Less Heavy Systems Engineering; How Much is Approriate?

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

Abstract

Many companies are aware of opportunities to improve systems development, system integration and complex project execution. Conventional Systems Engineering from the military and aerospace domain, although perceived as useful, also tends to be seen as “heavy” in terms of process and artifacts. In this paper we explore alternative Systems Engineering approaches that are perceived as lighter. We also explore how much Systems Engineering is appropriate.

June 6, 2011
status: preliminary draft
version: 0

At the Beginning of this Century

Spring 2000, preparing key-note for conference

Let's go for Light Weight Processes

You cannot be serious
You do not want Light Weight Architecting

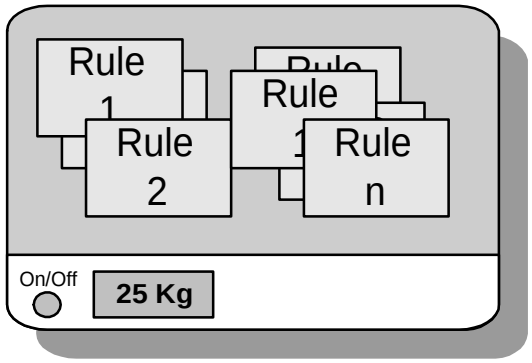
Oh yes, absolutely,
Light Weight Architecting is what
we need

Process Improvement Manager

Architect

Architecture Weight

$$\text{weight(architecture)} = \sum_{\text{all rules}} \text{weight(rule)}$$



weight (rule) = f (level of **enforcement** ,

scope (impact) ,

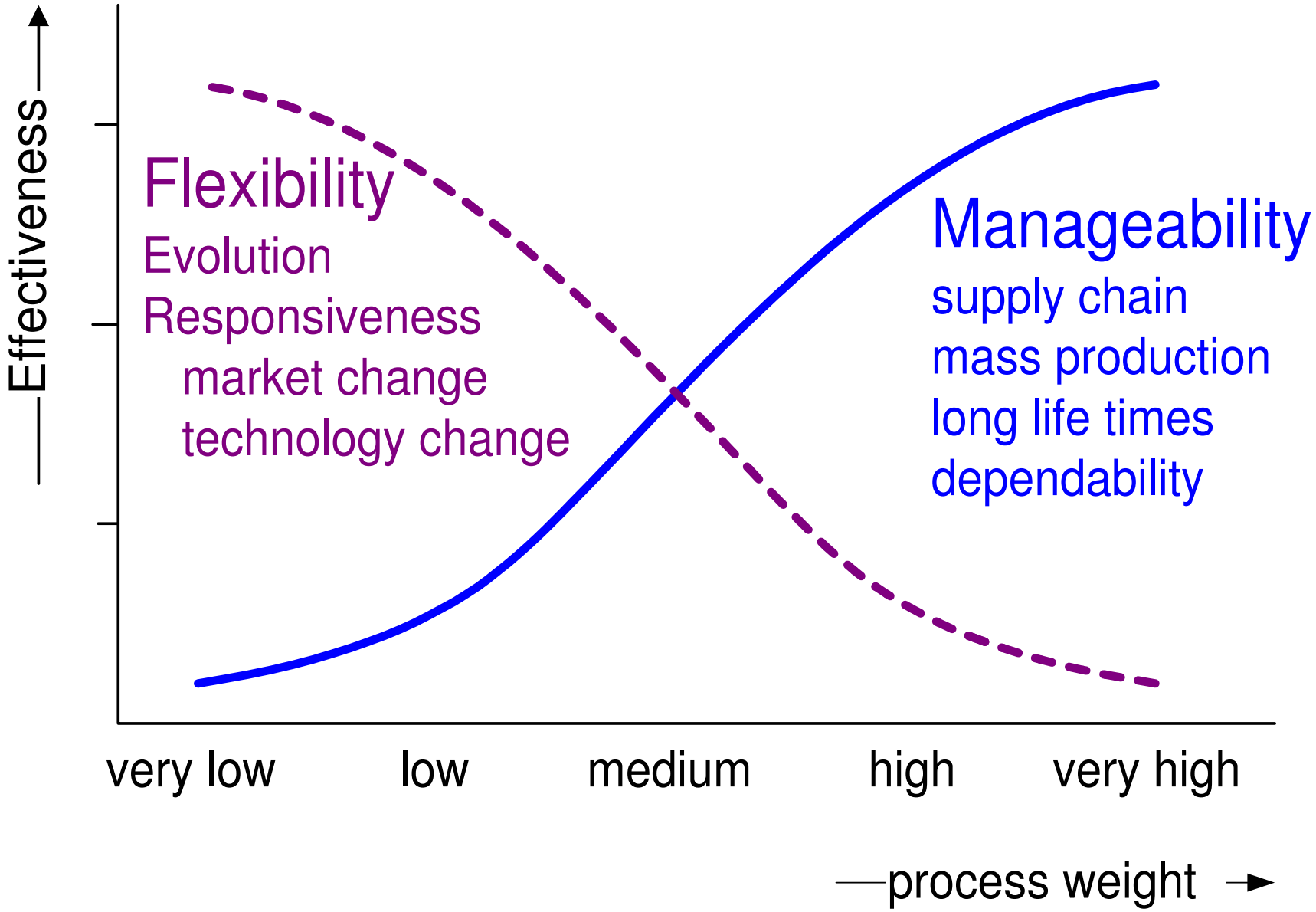
size,

level of **coupling** or number of dependencies)

guideline	conditional rule	mandatory rule
component	product	portfolio
single-line	multi-line	multi-page
stand-alone		builds on many rules

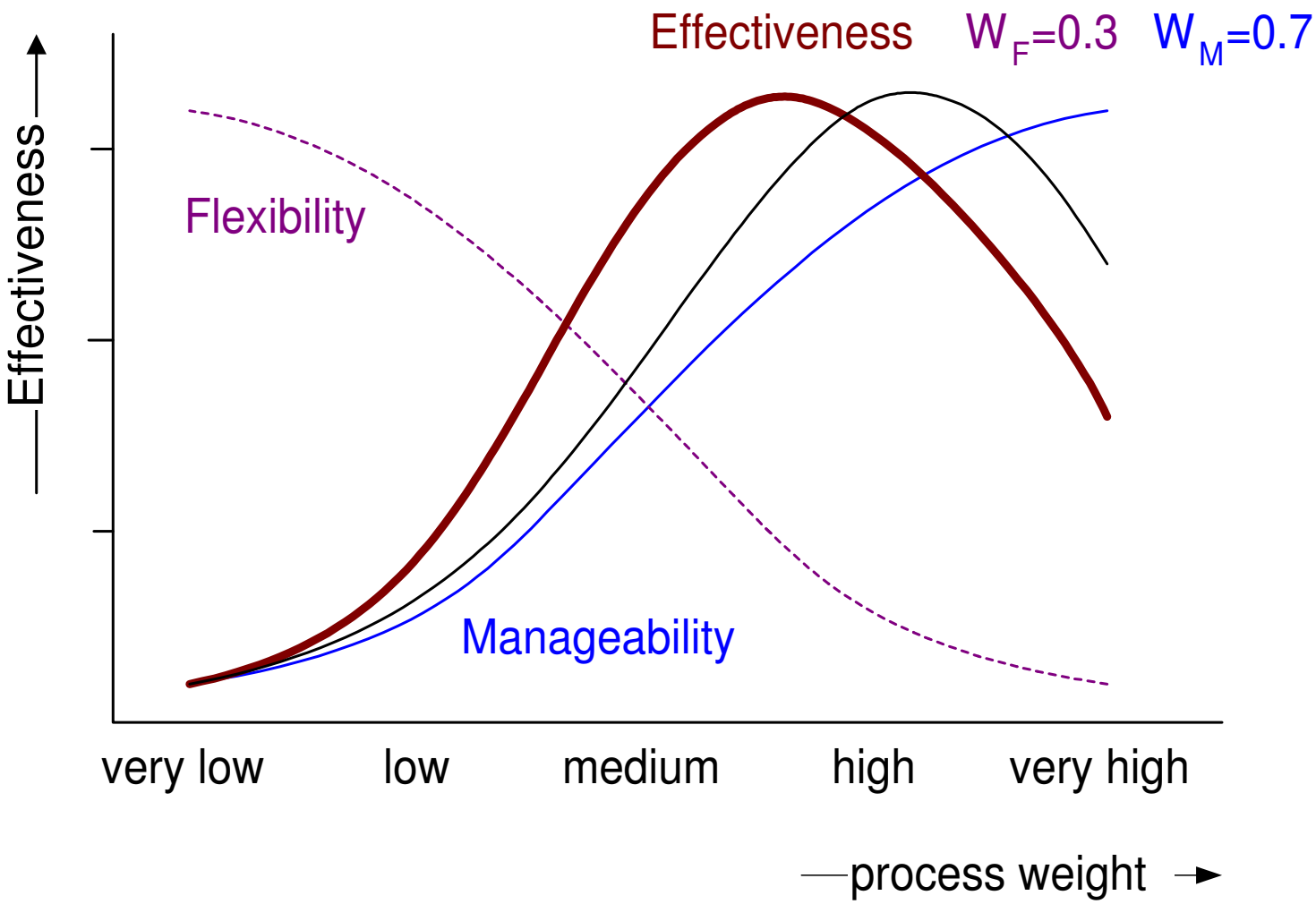
← low —— weight —— high →

Effectiveness(Flexibility, Manageability)



Effectiveness

$$\text{Effectiveness} = \text{Flexibility}^{W_F} * \text{Manageability}^{W_M}$$



Light Weight How To

$$\text{weight(architecture)} = \sum_{\text{all rules}} \text{weight(rule)}$$

2. Minimize the weight per rule

1. Reduce the rule set to the (business) essential

Understand
- your customer
- your customer's customer

etcetera

Minimize Rule Weight

weight(rule)=

f (level of **enforcement** ,

scope (impact) ,

size,

level of **coupling** or
number of dependencies)

minimize number of mandatory rules

empower, delegate

minimize implementation details
focus on essential concepts

Apply design principles on architecture

Multi-view architecting

Simplified Framework

Effectiveness (Customer Value)

Do the right things

What methods increase (understanding of) Customer Value?

