

Tutorial Software as Integrating Technology in Complex Systems

by *Gerrit Muller* Buskerud University College

e-mail: `gaudisite@gmail.com`

`www.gaudisite.nl`

Abstract

This tutorial describes the integrating value of software in complex systems. The extensive use of software technology to integrate other technologies has a significant impact on the product characteristics and on the product creation organization and process. This tutorial provides insight in the relation between software and the system, and it provides insight in the consequences for the product and the organization. Some recommendations are provided to cope with these consequences.

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.

October 20, 2017
status: concept
version: 0.1

logo
TBD

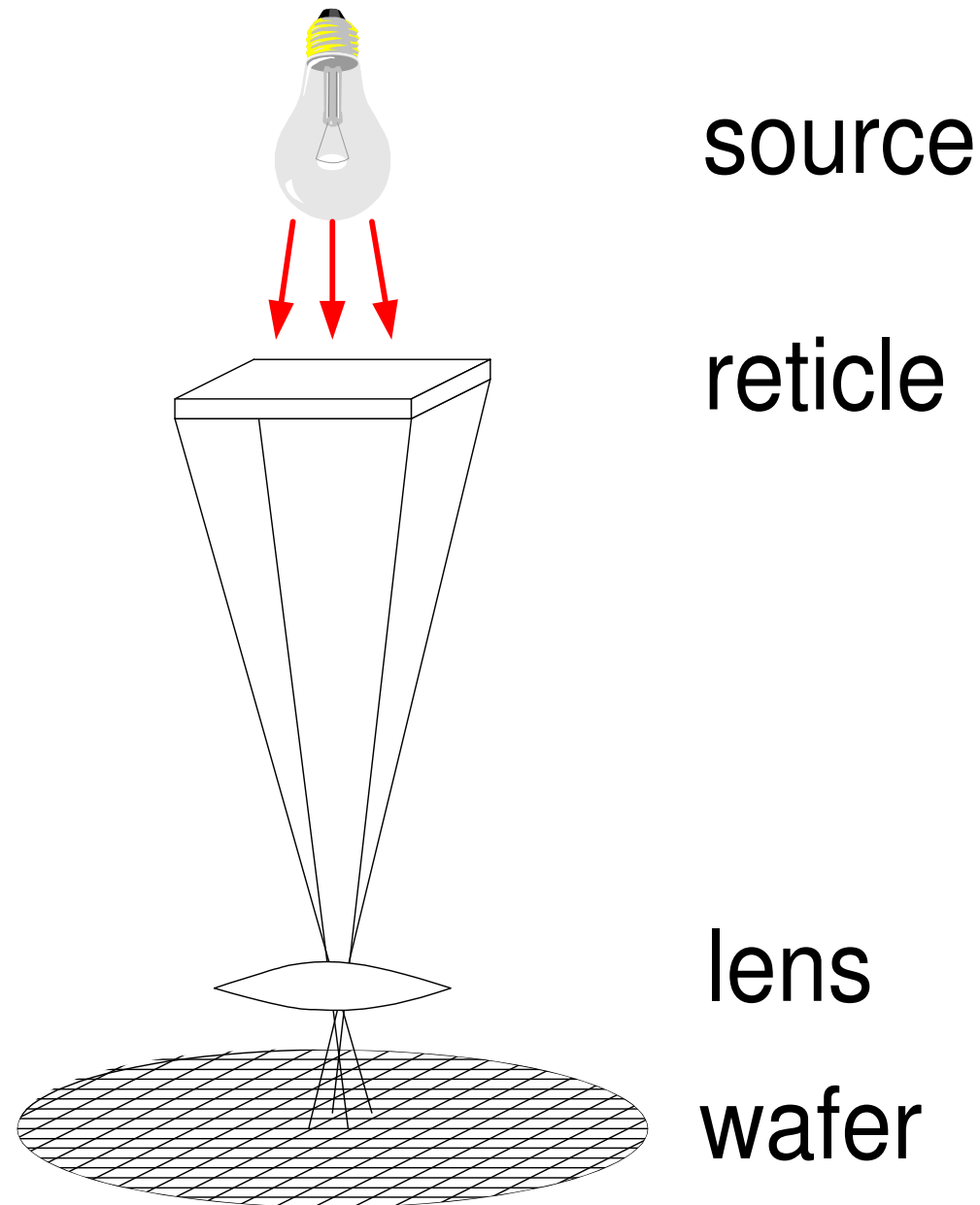
Program

- Case: the waferstepper and it's context
- The role of software in general
- Levels of abstraction
- Software -> System Functionality and Qualities
- Requirements perspective
- Evolution and Growth
- Why do we always have problems with software?
- Conclusion

