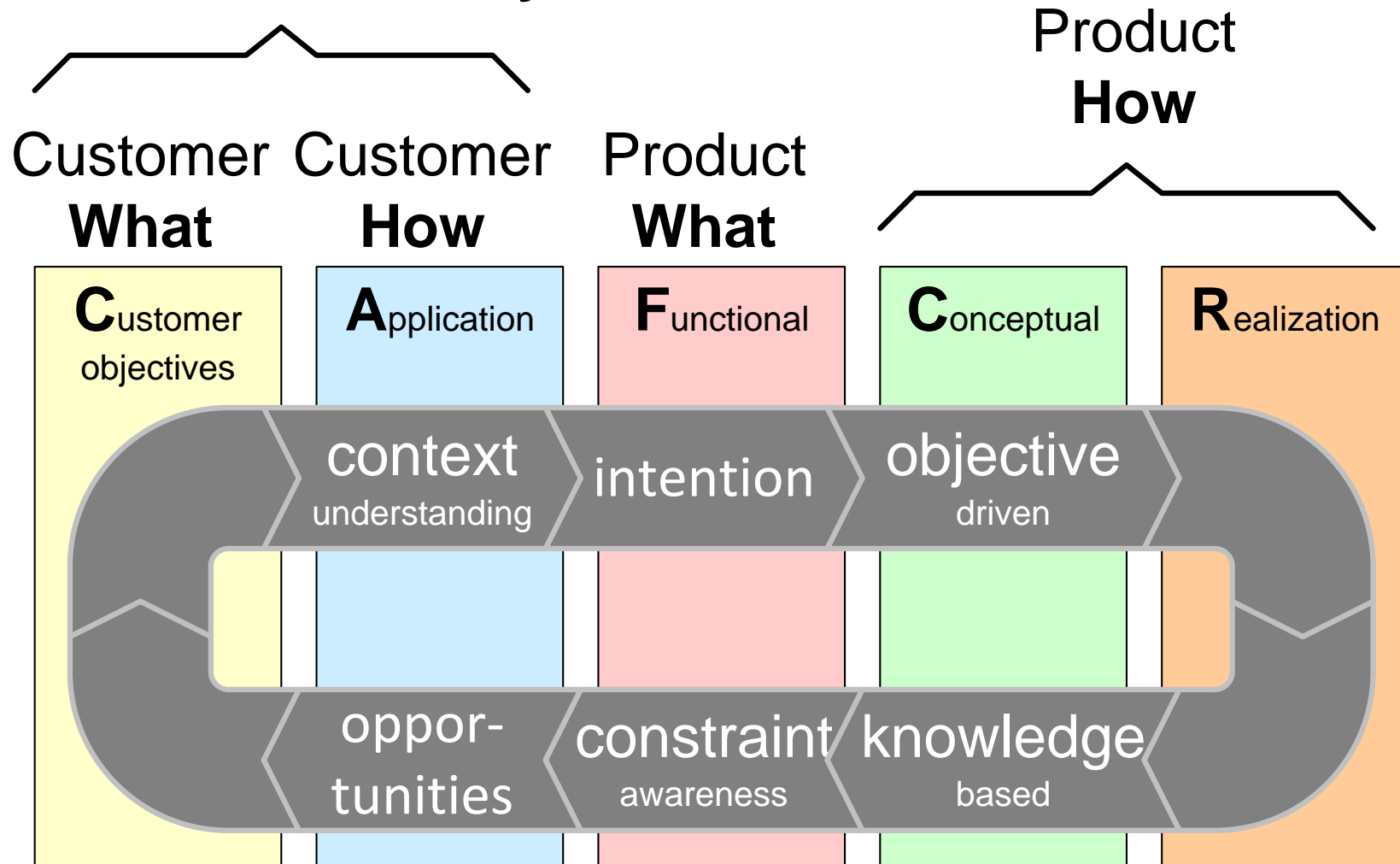
Every question or problem  
explodes into an even more  
extensive set of questions  
and problems

- Does not provide definite  
answers or solutions

What happens if we have many  
small images? Or if we have very  
large images?