What can you use in your own company to increase

(understanding of) Customer Value?

Efficiency (Effort, cost, and time per result)

Do things right

What methods improve the efficiency of the company?

What can you use to improve the efficiency of your company?

Work Form for KSEE 2011

	Effectiveness (Customer Value) *Do the right things* What can you use in your own company to increase (understanding of) Customer Value?	*Efficiency (Effort, cost, and time per result)* *Do things right* What can you use to improve the efficiency of your company?
Håkan Gustavsson Is it Lean or just common sense?		
Einar Jørgensen Globalising System Engineering and Lean Principles		
Odd Guldsten Complex power systems for offshore oil&gas topside installation		
John Bjarne Bye Lean Transformation		
Jon Wade Systems Engineering: At the Crossroads of Complexity		
Andreas Thorvaldsen Manufacturing Systems Modelling		
Kristian Frøvold Early Validation through the A3 method		
Gerrit Muller Less Heavy Systems Engineering; How Much is Appropriate?		

Explanation of Work Form

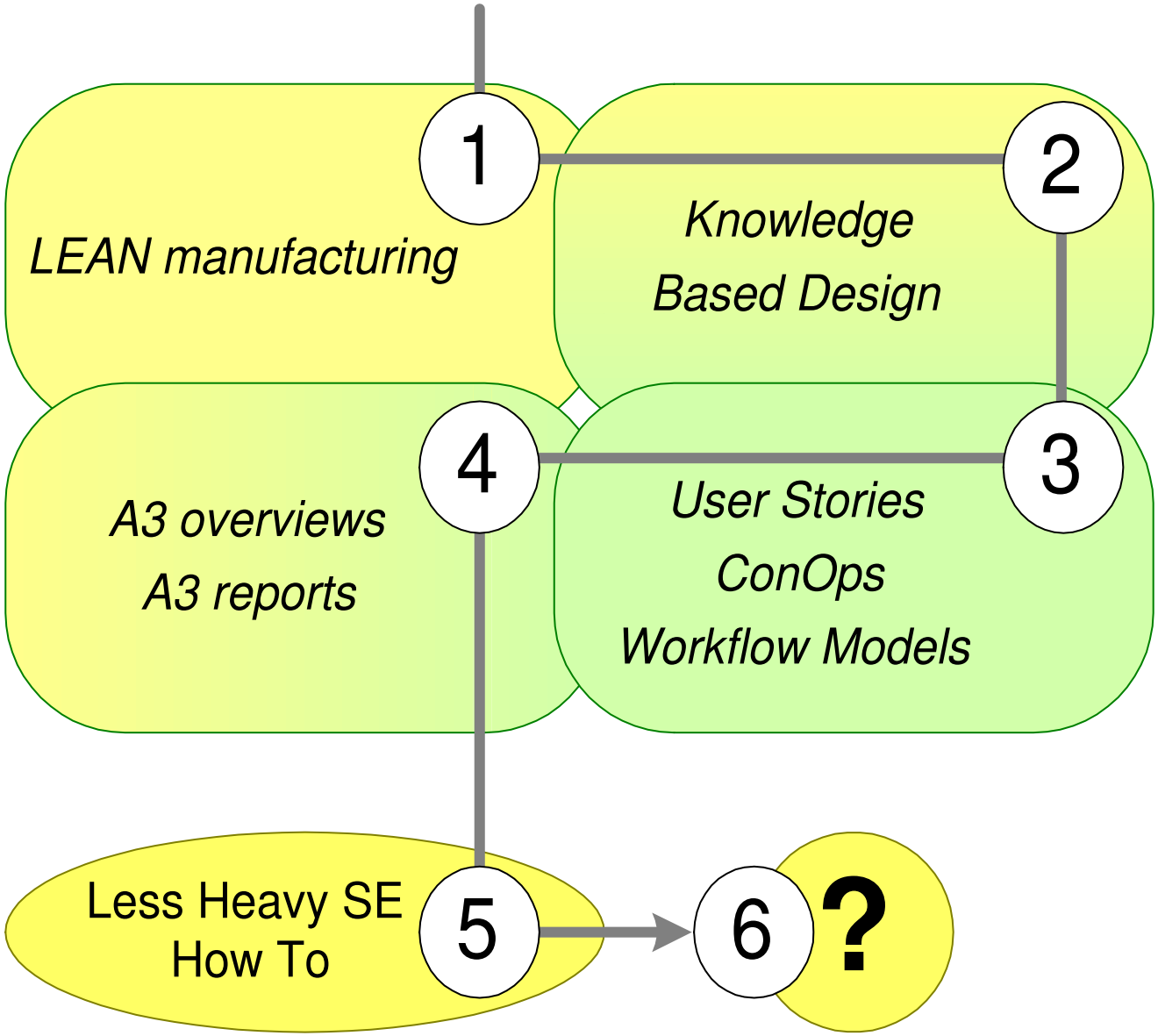
We expect that everyone fills in the form during or at the end of every presentation.

The purpose is to stimulate you to reflect on possible value for your own company.

We recommend to write down specific examples.

The last presentation will look back at at all presentations.

Time to Harvest! Figure Of Contents™



LEAN Manufacturing

Toyota Production System (TPS)