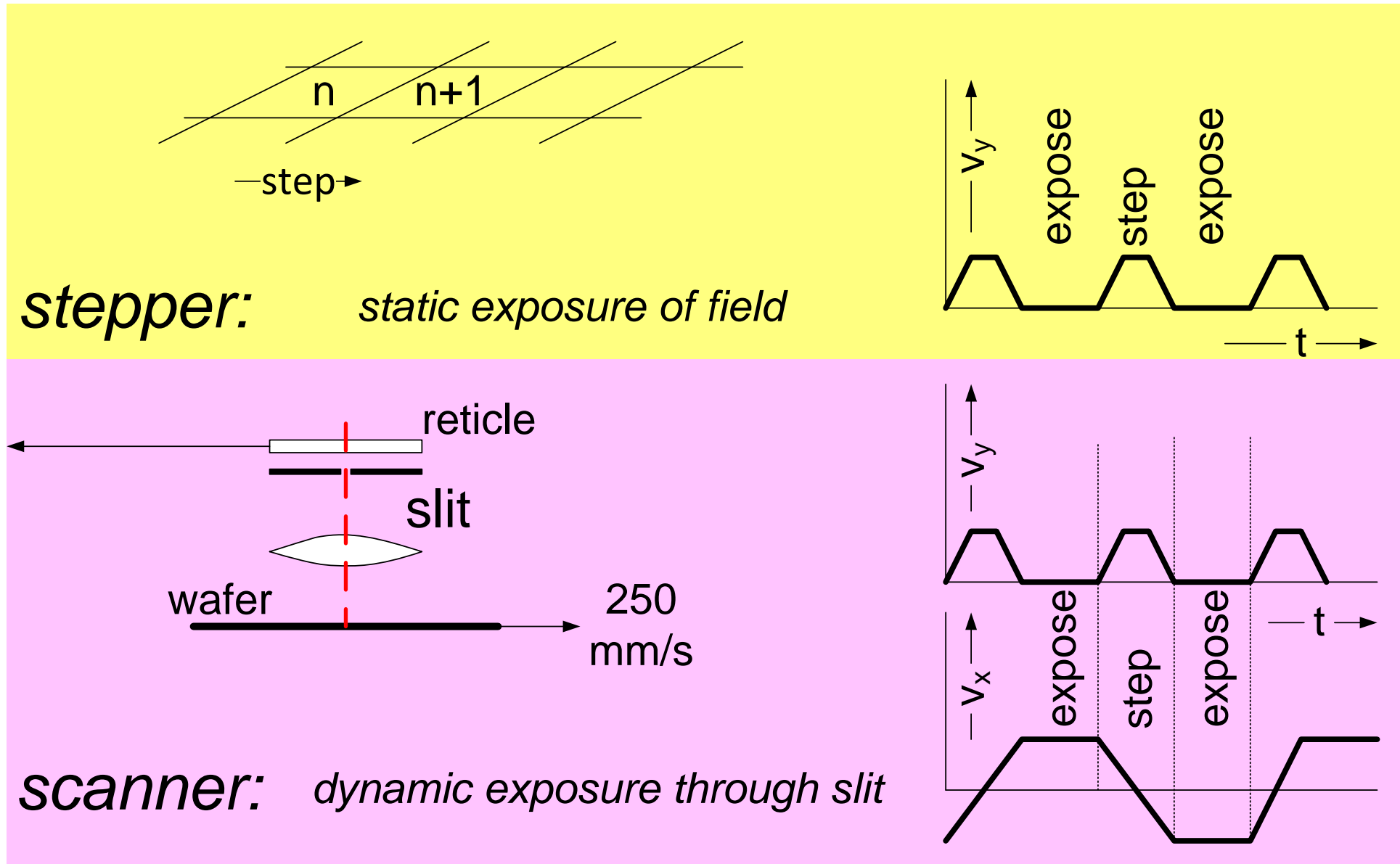
Twinscan AT1100



What is a waferstepper

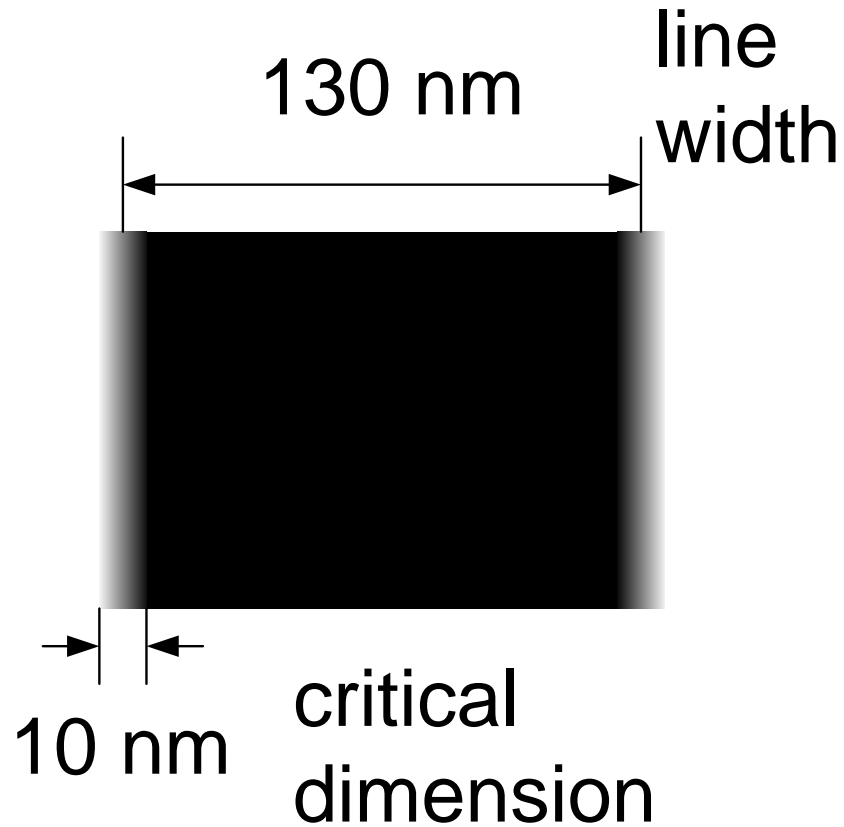


From stepping to scanning

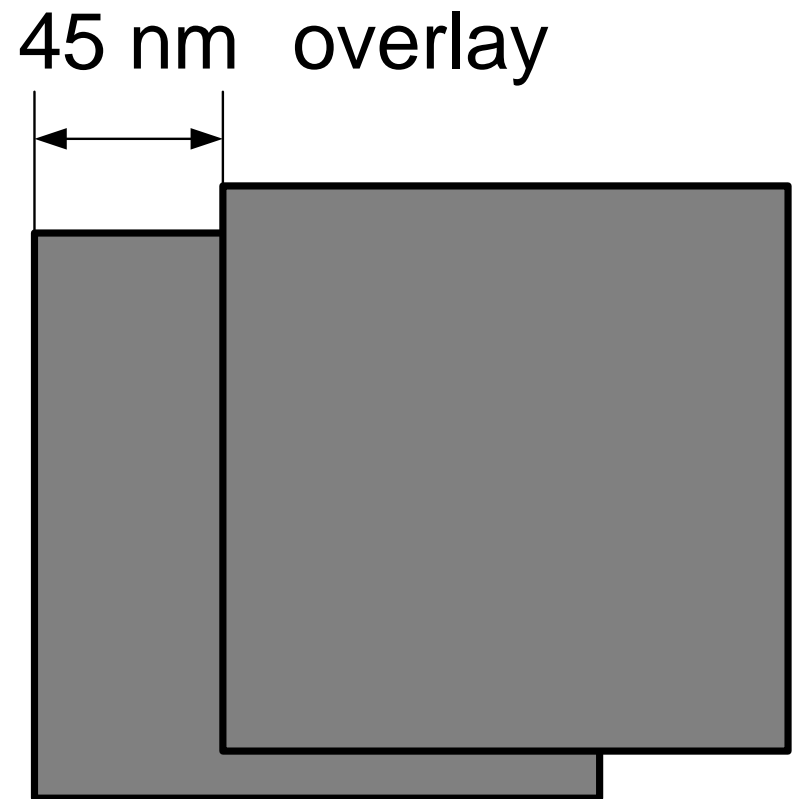


Key specifications waferstepper

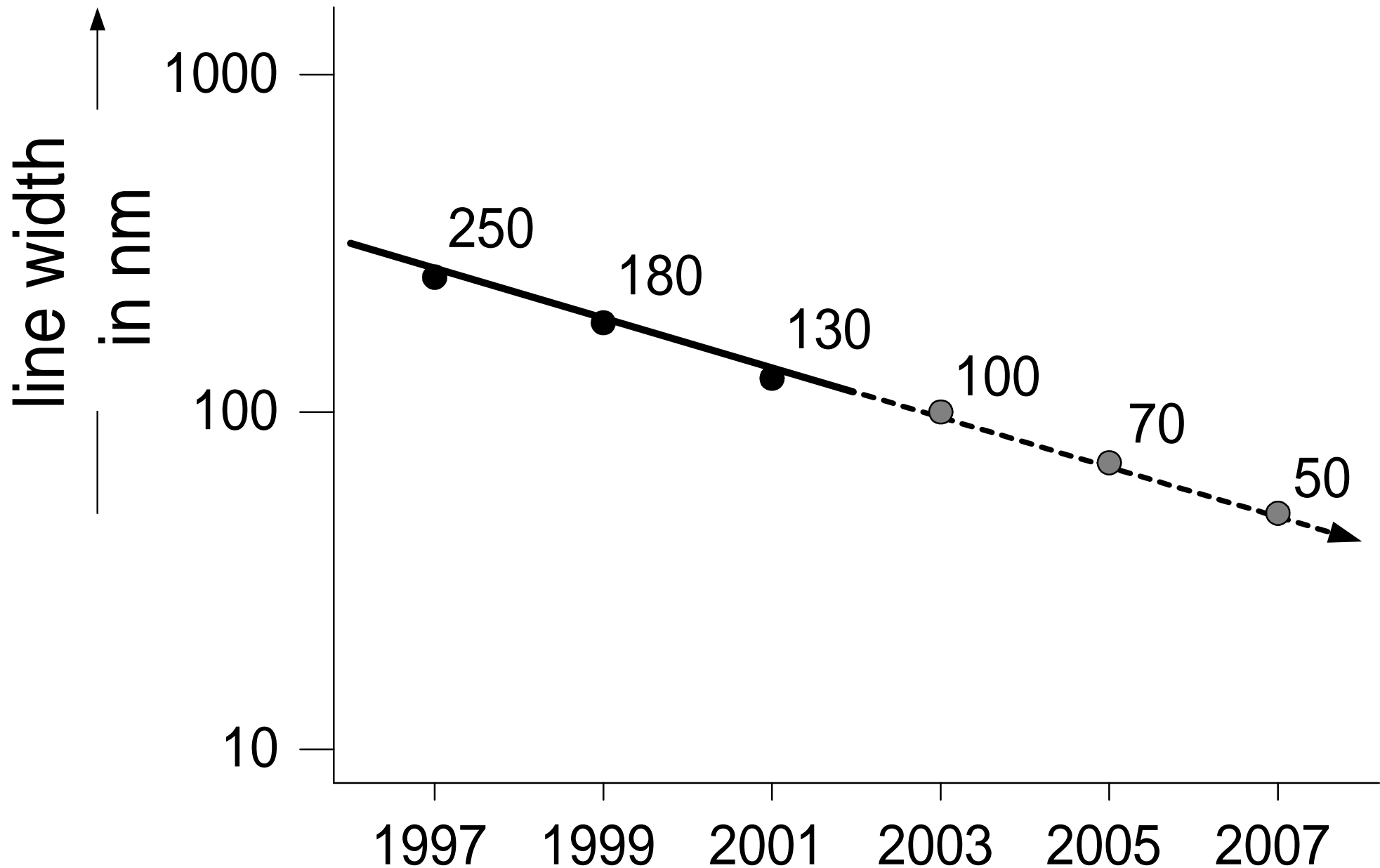
imaging



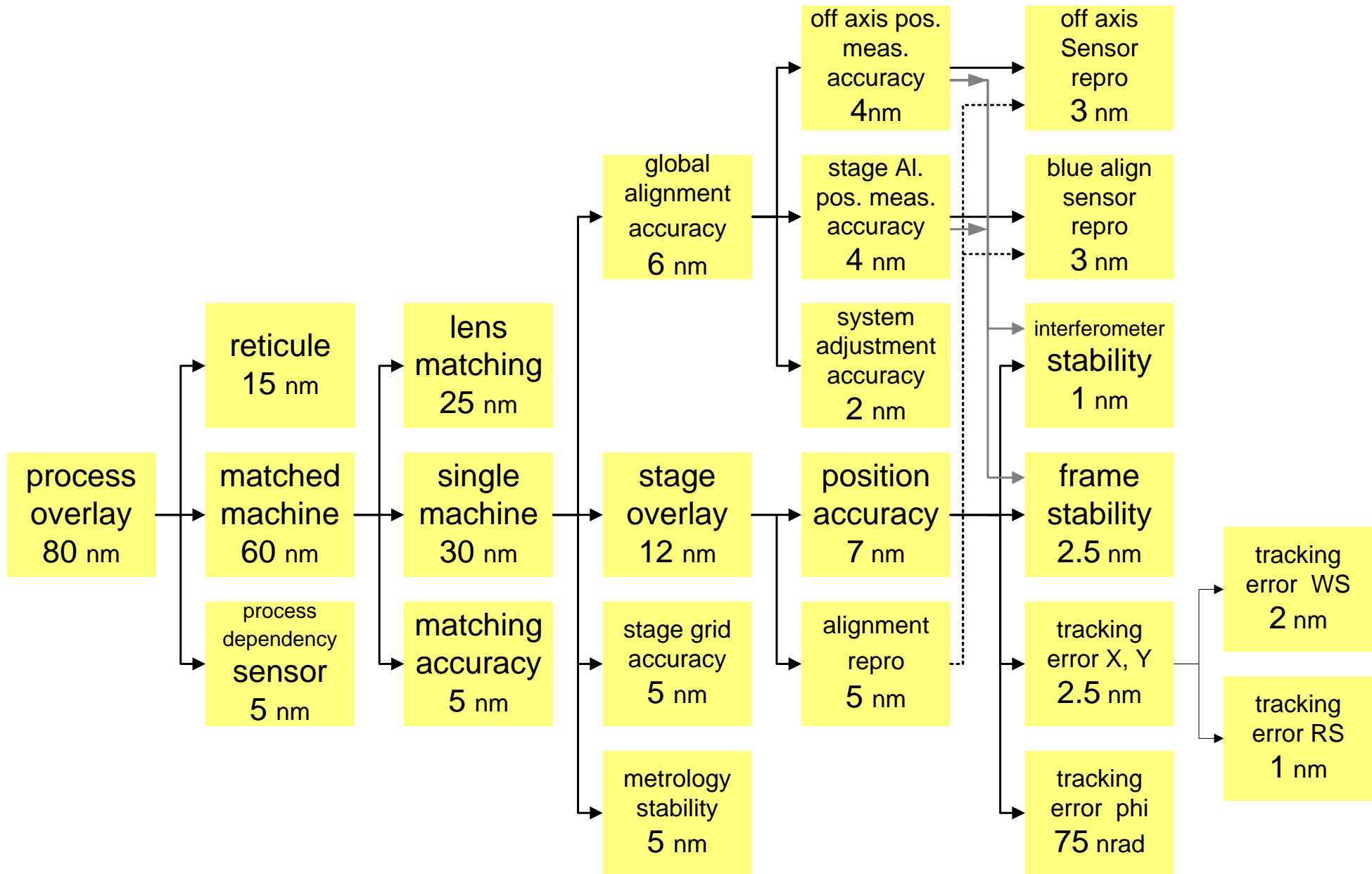
alignment



Moore's law




Overlay budget (1999)

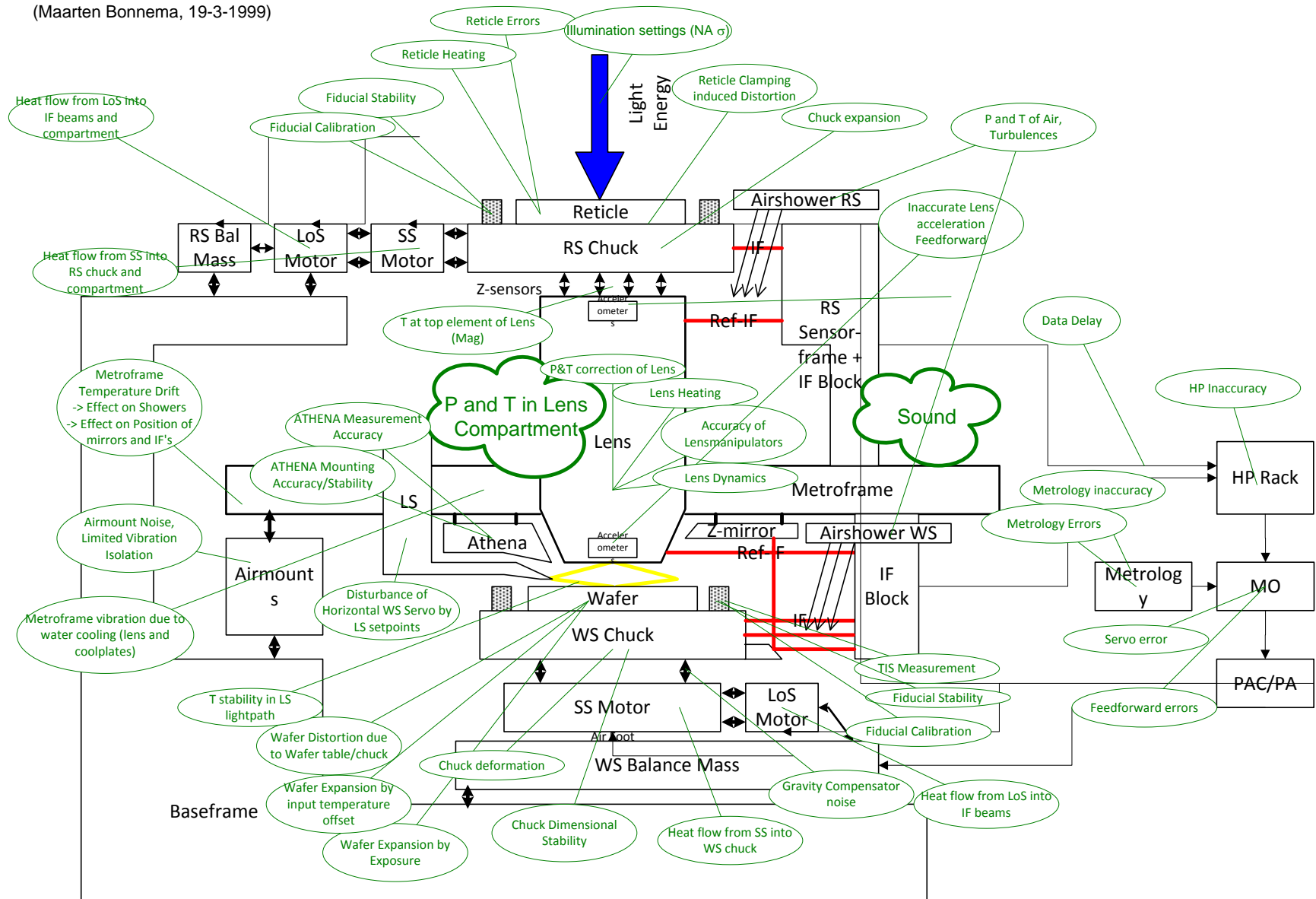


Everything influences overlay

Overlay Influence Diagram.

(Maarten Bonnema, 19-3-1999)