# Integrating CAFCR; too glossy

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



# Architect characteristics; too detailed, too much text

---

## *Engineers perception*

- Asks questions
- Identifies risks and problems
- Voices opinions
- Does have **holes** in know how
- Does the nice work
- Does not provide definite answers or solutions

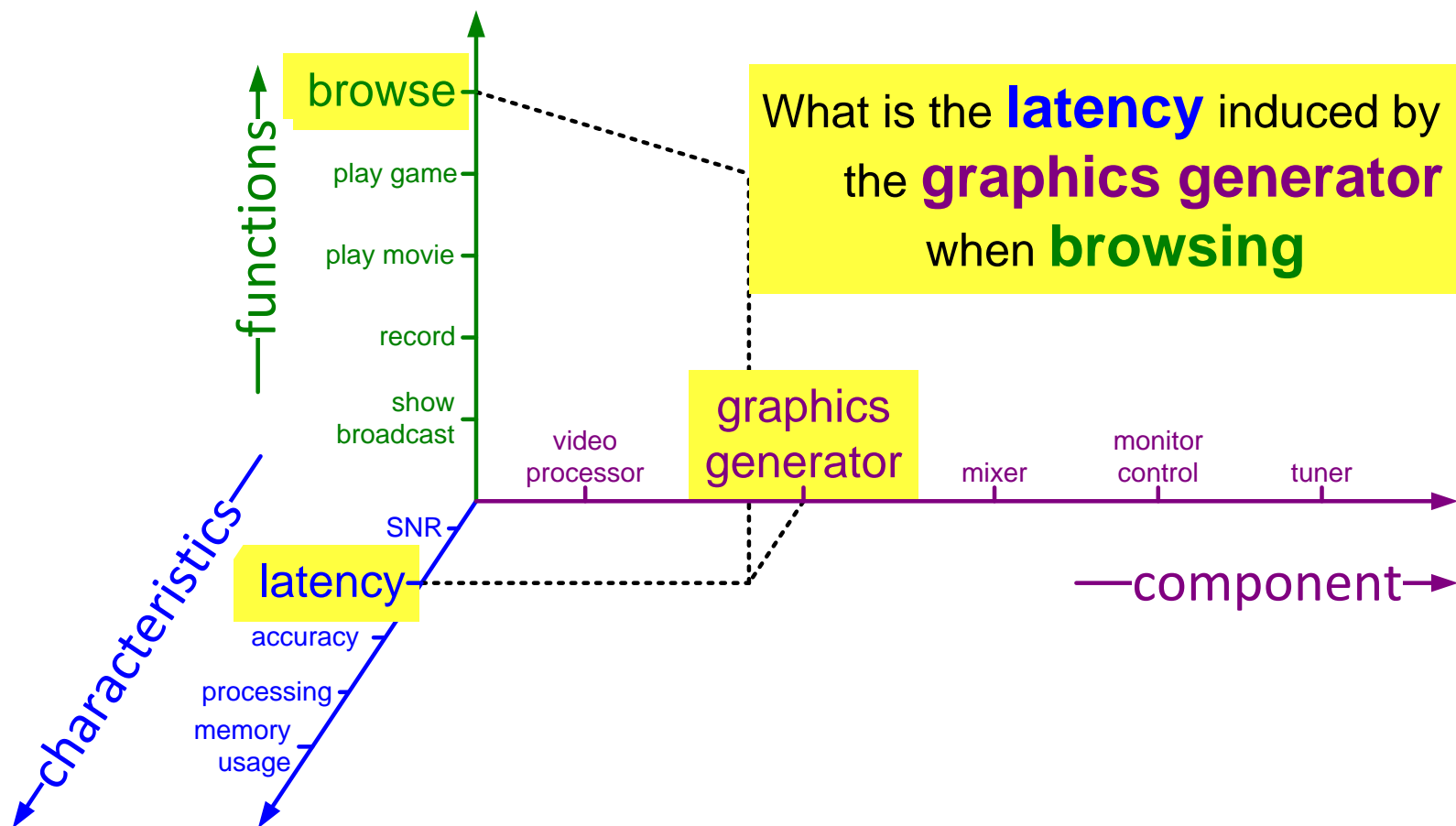
## *Required characteristic of the architect*

Questions are the primary tool of an architect

- + Prevention is valuable
- + Focus, selection is key, which requires vision and choices
- + Sometimes the architect needs to dive in deep
- + There is more nice work needed than we ever can do
- + Keeps repeating the previous actions

# Question generator; other domain

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



# Architecture Awareness Phases; too complex