strive for flow

eliminate

- non-value-adding work
- overburden
- unevenness

Just In Time

autonomation

evolved into

LEAN manufacturing

customer value

empowerment

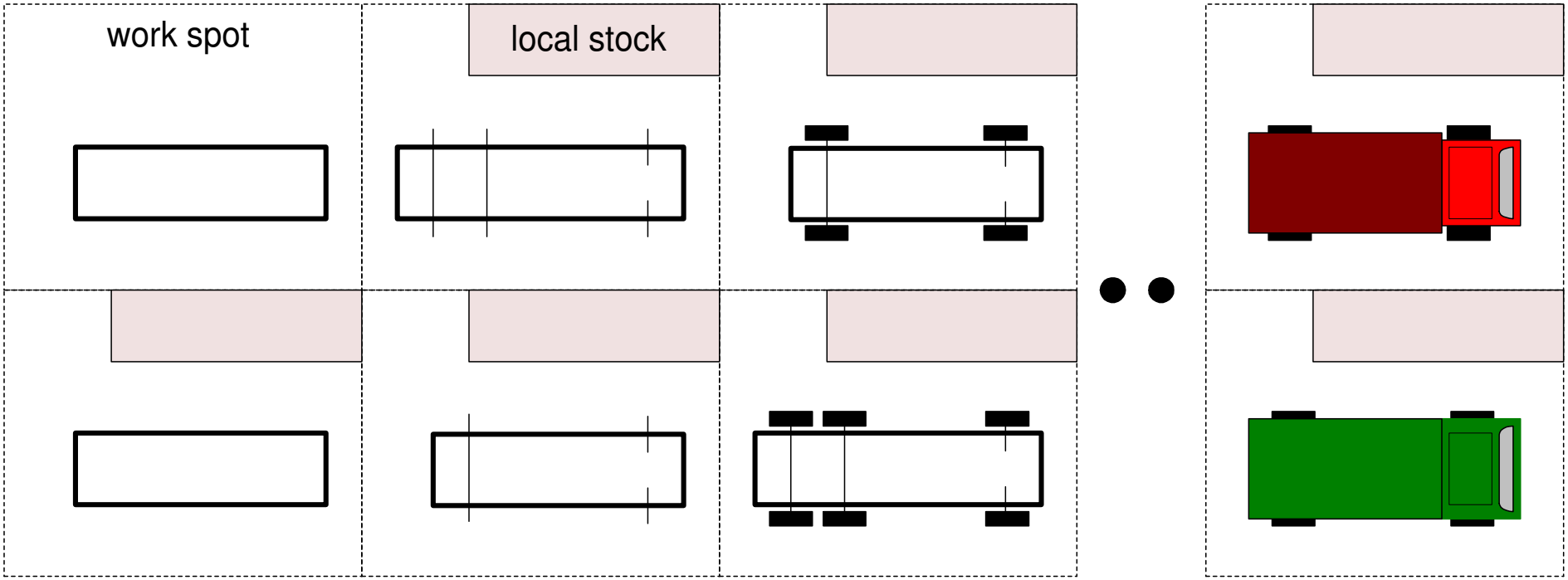
continuous improvement

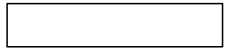
A3 reports

value stream mapping

result: organic flow manufacturing
efficient, flexible, short cycle times

Example of LEAN Manufacturing in Automotive





scheduling
white board

One heart beat

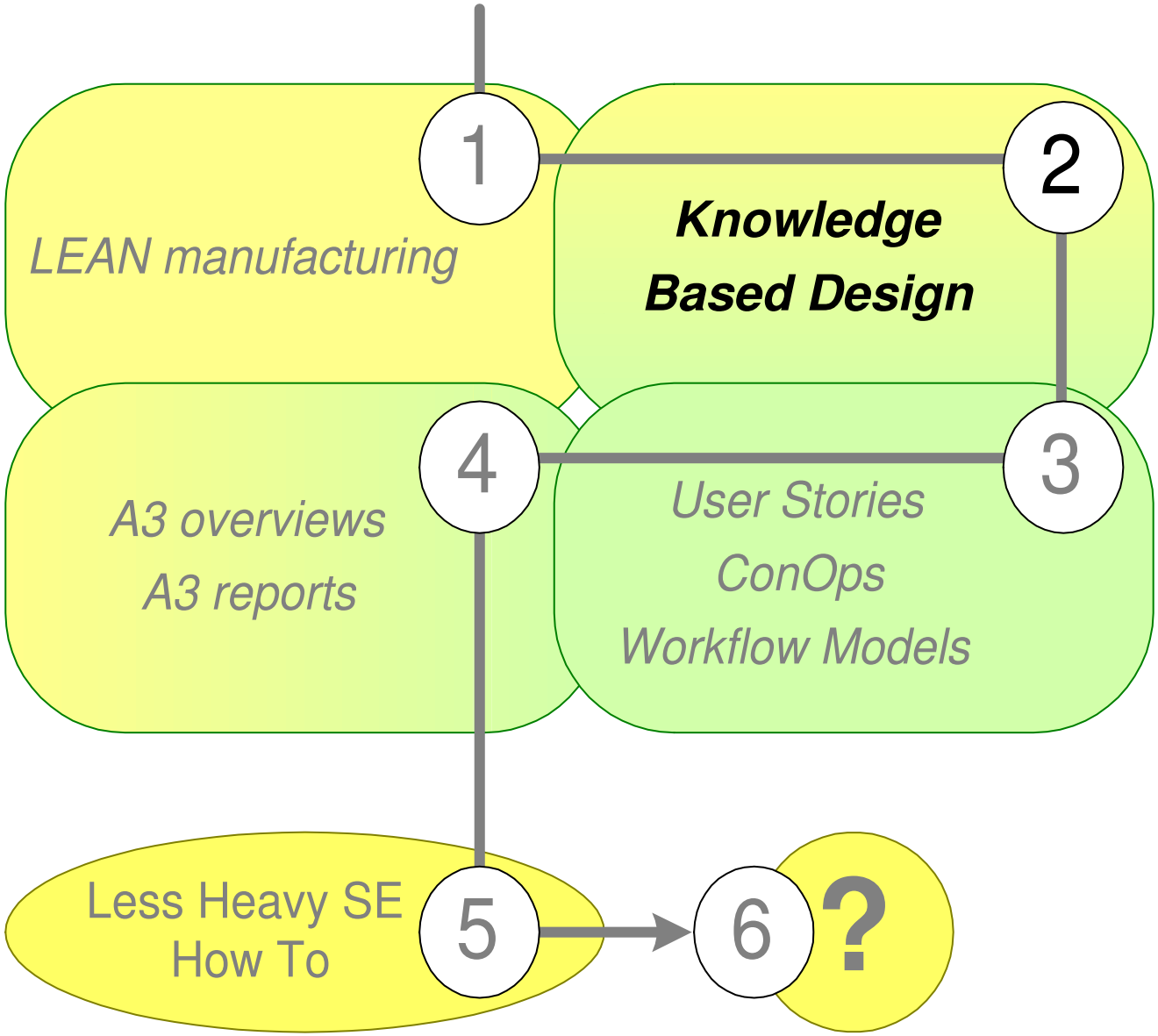
Every truck is unique

Local scheduling

Many practical local solution

by Continuous Improvement

Knowledge Based Design



Knowledge Based Design

LEAN manufacturing

repeatable, production oriented

inspires

LEAN product development (LPD)

creative, development oriented

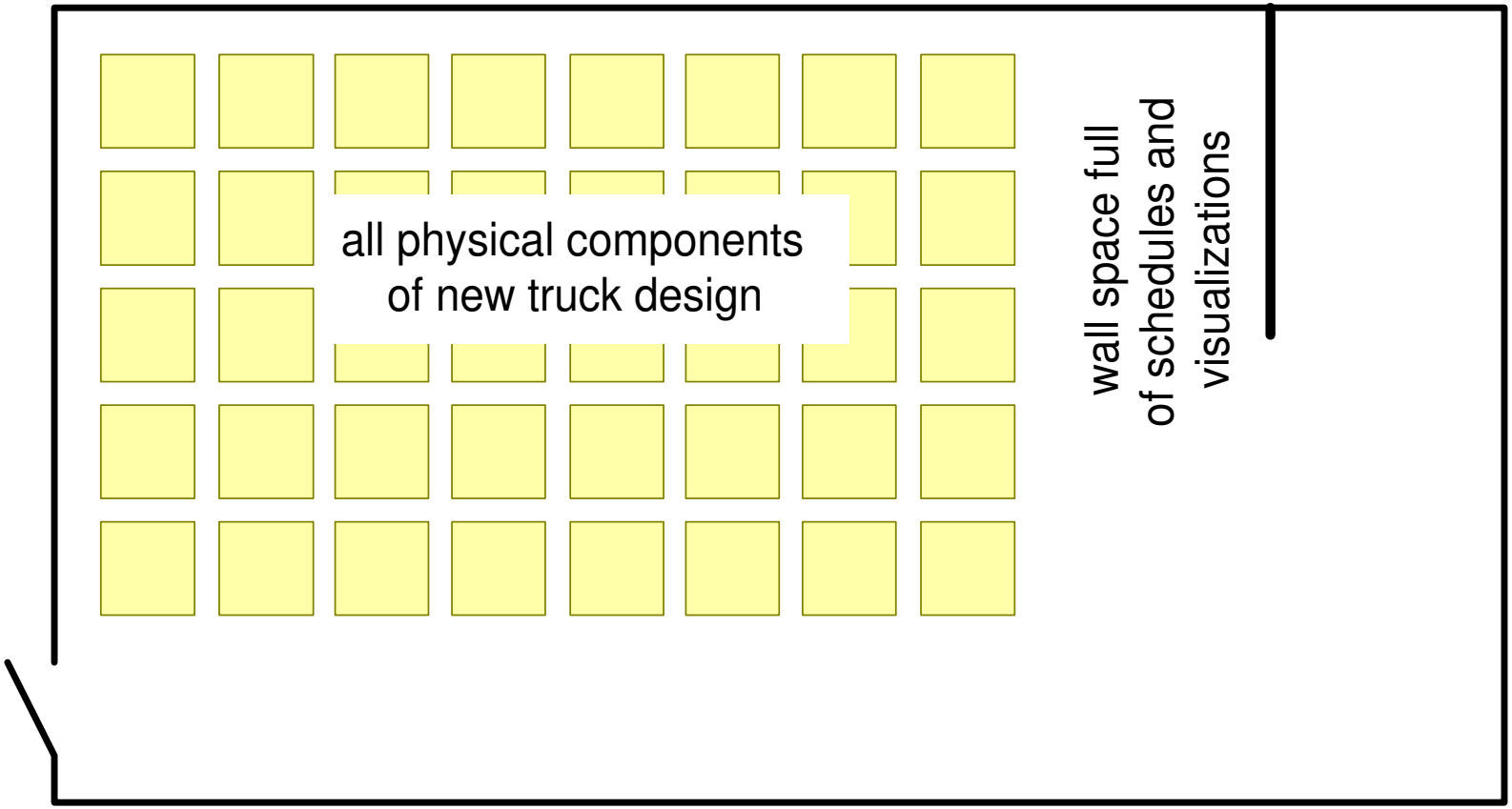
evolves into

Knowledge Based Design

a Norwegian variant of LPD

improving efficiency by (re-)using knowledge

Example of LPD in Automotive



development supported by team location

tactile and visual support

developers drive trucks themselves (customer understanding)

Reflections on Knowledge

Knowledge is abstract and intangible.

is data in a computer knowledge?

are text and figures in a book knowledge?

Value is obtained when knowledge is applied properly.