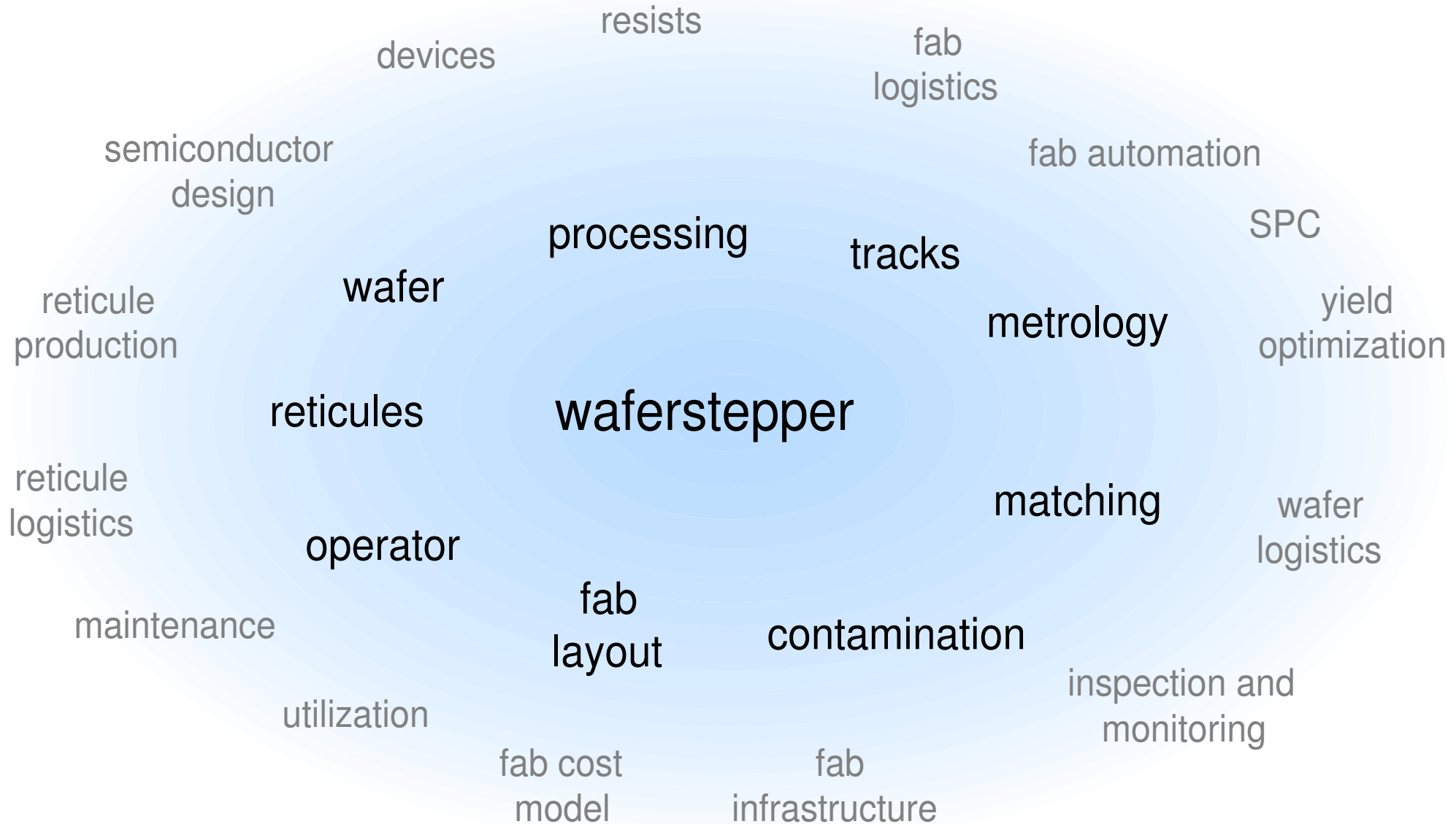
 : Fiducial



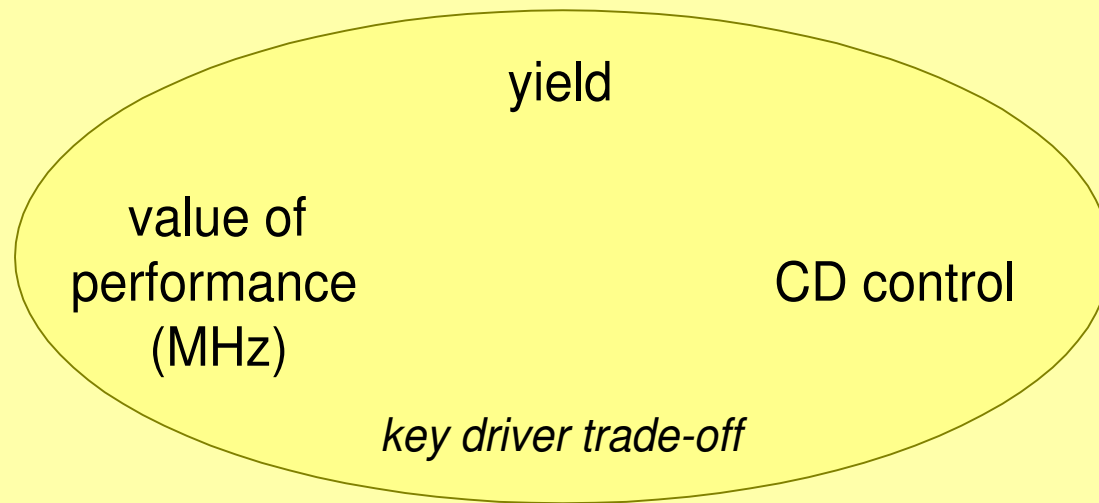
Exercise 1, 10 minutes

Make a 3 picture description (What, How, biggest challenge) of your own system.

Fab Context of Waferstepper



Business Context



other players:
equipments vendors
system integrators
lease companies
fab designers
consultants
mask makers
resist makers
wafer makers
OEM's: laser
intimate partners: lens

business models of the customer:
design houses
foundries
vertical integration

Limited number of customers;
Many systems per customer

Human Context: Stakeholders

"external"

customer
purchaser
decision maker
user
operator
maintainer

other
government
customer's customer
banks, insurance

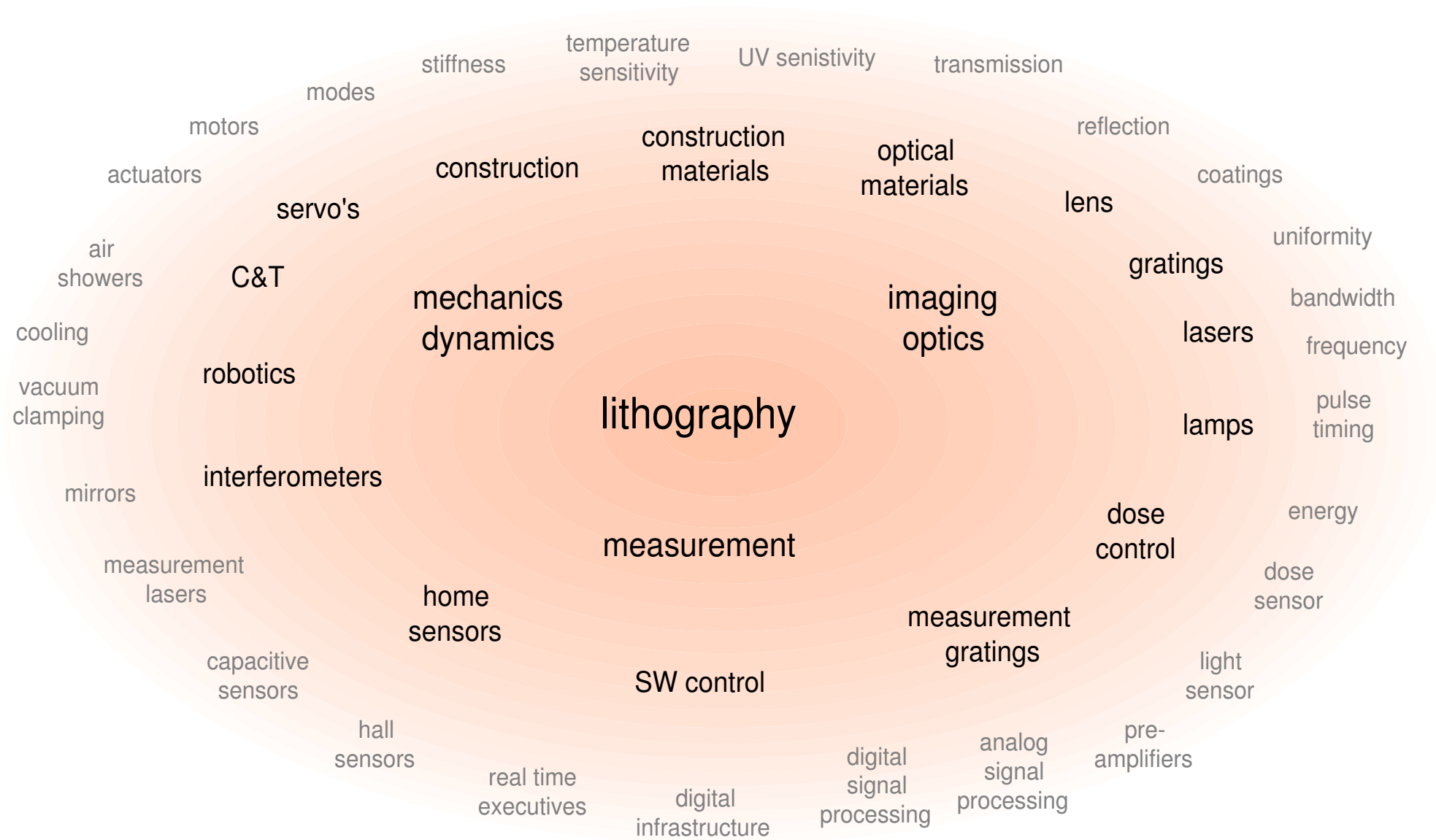
"internal"

managers
business manager
marketing manager
product manager
operational manager
project leader
sales manager
quality manager
logistics manager
line manager
technology manager

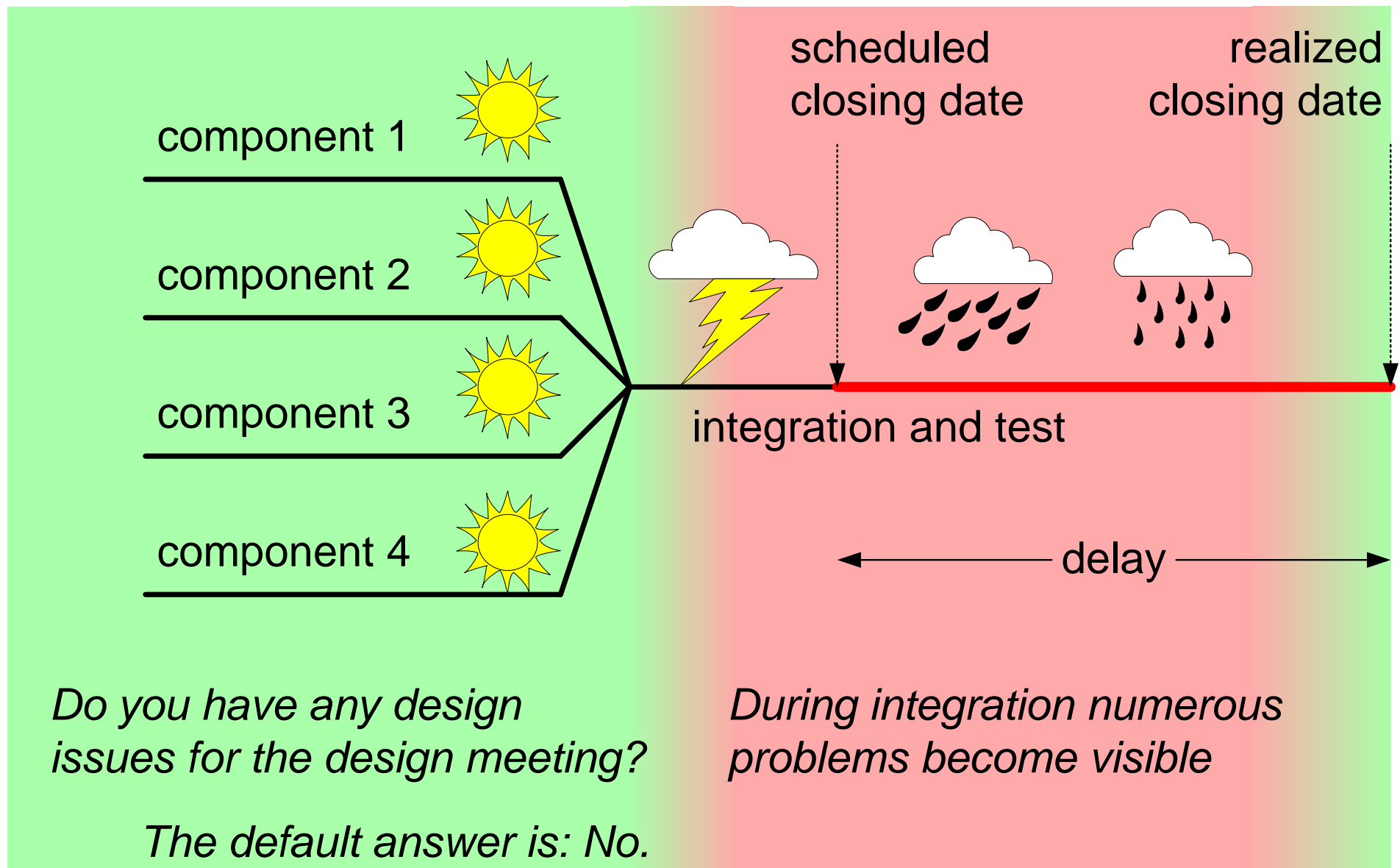
engineers
system engineers
experts
manufacturing engineers
customer support

suppliers
component manufacturer
outsourced design

Multitude of Disciplines



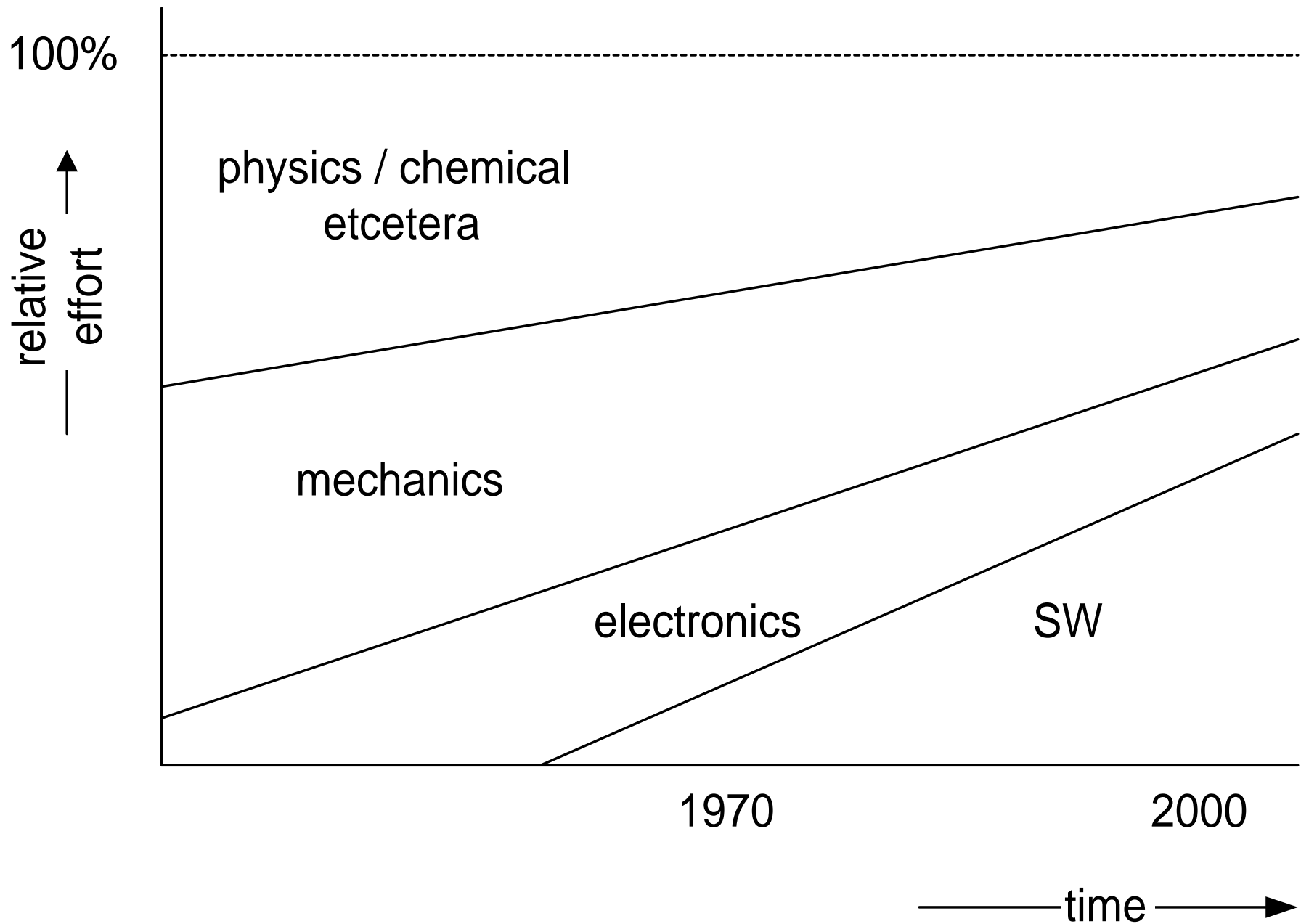
Symptom: Delays appear during Integration



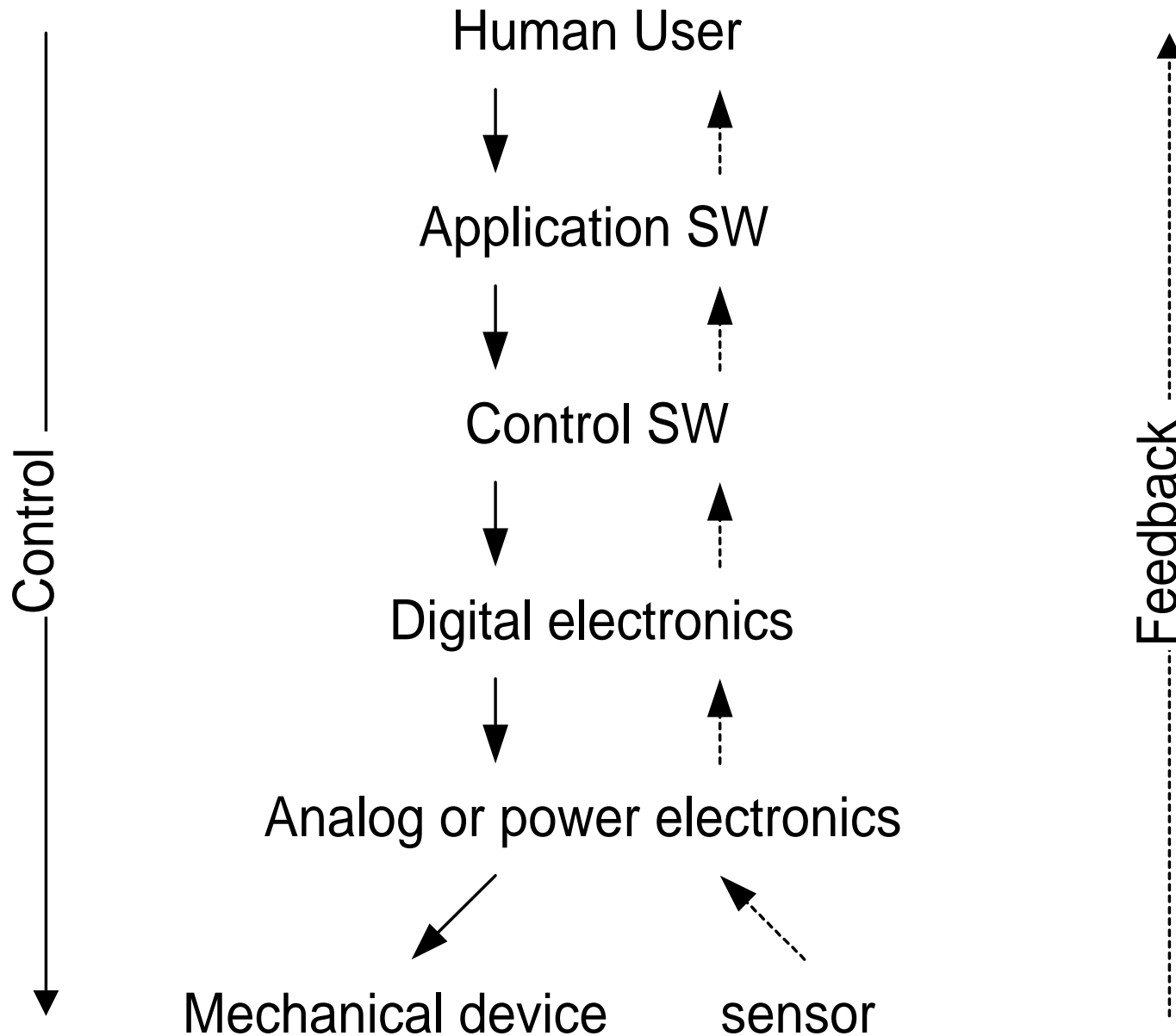
Exercise 2, 10 minutes

Make a 3 picture description (Application context, Value chain, technologies) of your own system.