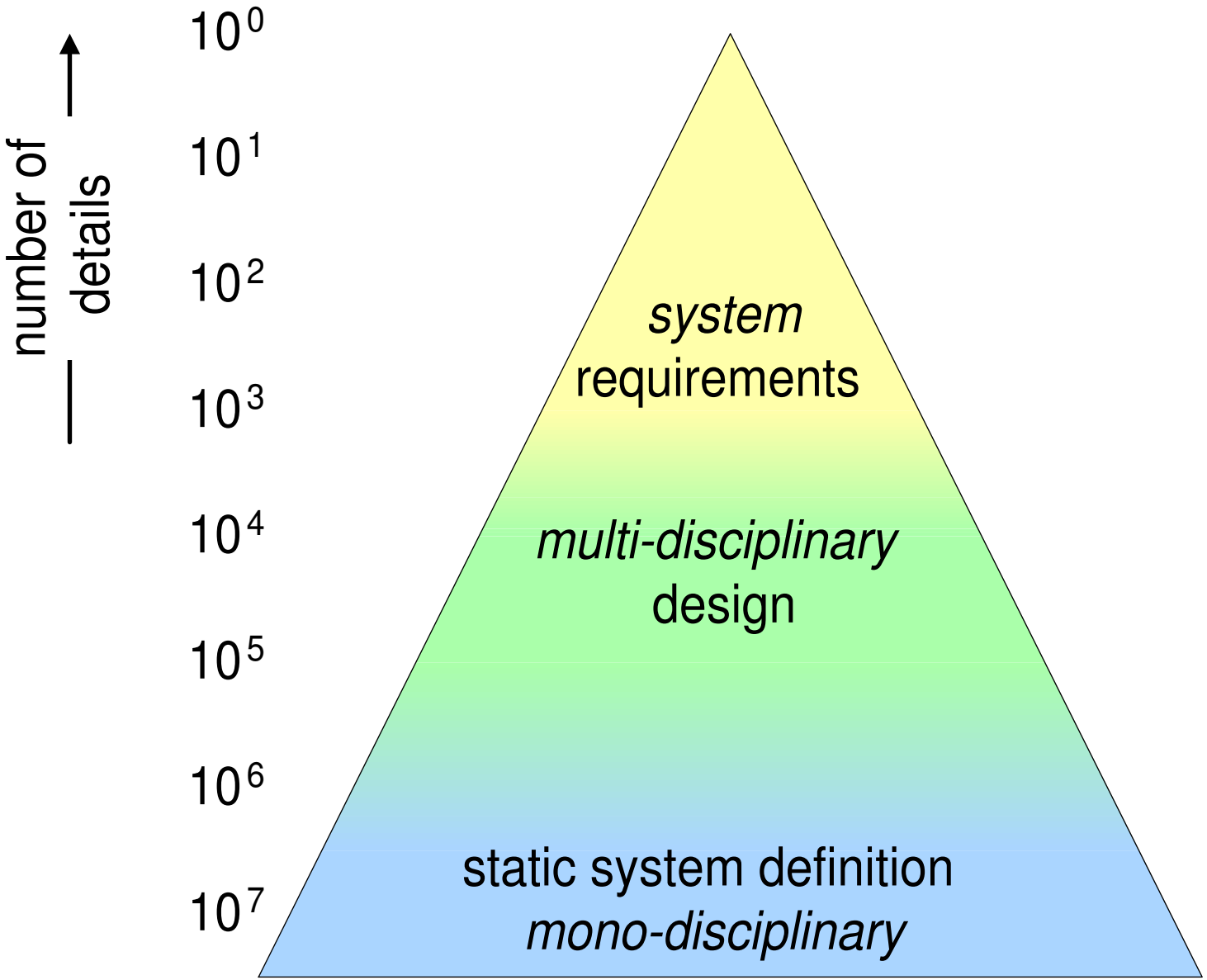
competence = knowledge + skills

Humans need experience to develop skills.

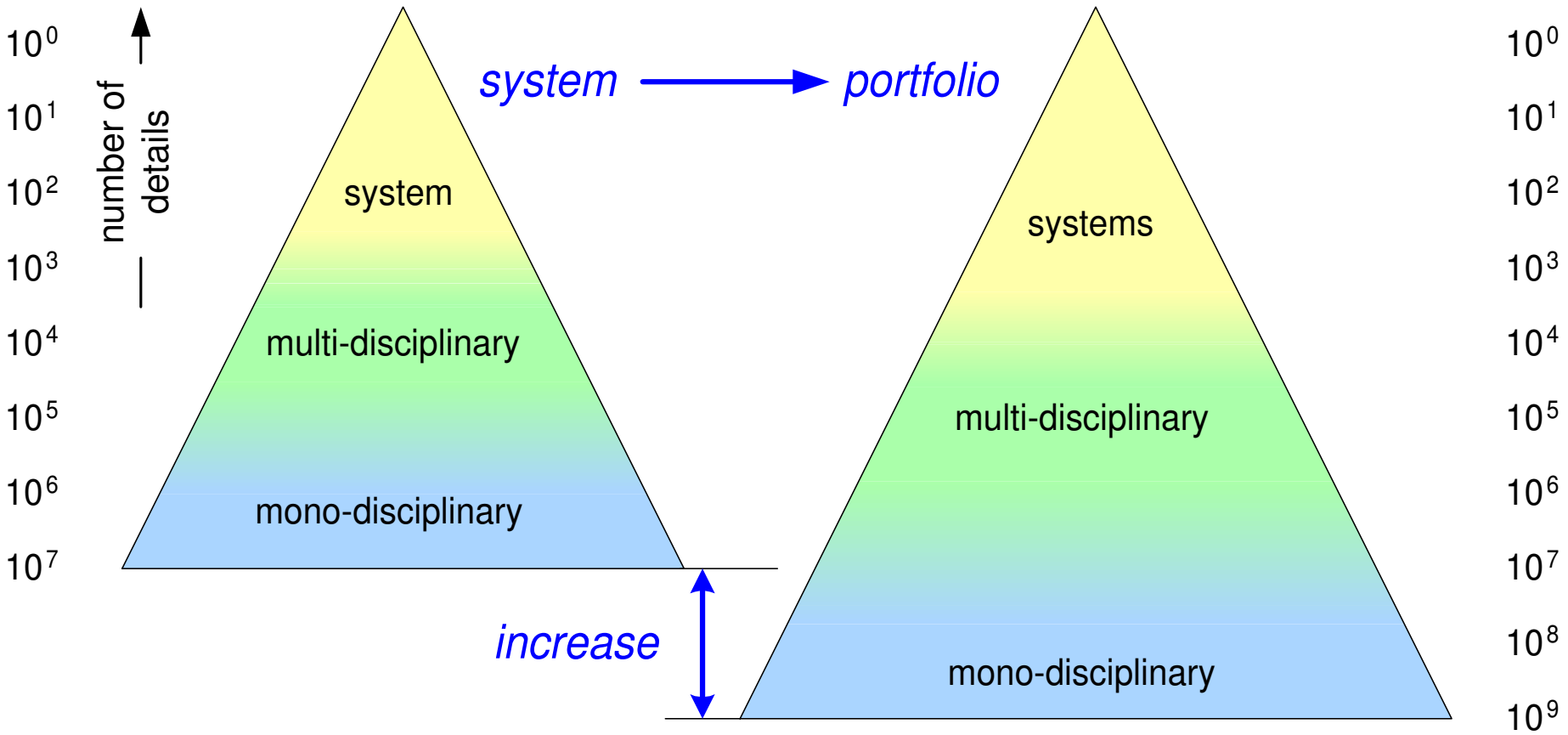
skills are practical, developed by doing

Skills and experience are complementary to knowledge.