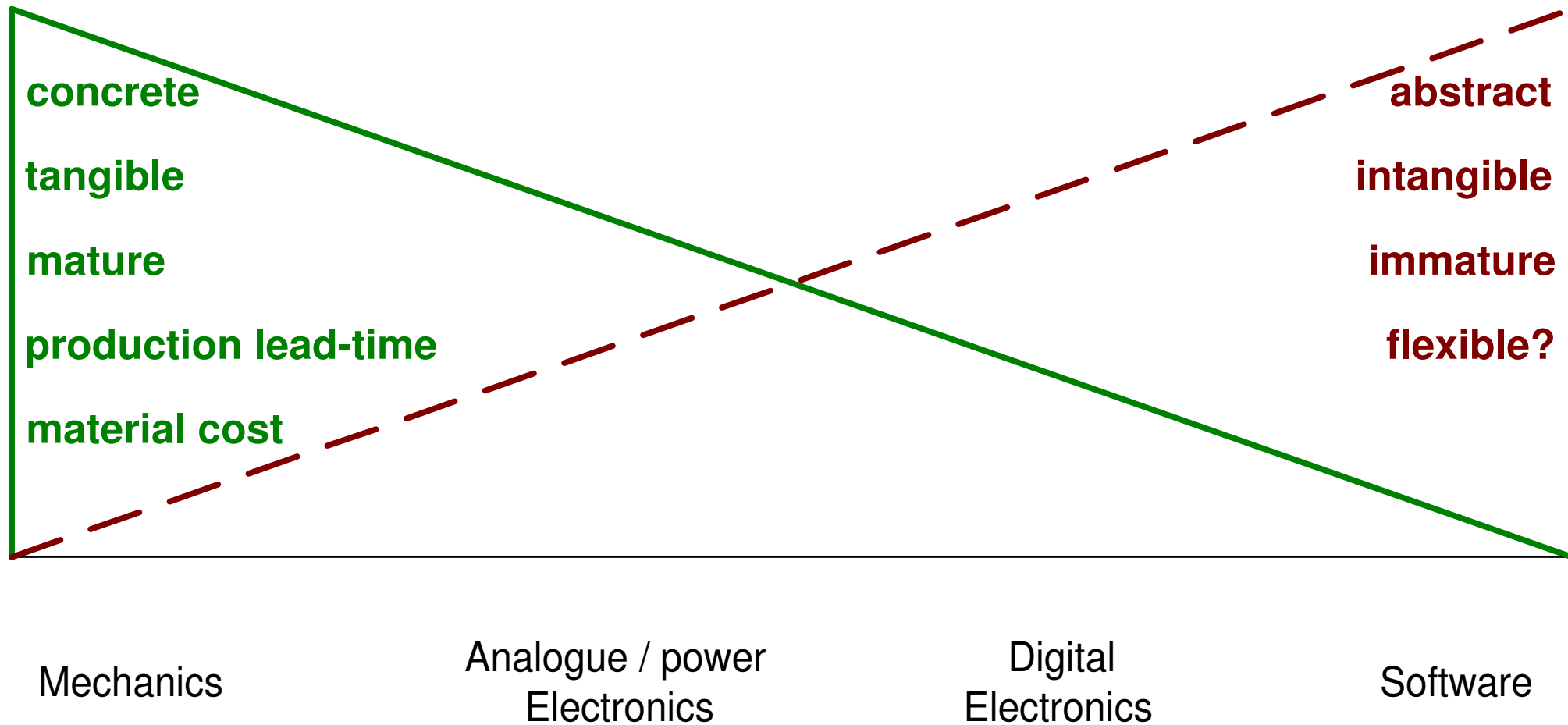
Relative Contribution of SW



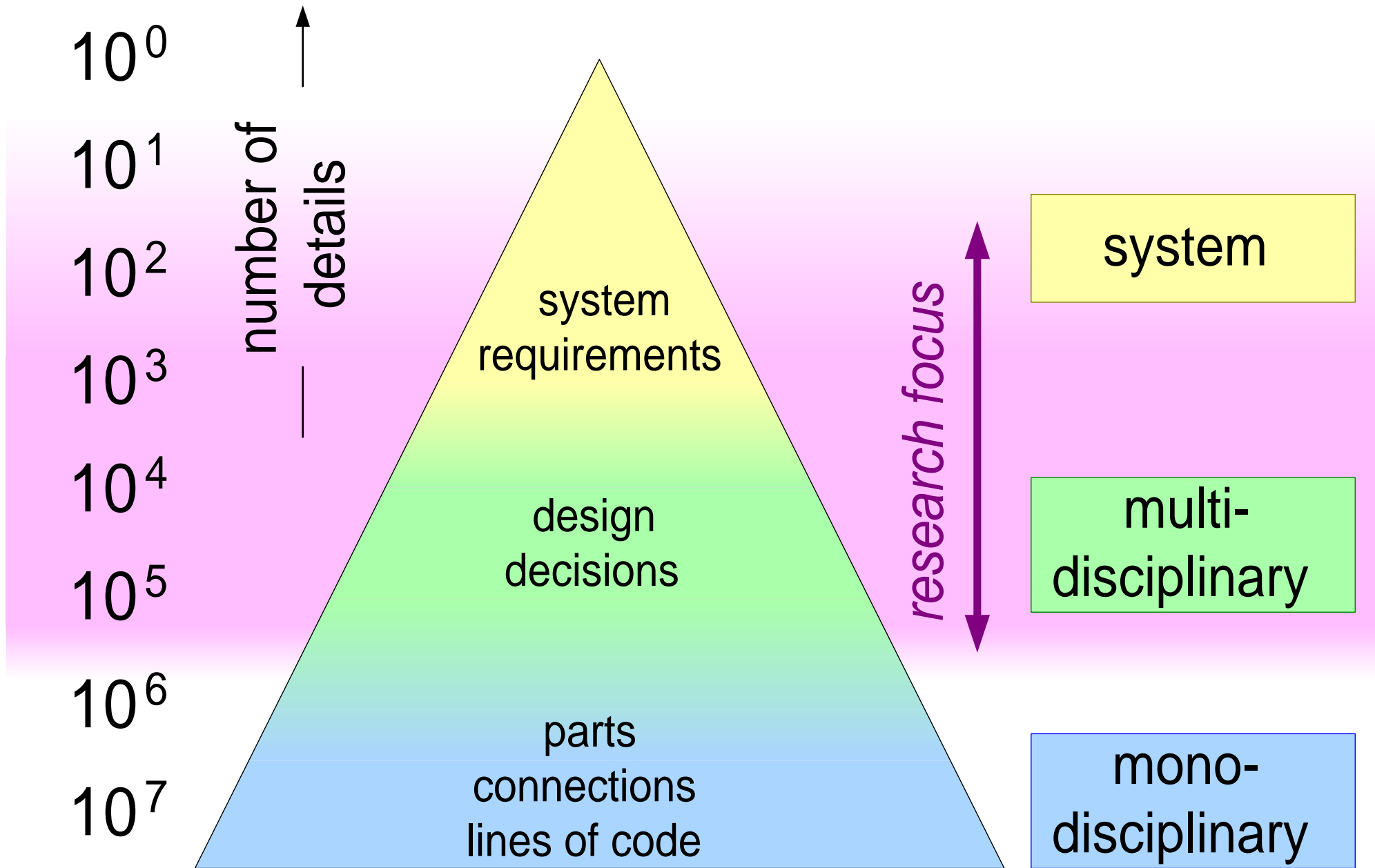
Control Hierarchy along Technology axis



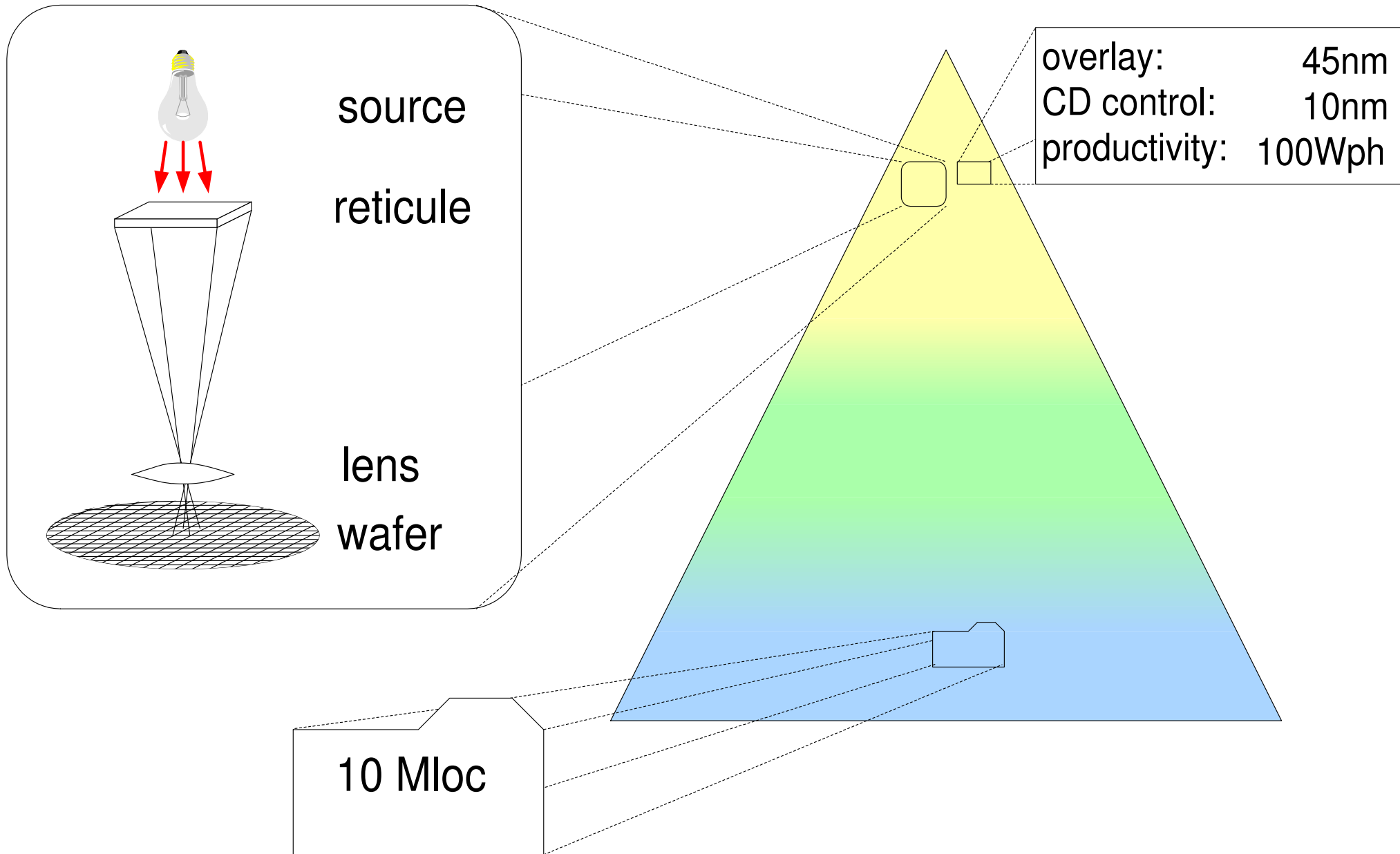
Characterization of disciplines



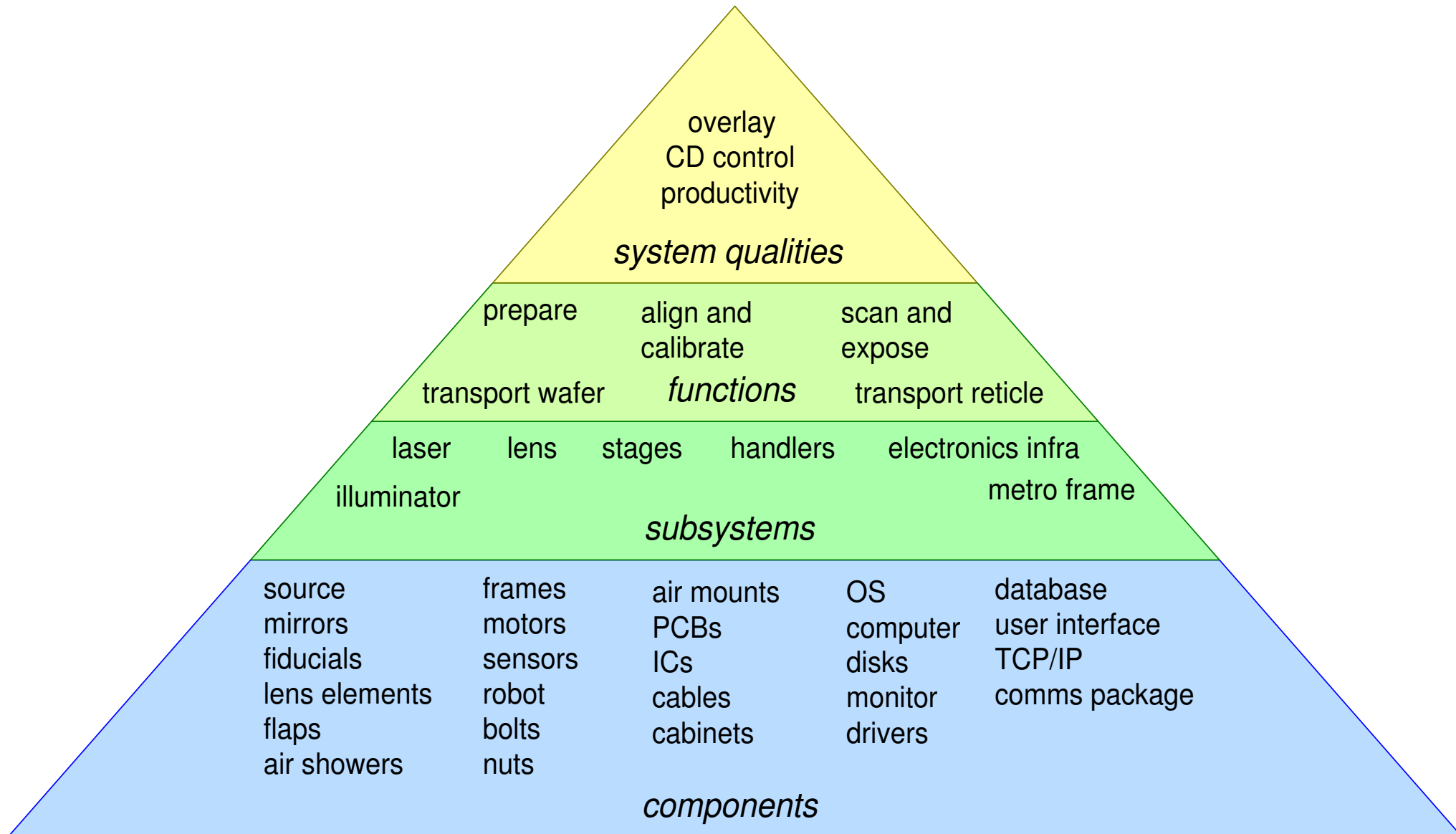
Exponential Pyramid, from requirement to bolts and nuts