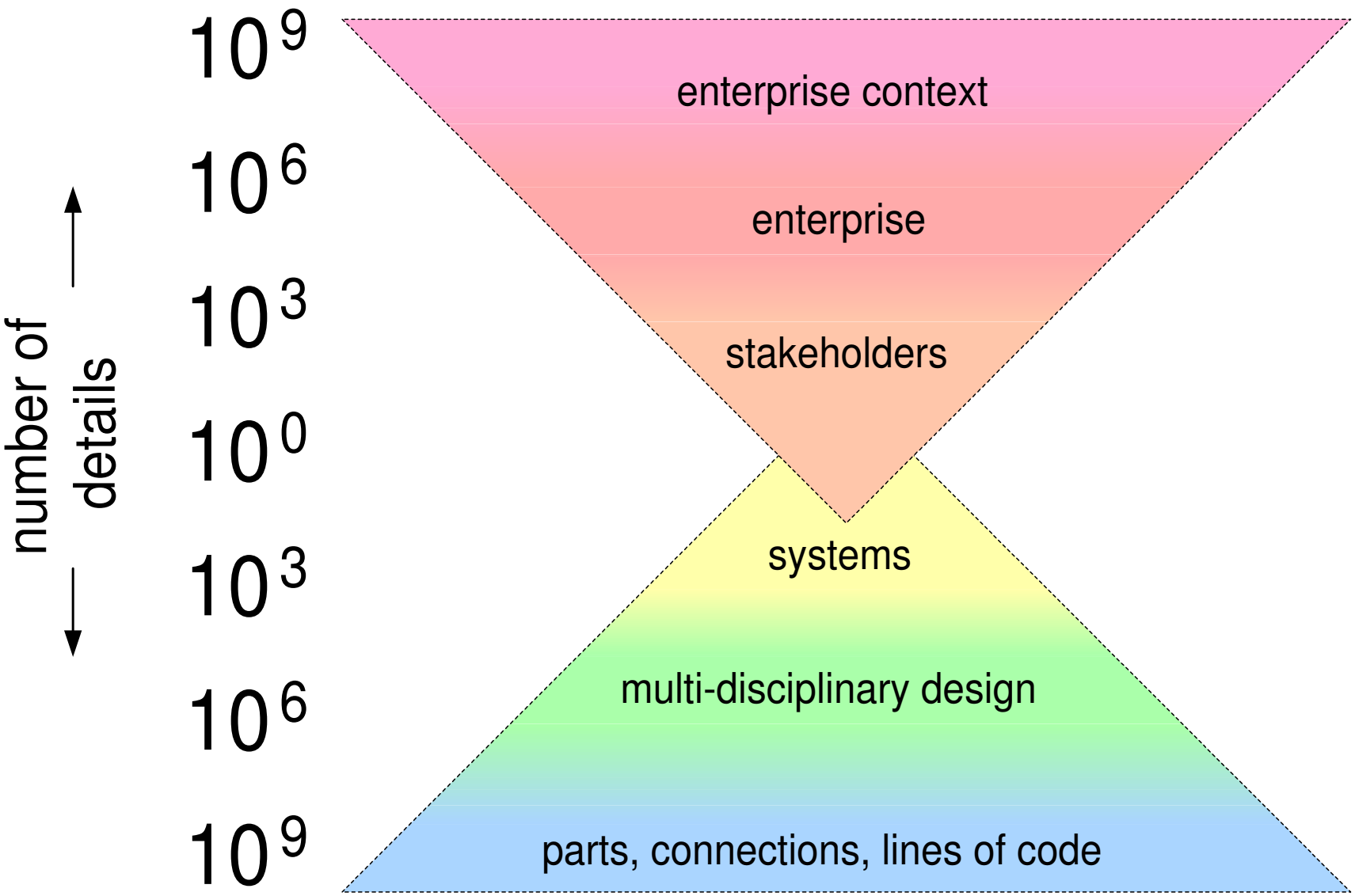
Level of Abstraction Single System



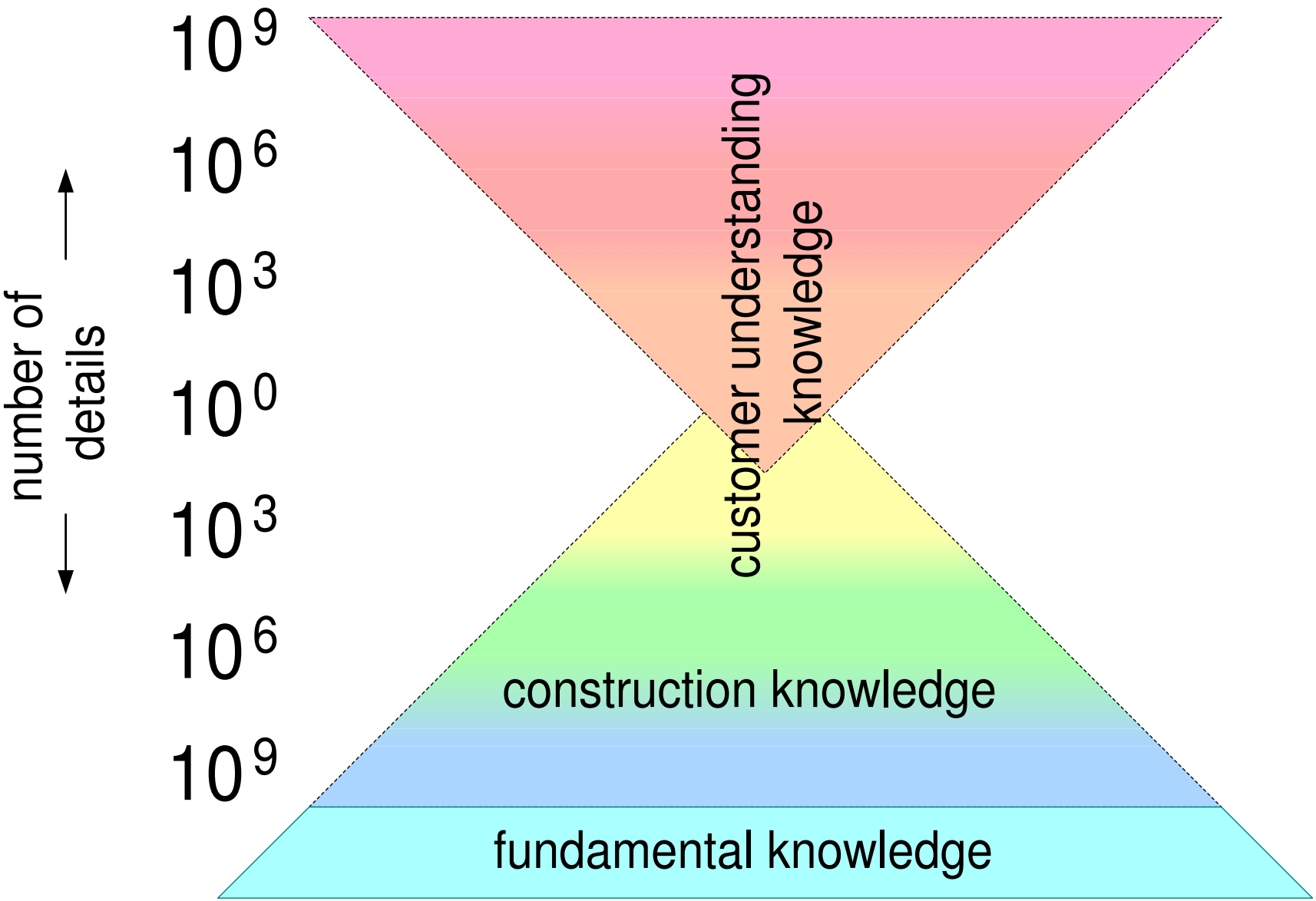
From system to Product Family or Portfolio



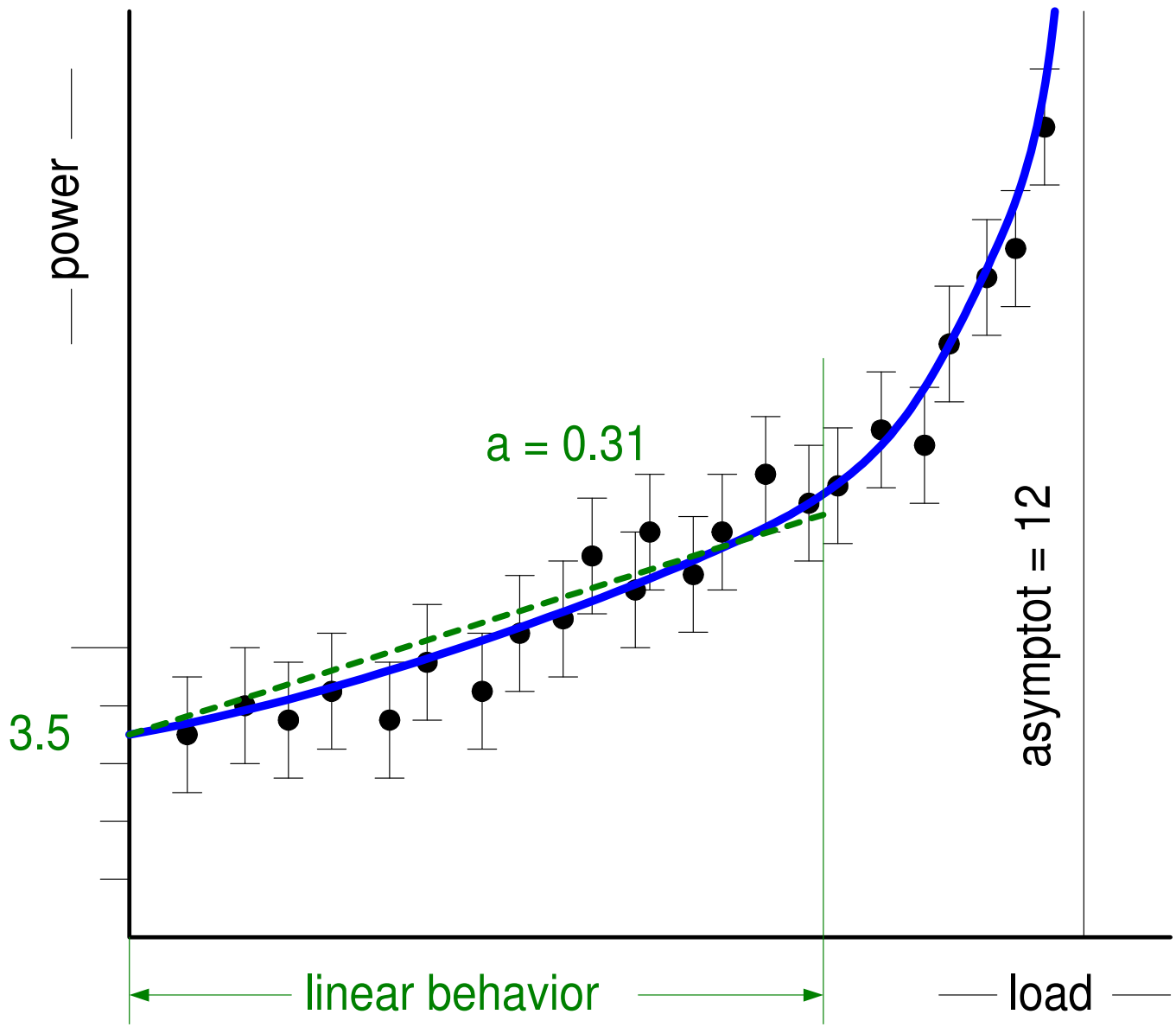
Product Family in Context



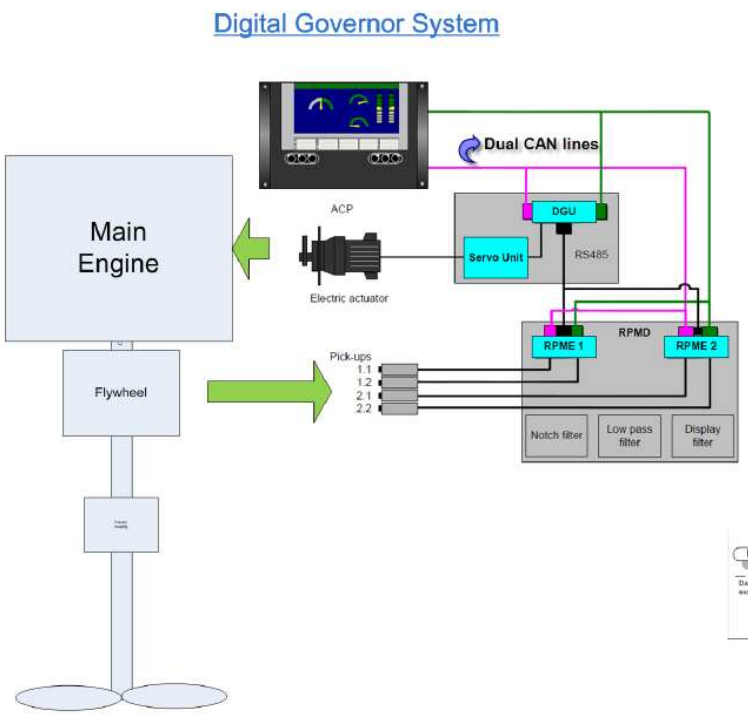
Knowledge at Multiple Levels



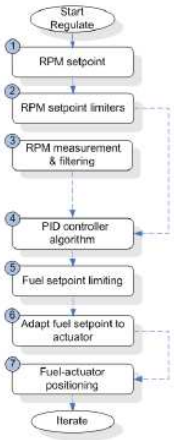
Example of Fundamental Knowledge

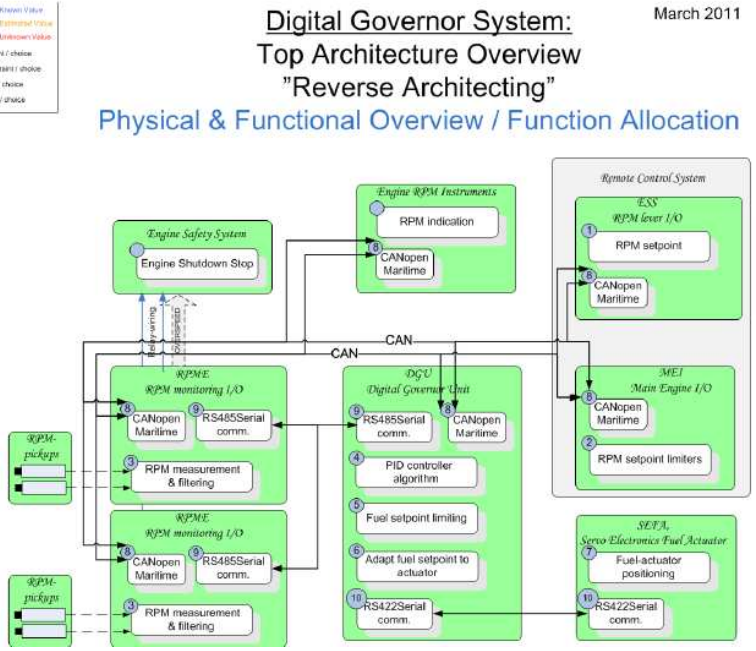


Example of Construction Knowledge



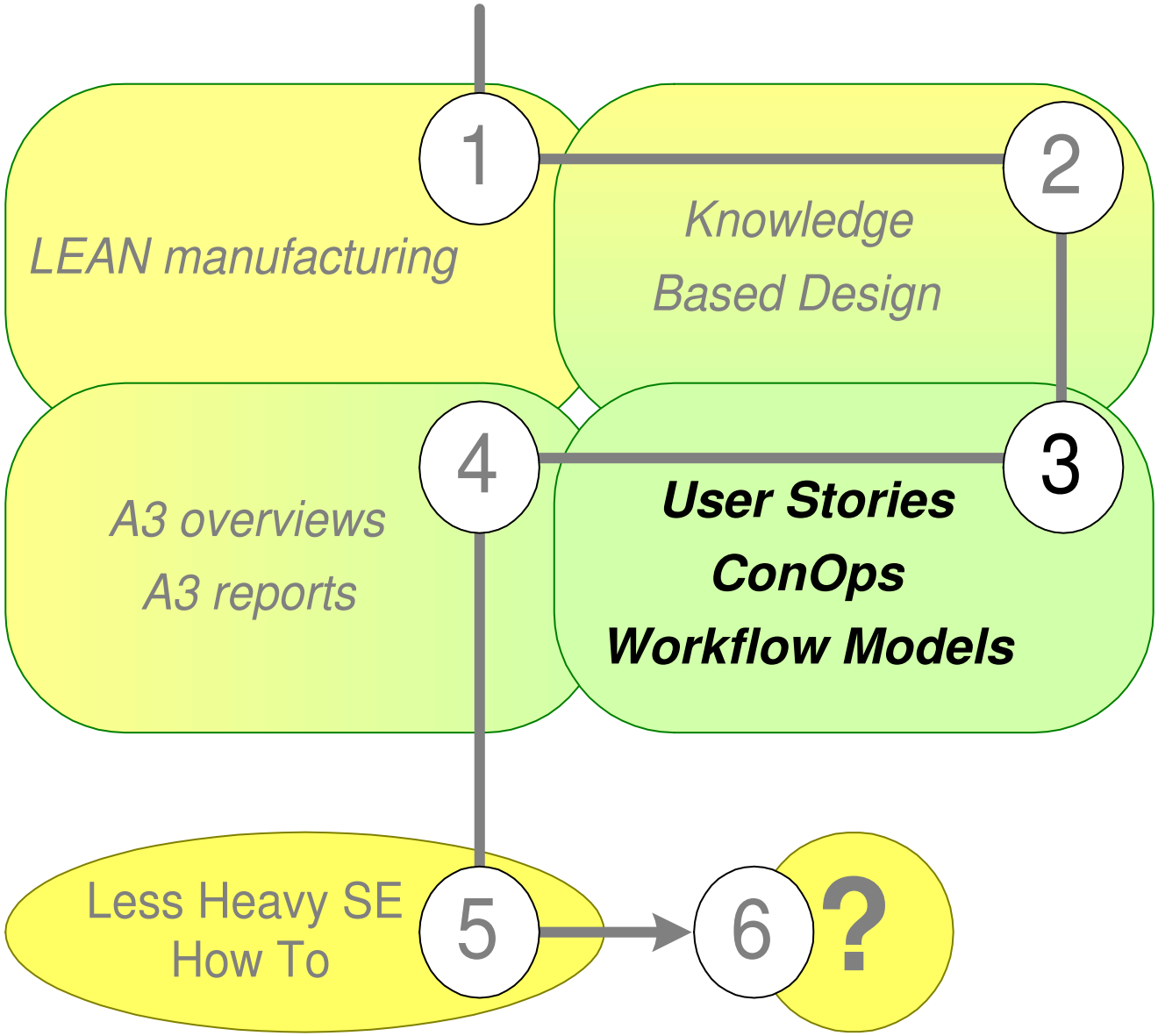
KONGSBERG





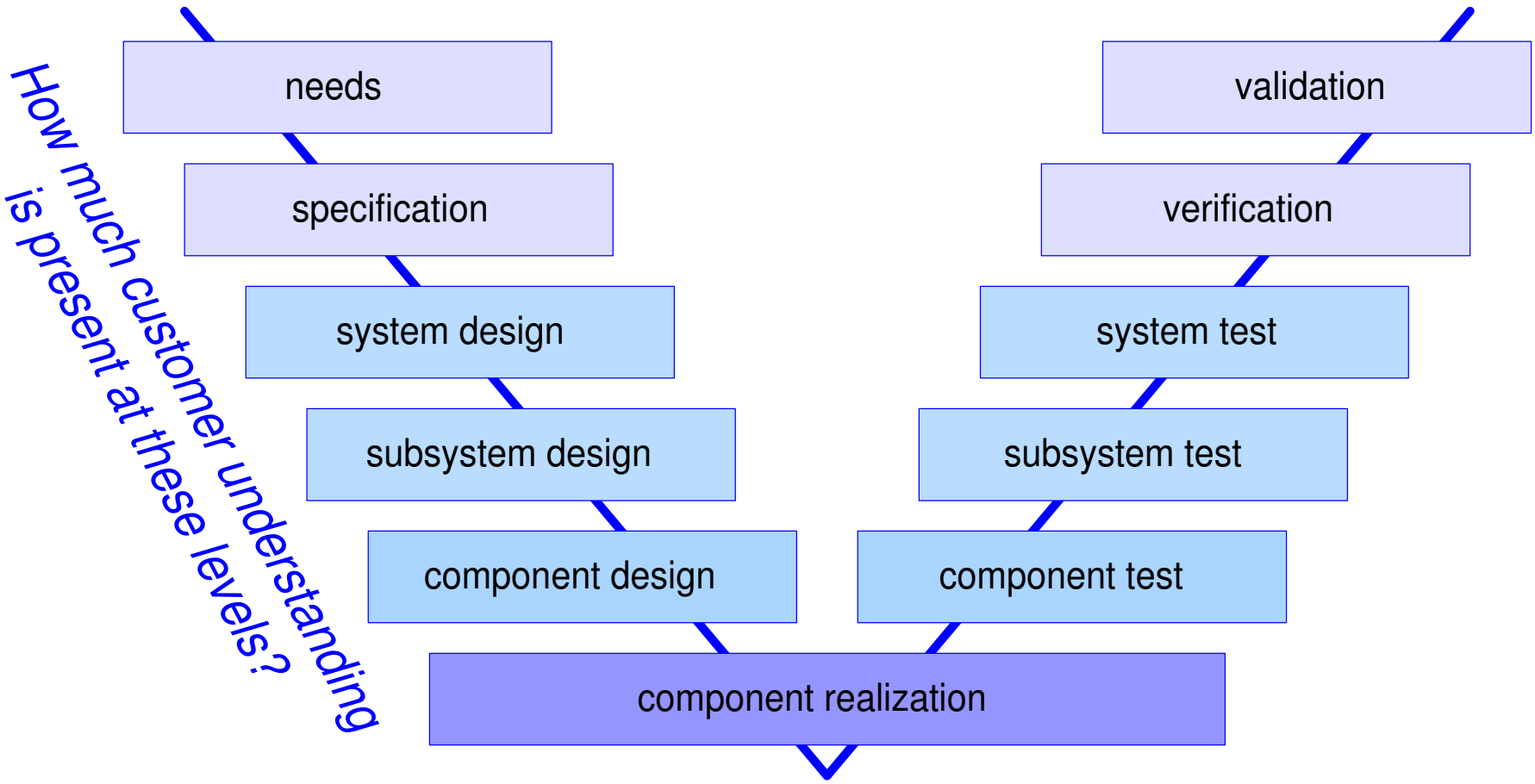
source: Bjørnar Wiulsrød
SESG presentation
March 2011
http://www.gaudisite.nl/SESG_Wiulsr%EF%BF%BDdA3overviews.pdf