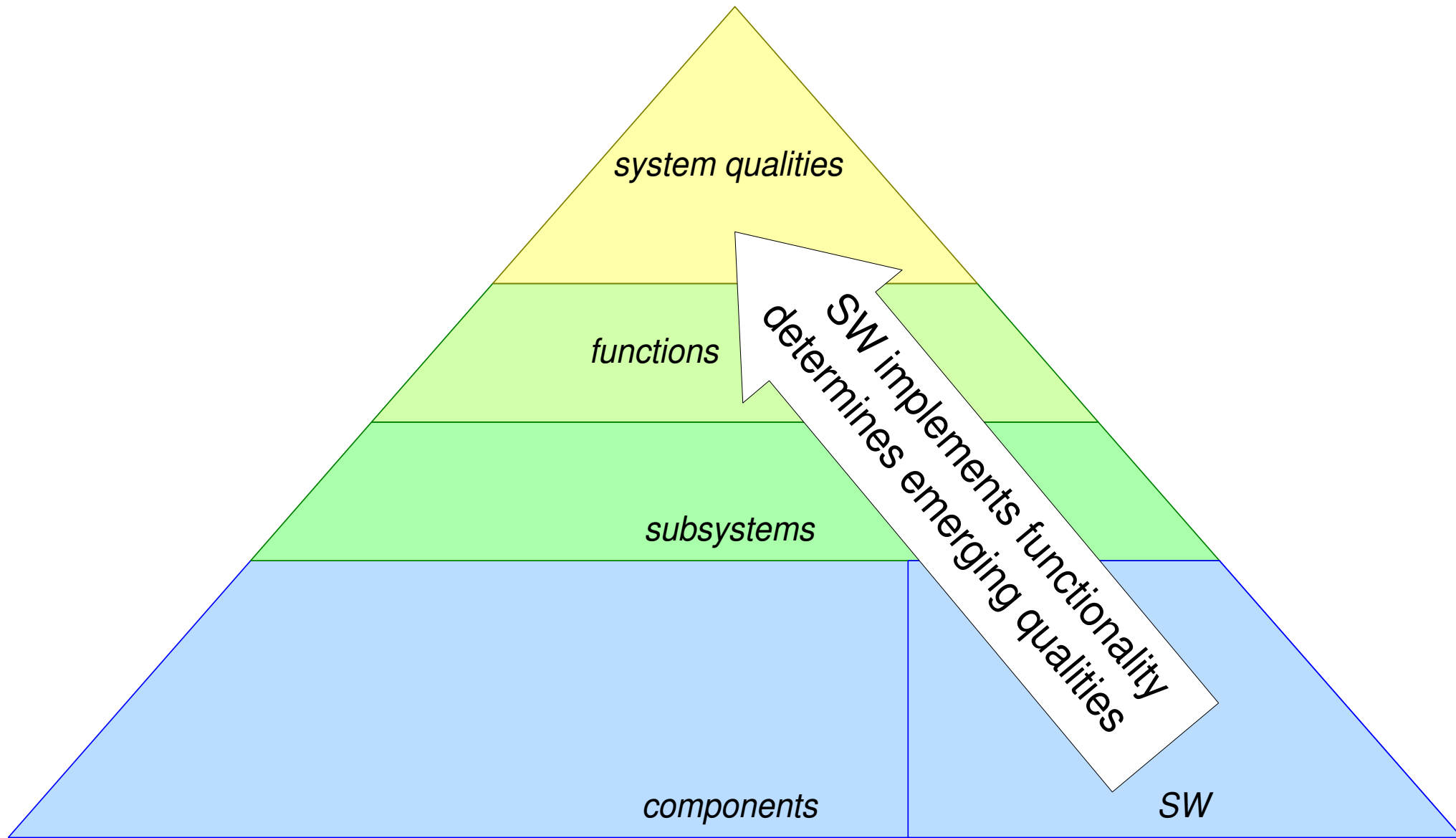
Waferstepper Example



From Components to System Qualities



Role of Software

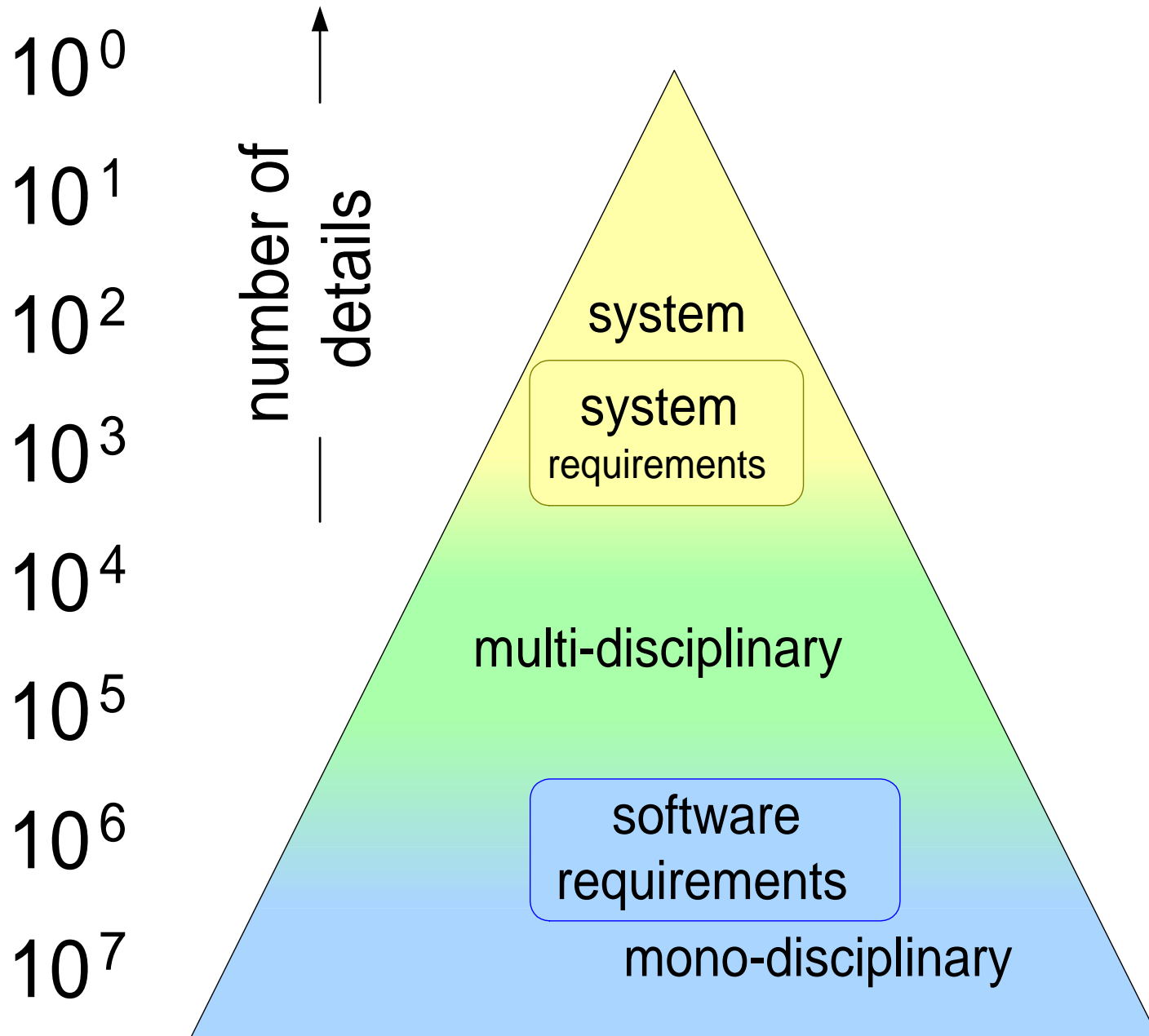


Exercise 3, 10 minutes

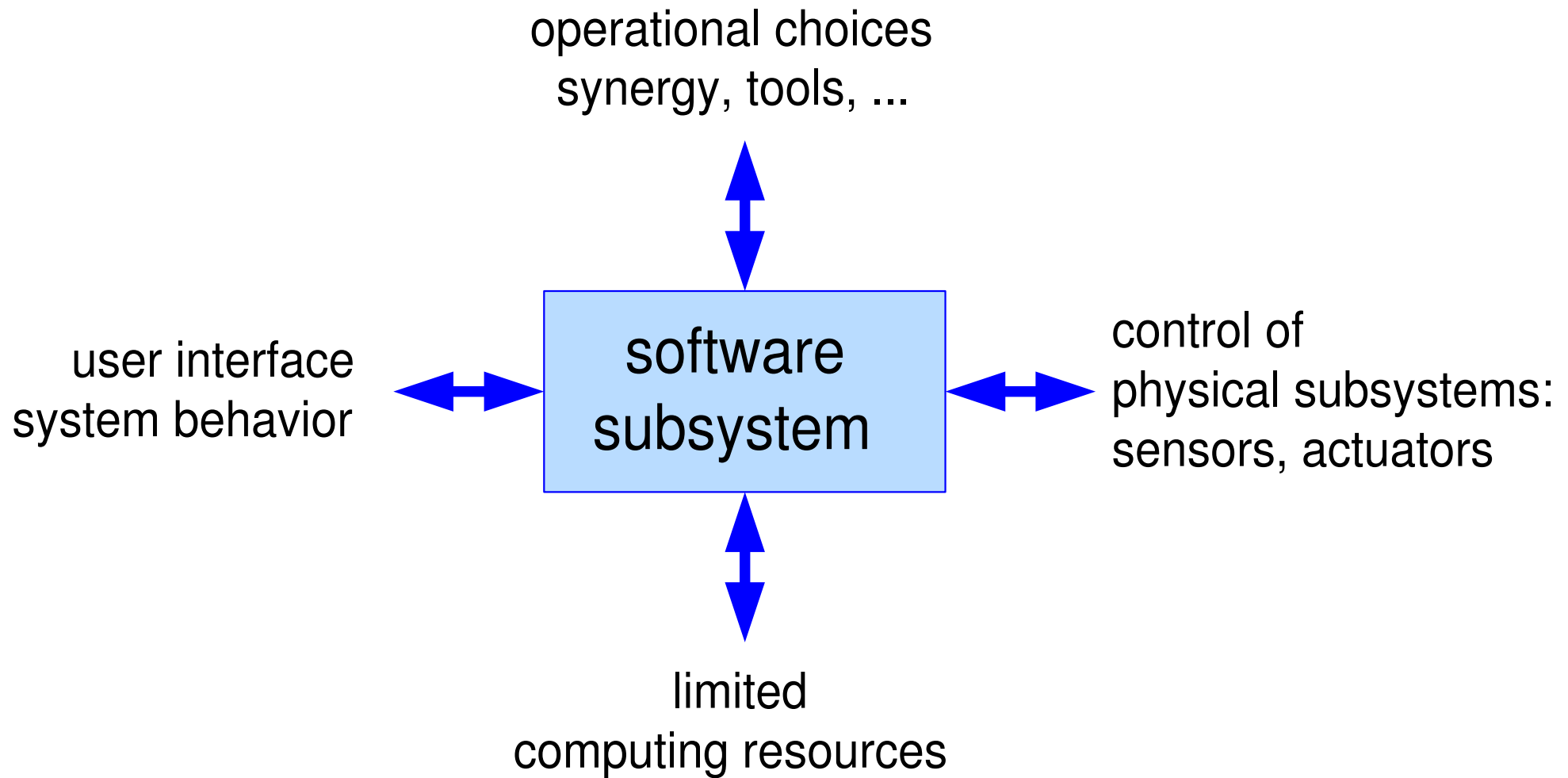
Make a toplevel decomposition of the software in your system and estimate the amount of software of the constituting parts

When SW engineers demand "requirements",
then they expect *frozen* inputs
to be used for
the design, implementation and validation
of the software

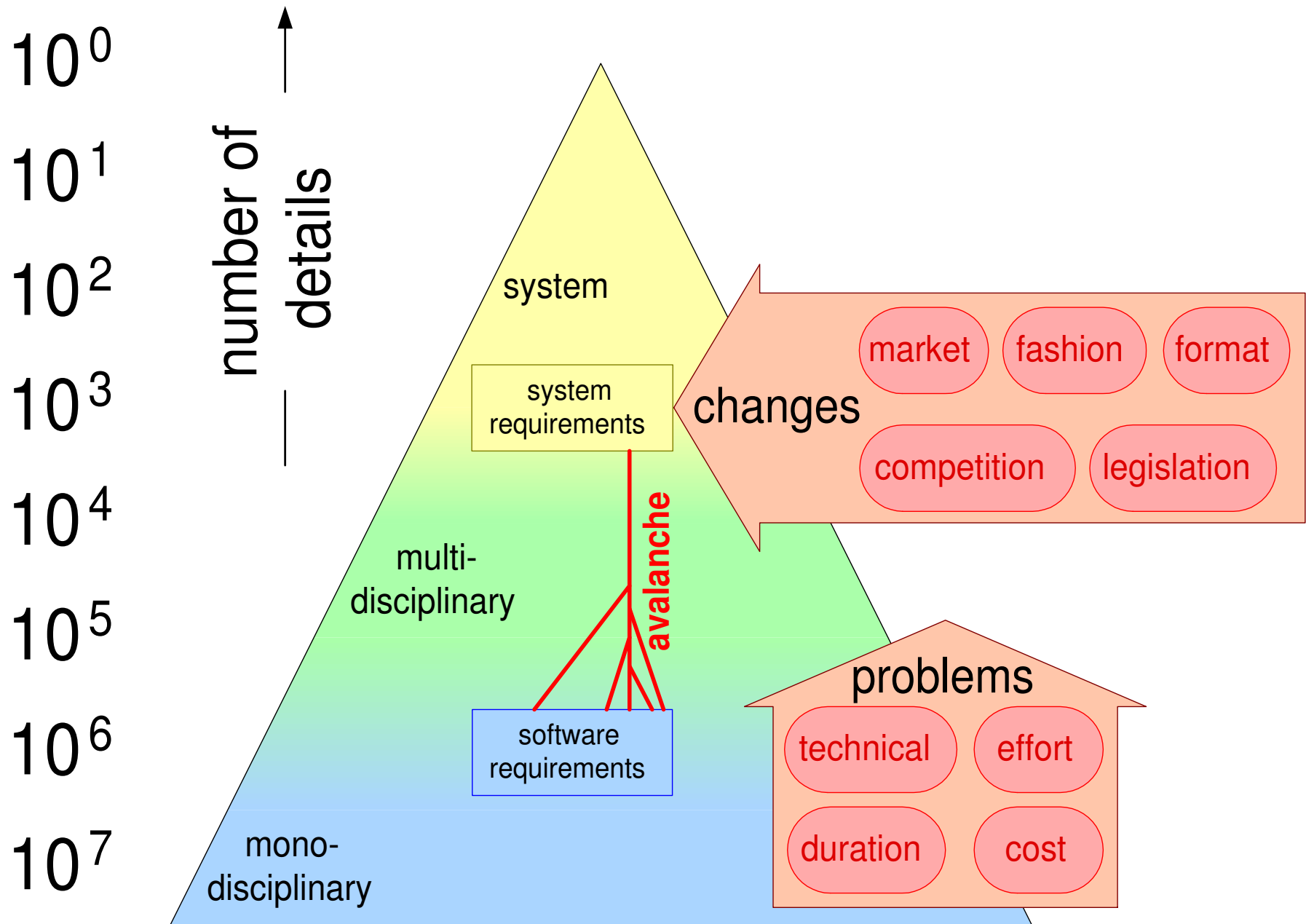
System vs Software Requirements



Why is the Software Requirement Specification so Large?



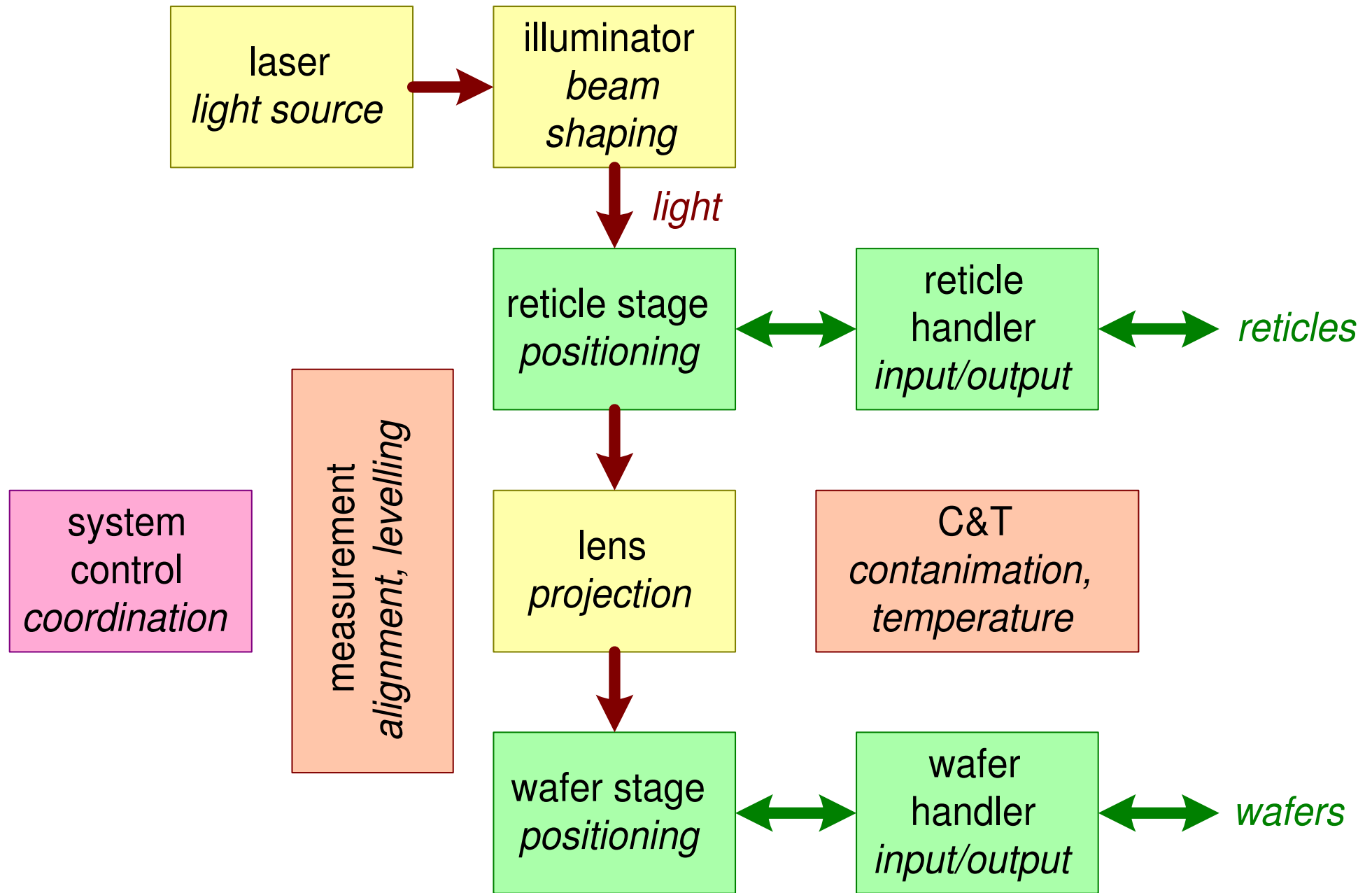
And why is it never up-to-date?



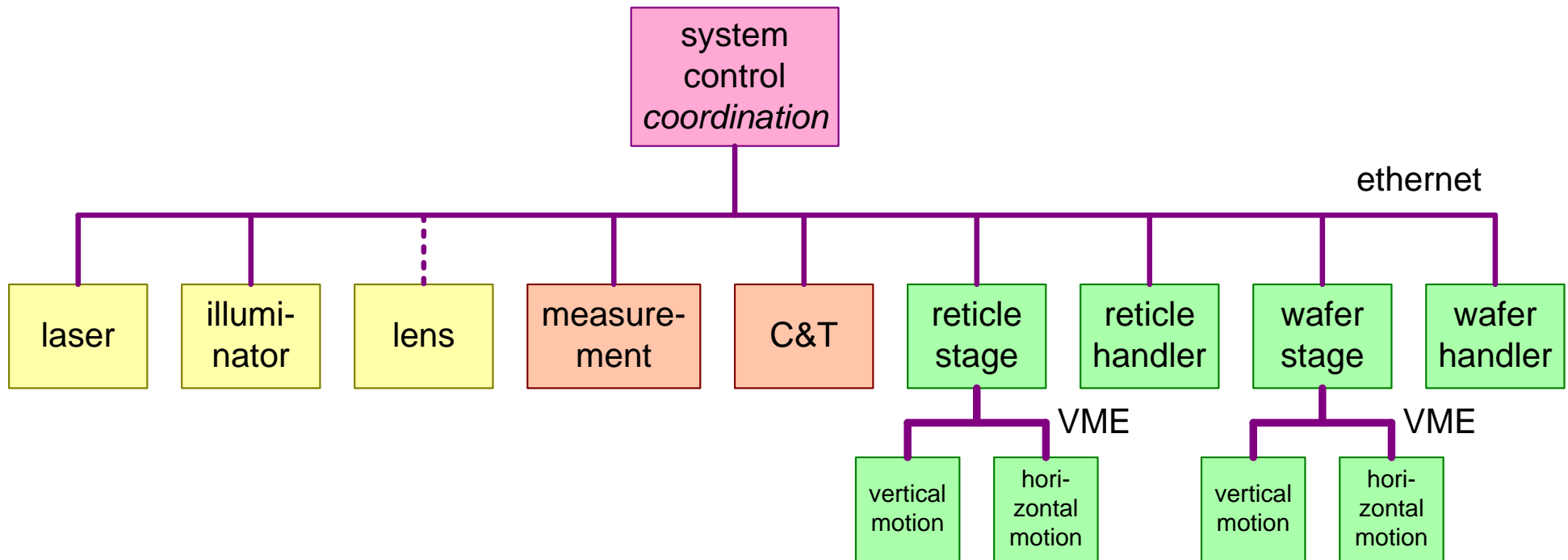
Exercise 4, 2 minutes

How many pages are in your Software Requirements Specification?

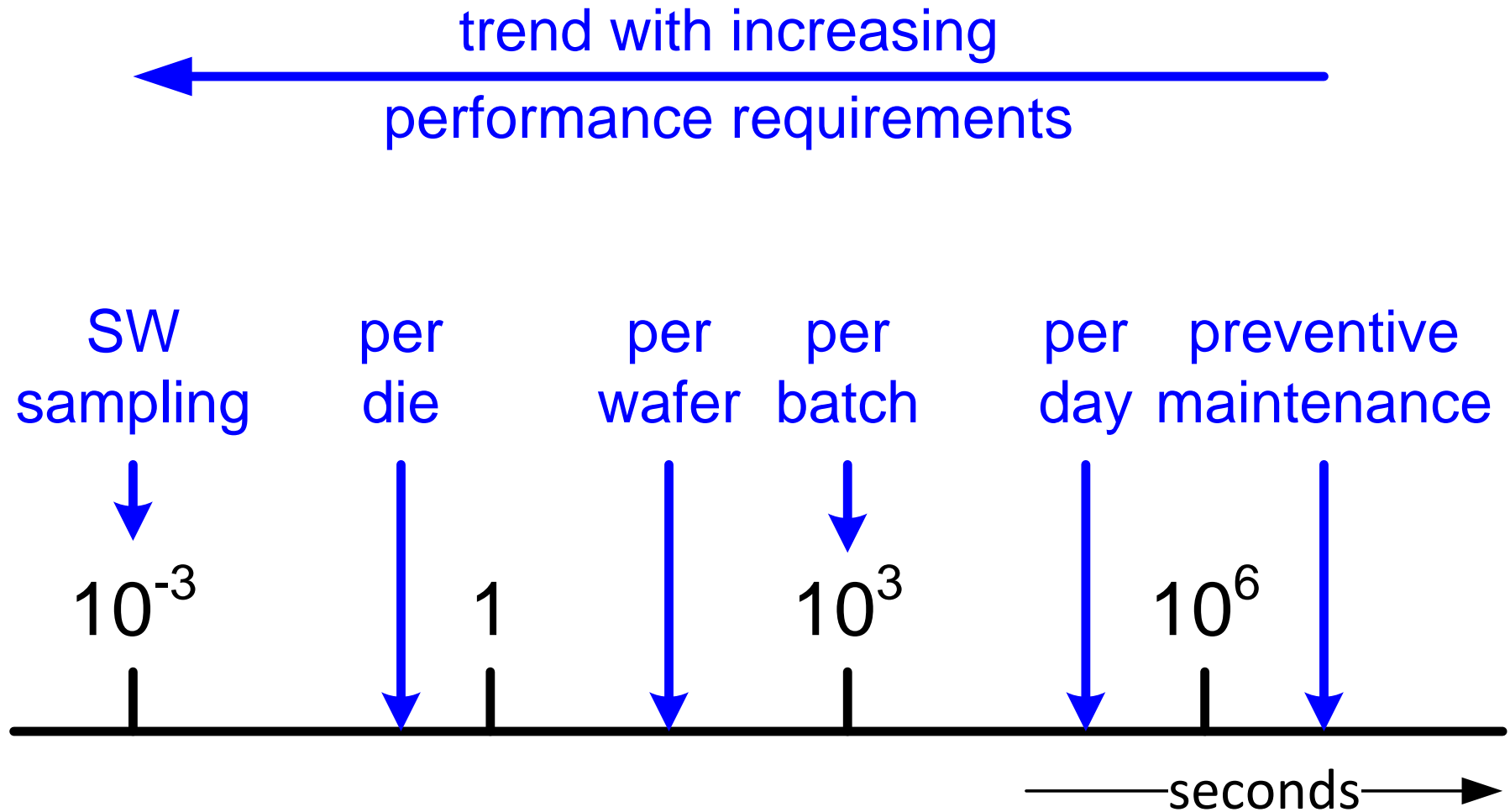
Block Diagram of a Waferstepper



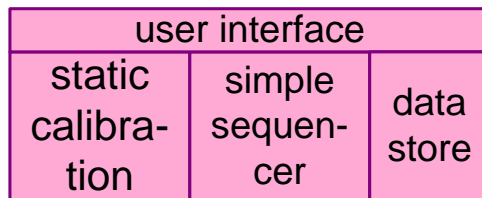
Control Hierarchy of a Waferstepper



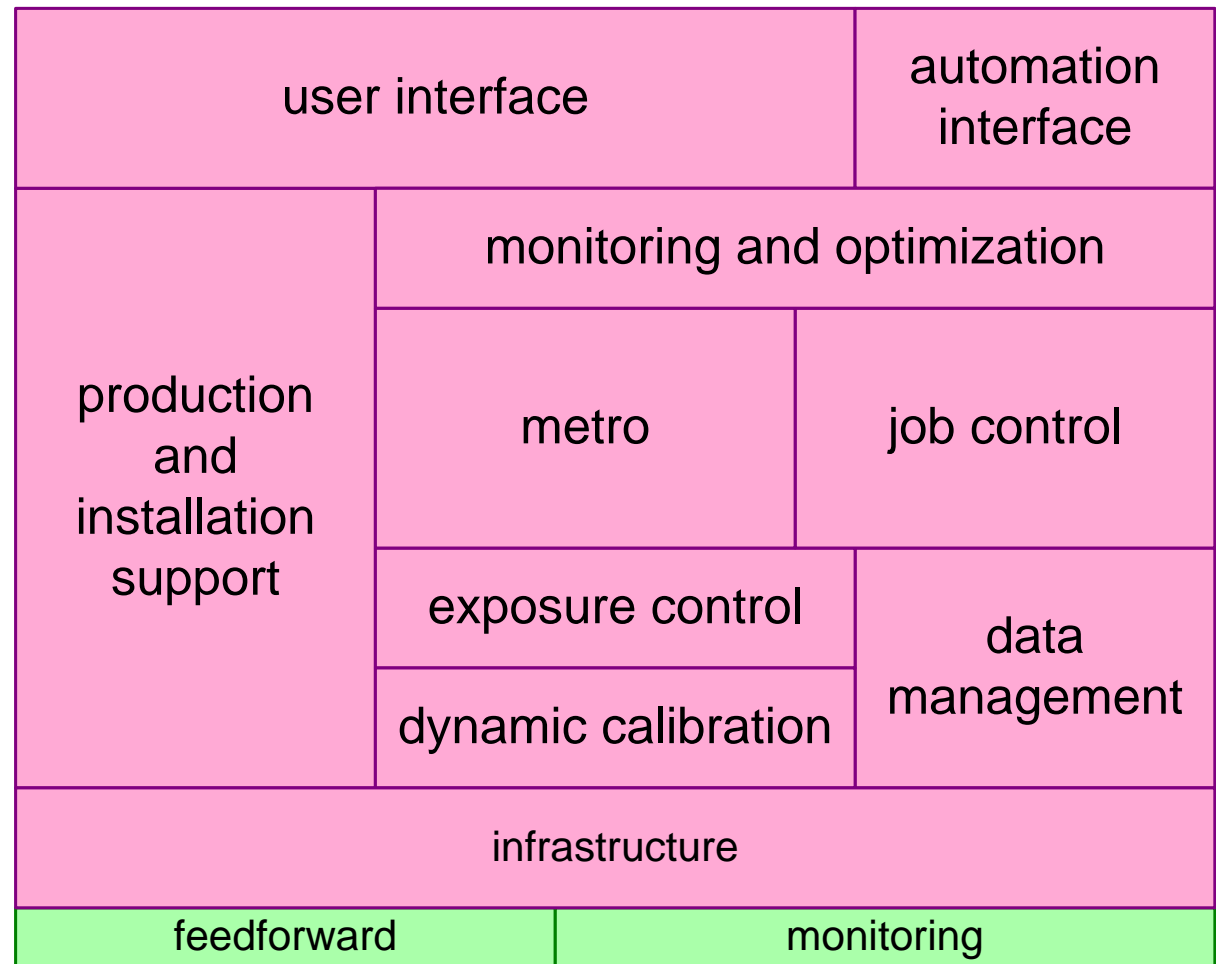
Frequency of Control Actions



Evolution of System Control

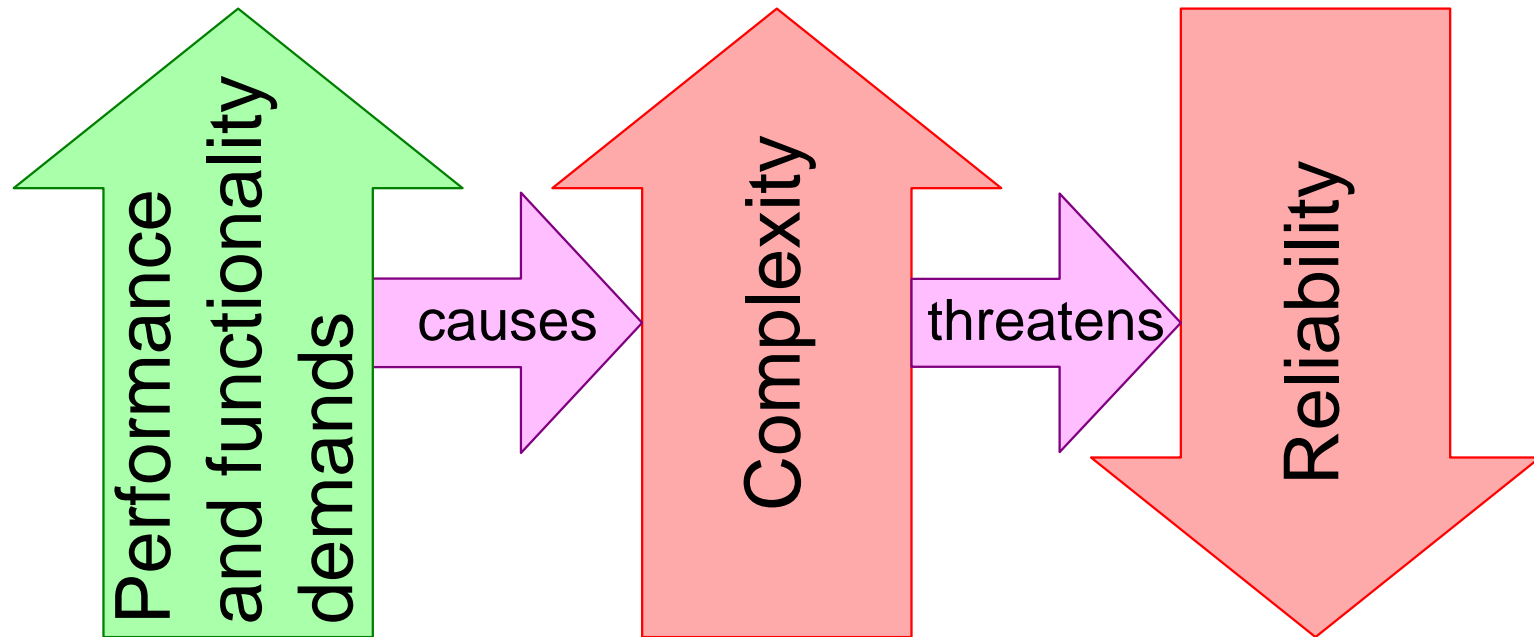


1990
150 kloc



2000
2000 kloc

Consequences of Evolution



loss of overview (150kloc fits in 1 mind, 2Mloc not)
(more than?) exponential increase of coupling
1:1 relation HW:SW becomes n:m relation



Exercise 5, 10 minutes

Visualize the (SW) evolution of your system. What is your current phase?