Customer Understanding



How well do Your Engineers Understand Your Customer?

In every hand-over and every conversion knowledge is lost



Methods to Capture Customer Understanding

User Stories

Specific stories to explore specification and design.

Contain social and environmental details to make engineers aware

ConOps

Concept of Operations, used in Defense Domain

Factual description of Operational use, a.o. with scenarios

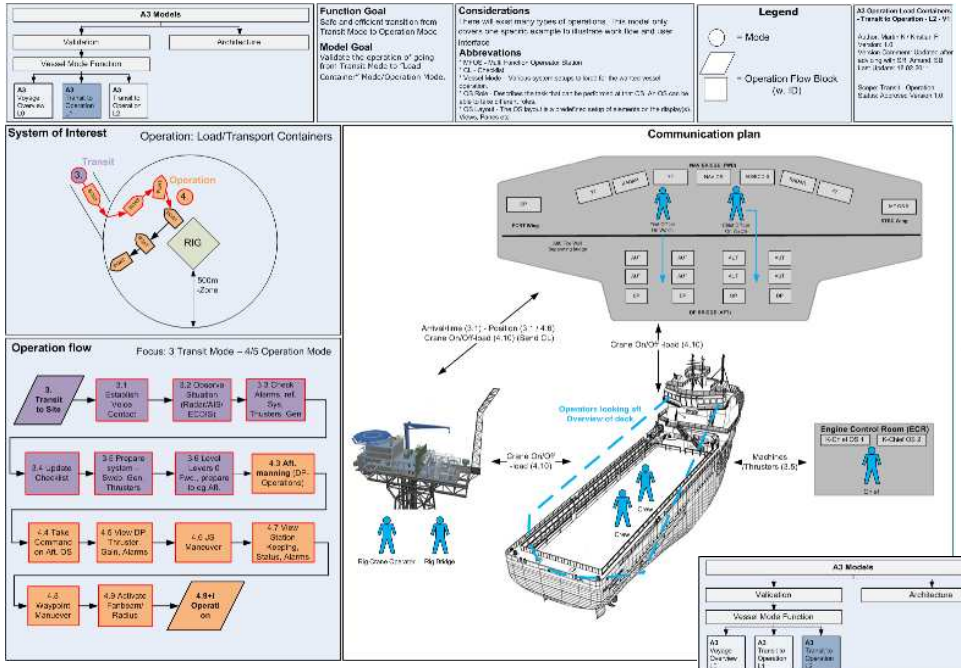
Work Flows

Systematic description of user operations.