Different Focus of Software and System

System engineering focus

qualities
productivity
image quality
reliability

→ *concepts*
domain requirements
models

concerns
integral design (quality, balance)
system context
lifecycle
operational processes

education
principles
heuristics
analysis and synthesis
processes

SW engineering focus

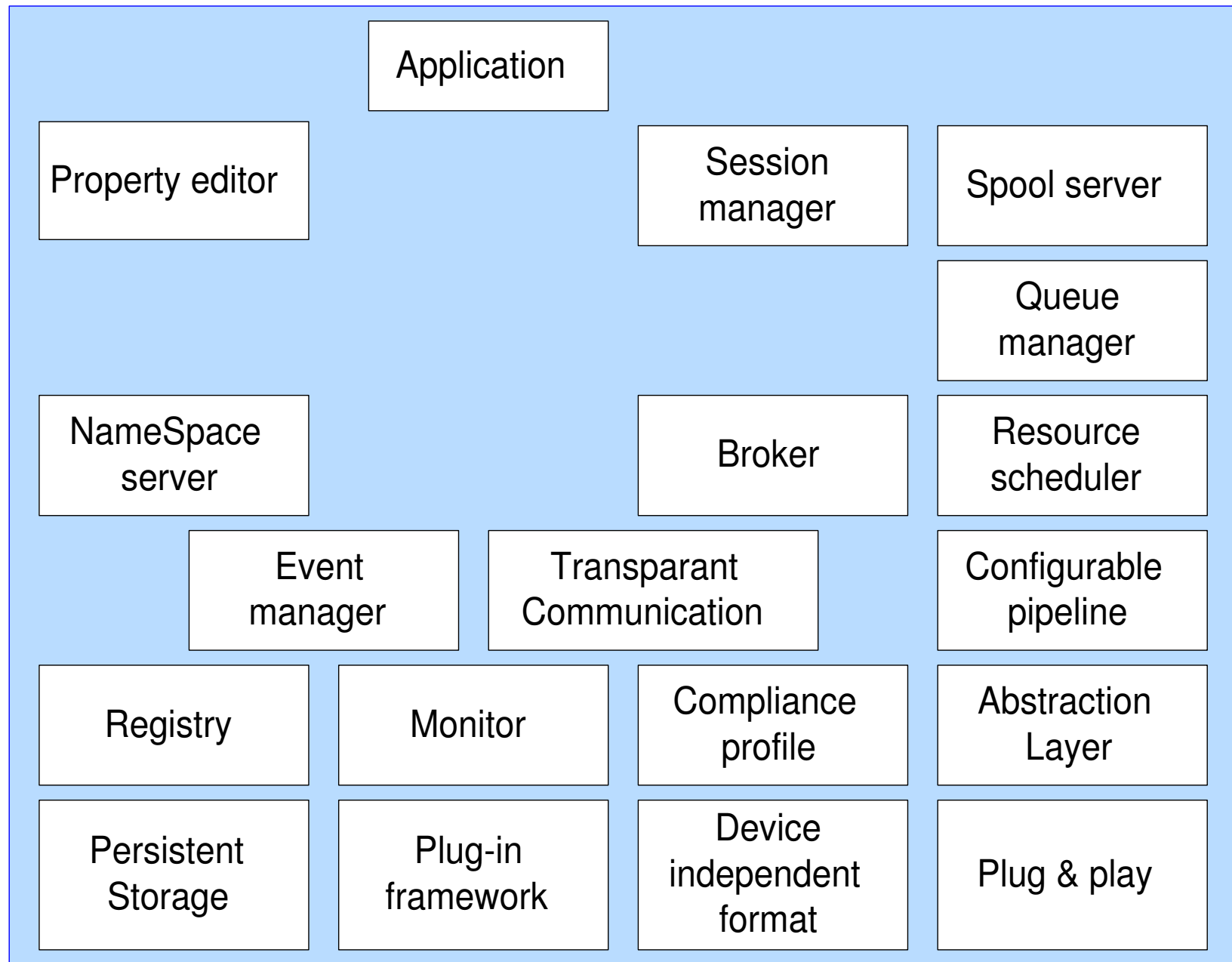
qualities
functionality
maintainability
variability

→ *concepts*
structure
(generic) mechanisms

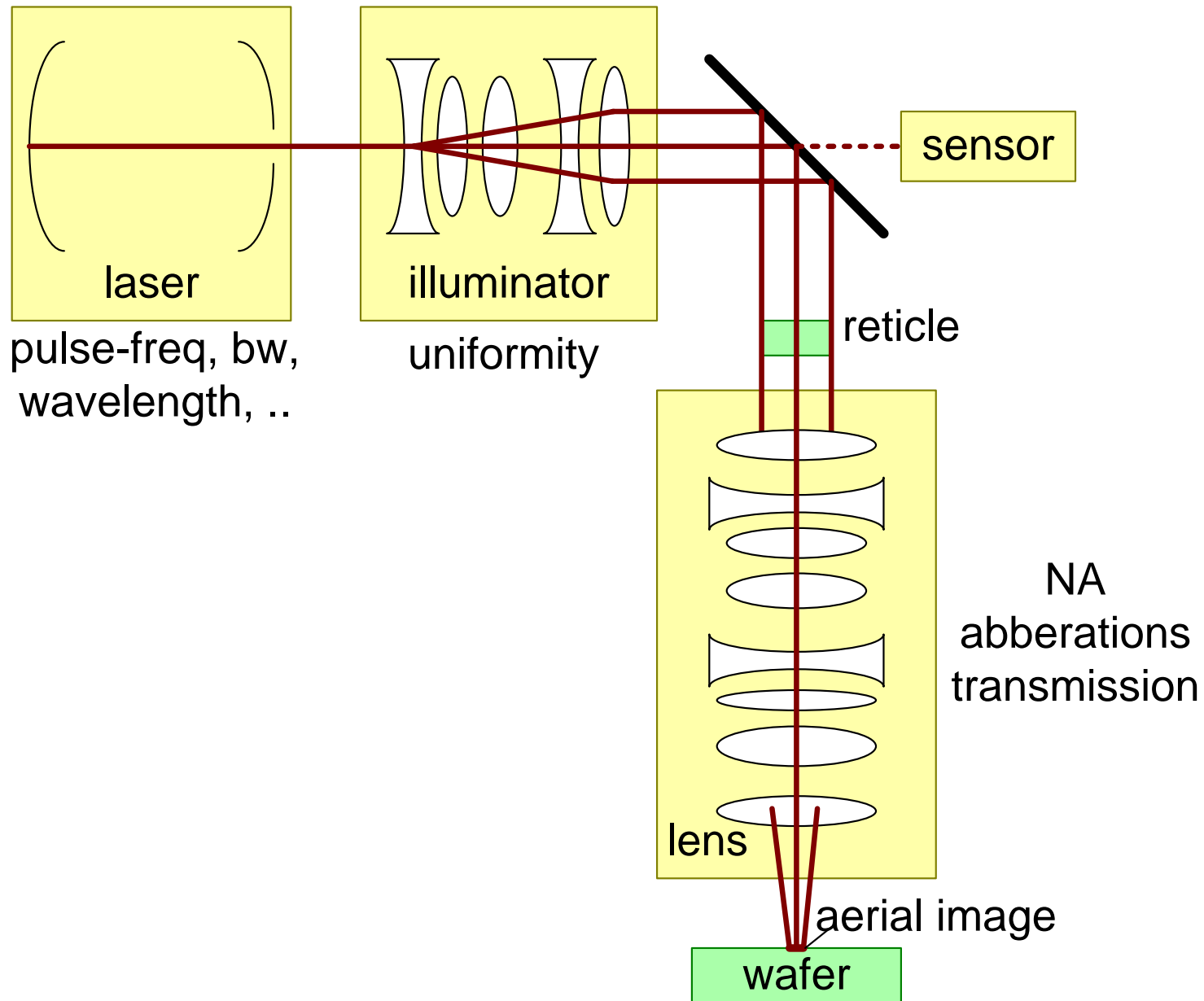
concerns
configuration management
release procedure
tools
SW processes
SW problems, change requests

education
languages
operating systems
algorithms
formal methods

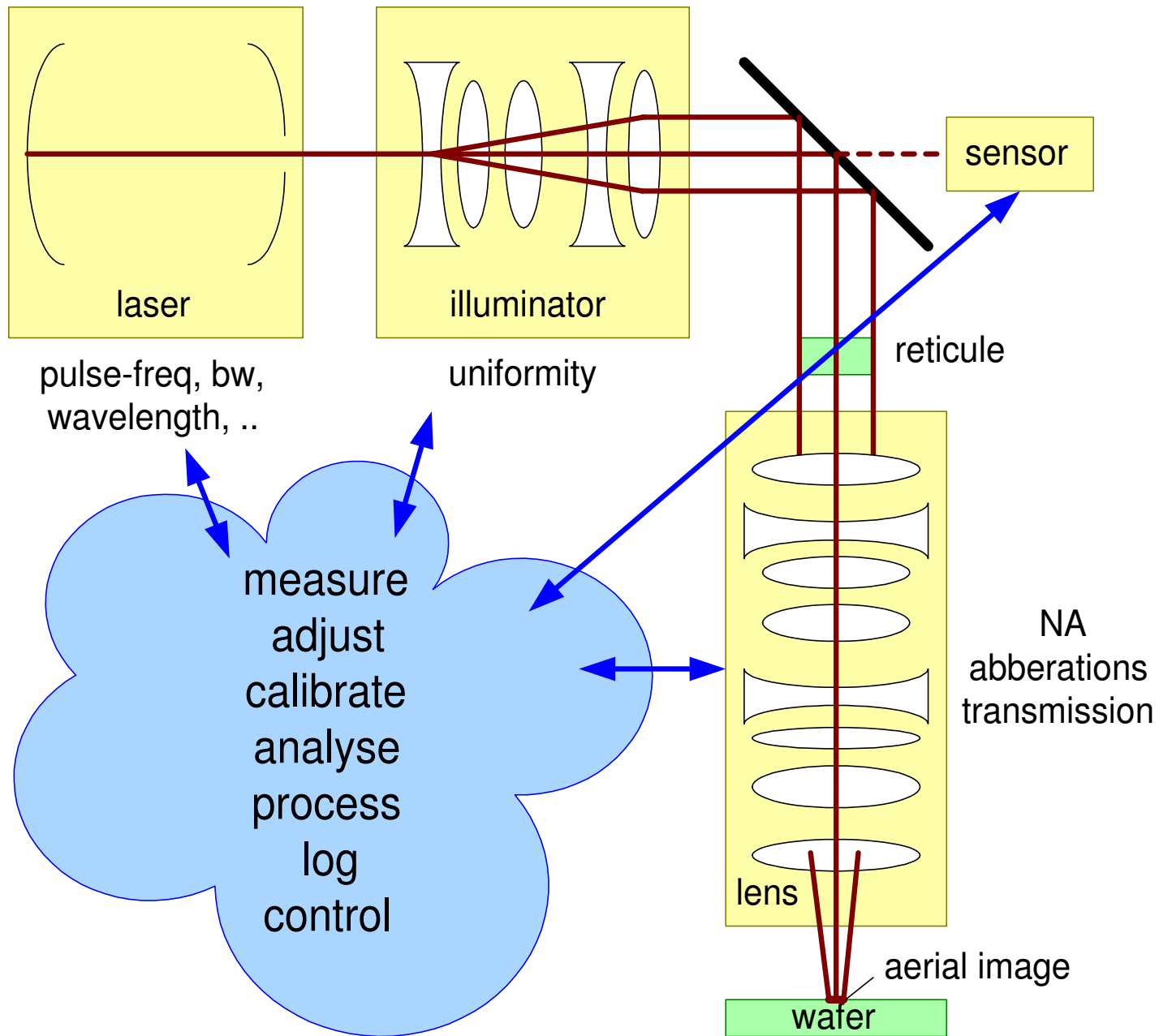
Caricature of a SW Architecture



Caricature of Physics Systems View



Relation SW and Physics



Symptoms of too isolated SW efforts

symptoms

SW people are clustered together

SW is alpha tested before system integration

SW team uses own specification and design process

SW specification is in SW jargon or formalism

counter measures

colocation per function, subsystem or quality

continuous system integration

higher level processes are shared

interaction between SW,
HW and system engineers

Exercise 6, 5 minutes

What is the degree of integration or isolation of SW in your organization?

Different Mindsets and Characteristics