Annotated with Where, When, Who, What

This is one class of methods, there are many more methods

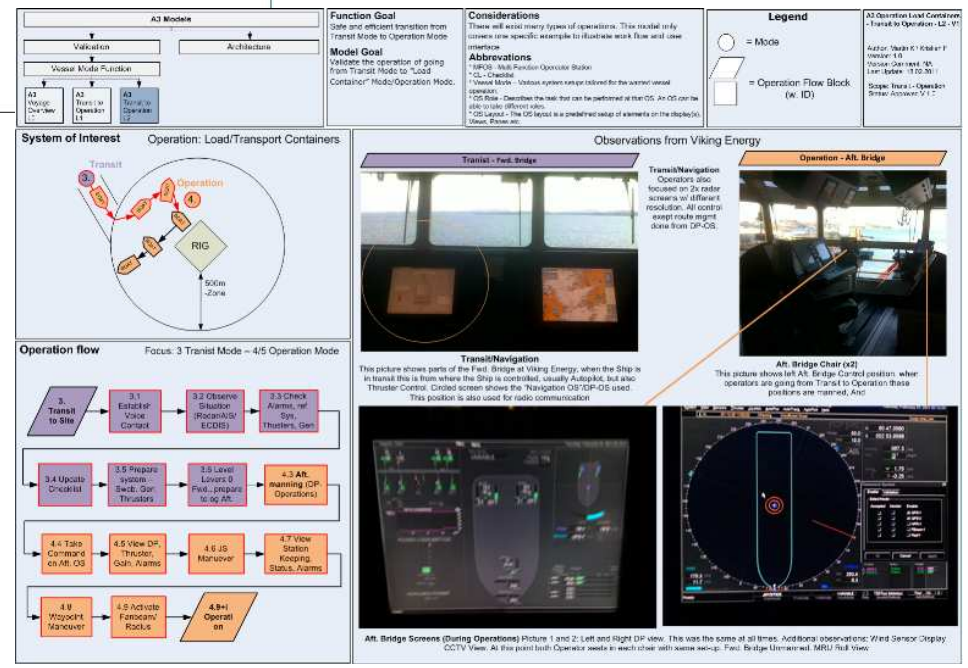
Example of Customer Knowledge



operations

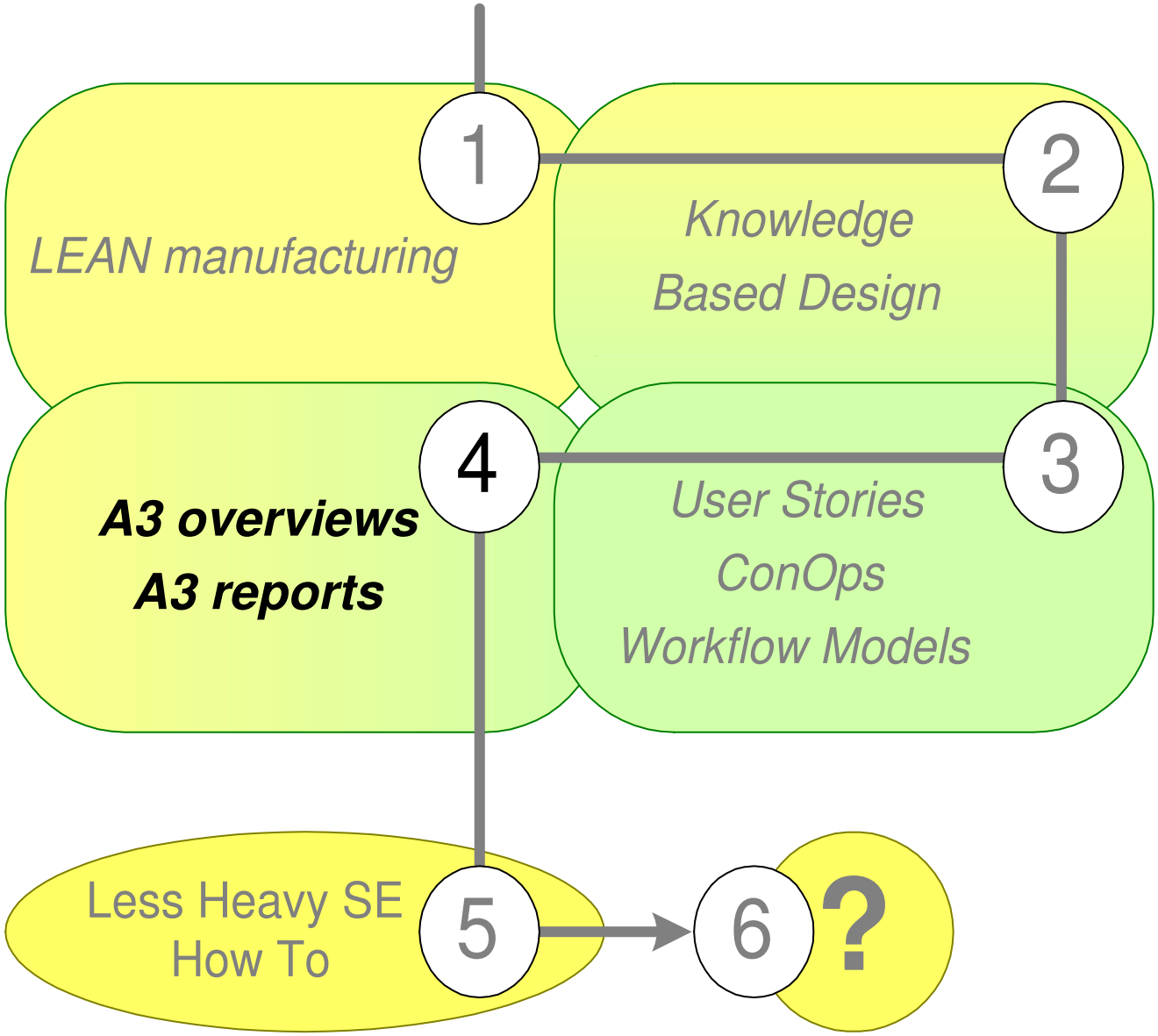
workflow

geographical

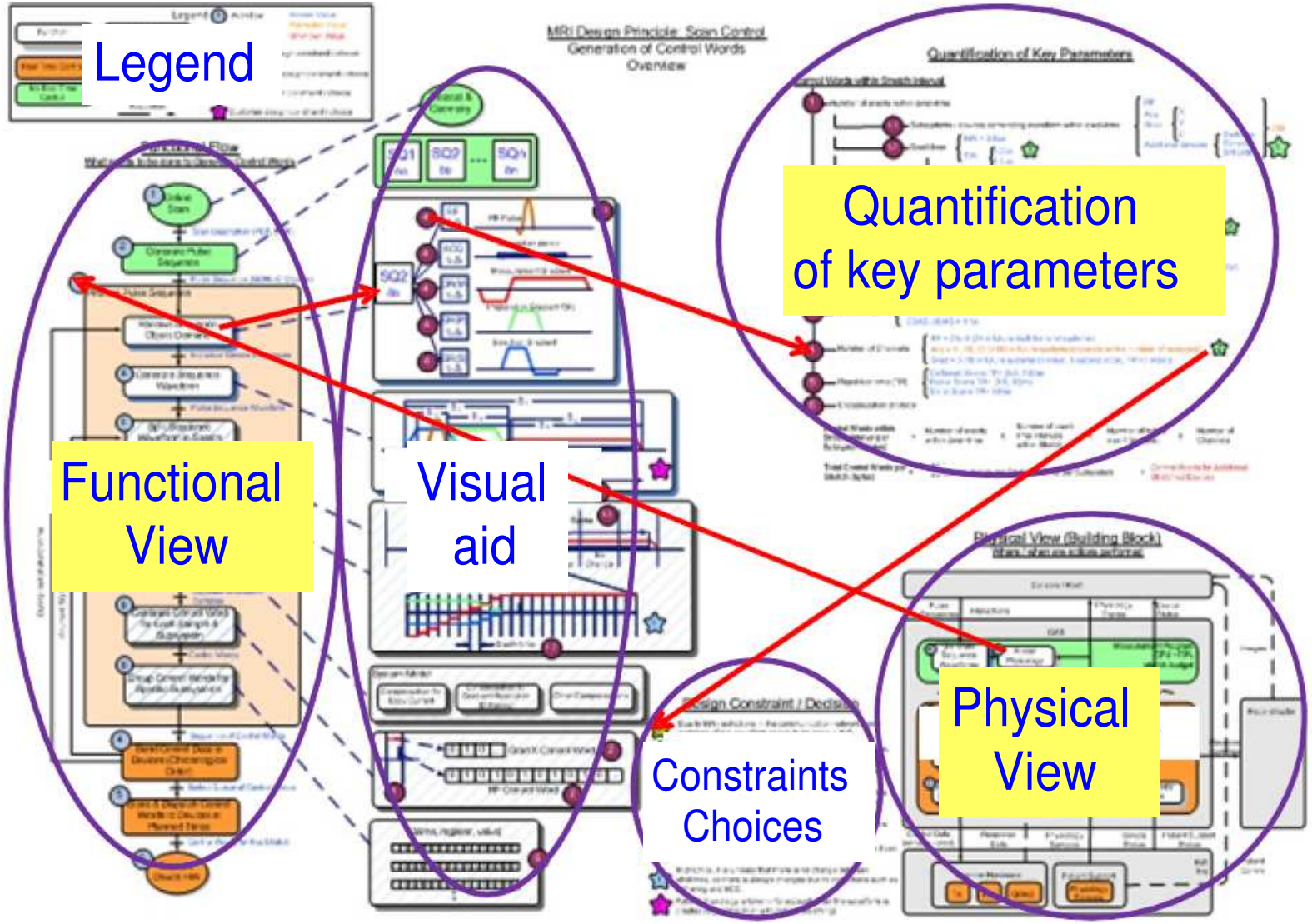


source: master project papers
Kristian Frøvold and
Martin Kruse

A3 Overviews and A3 Reports



A3 Overview Fundamentals



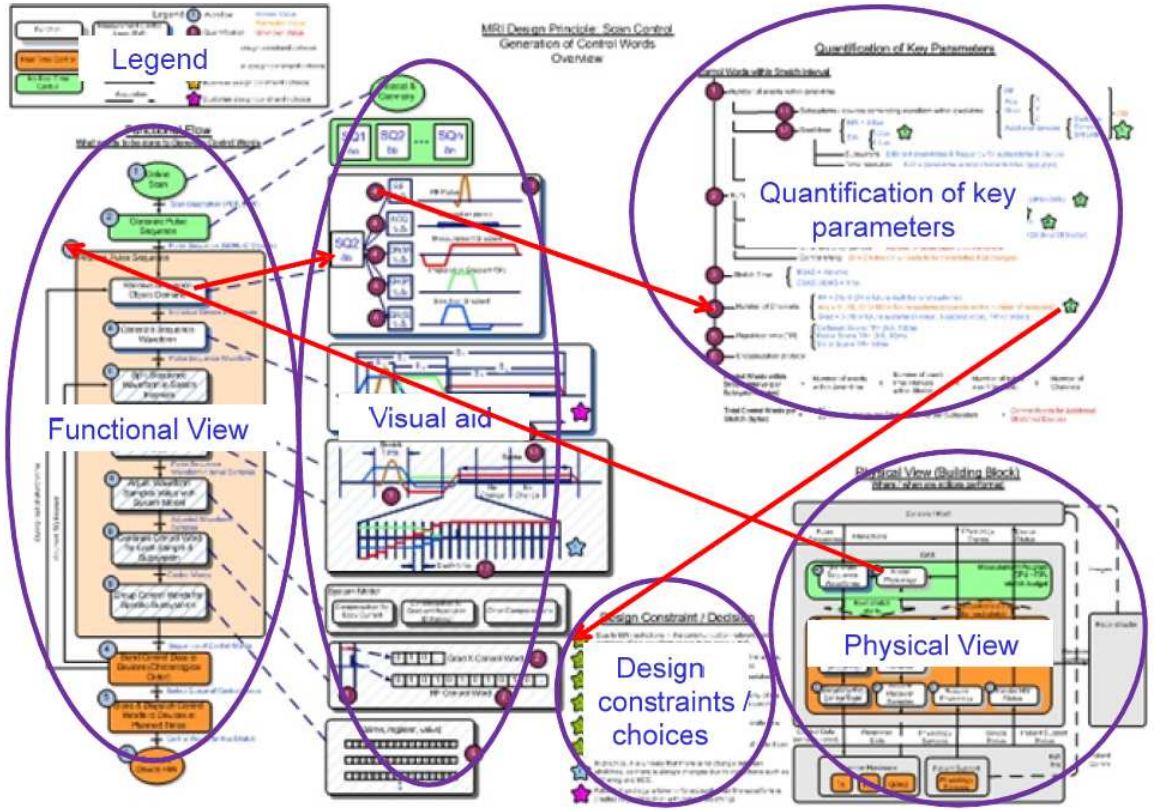
A3 Architecture Overviews Focusing architectural knowledge to support evolution of complex systems
by: Daniel Borches and Maarten Bonnema, INCOSE 2010

A3 Overview Fundamentals (2)

multiple related views

quantifications

one topic
per A3



source: PhD thesis Daniel Borches http://doc.utwente.nl/75284/

capture
"hot" topics

digestable
(size limitation)

practical

close to stakeholder experience

Evaluation of Conventional Design Spec

Results of Questionnaire System Design Specification

Statement 4: Current SDS document is useful for your work

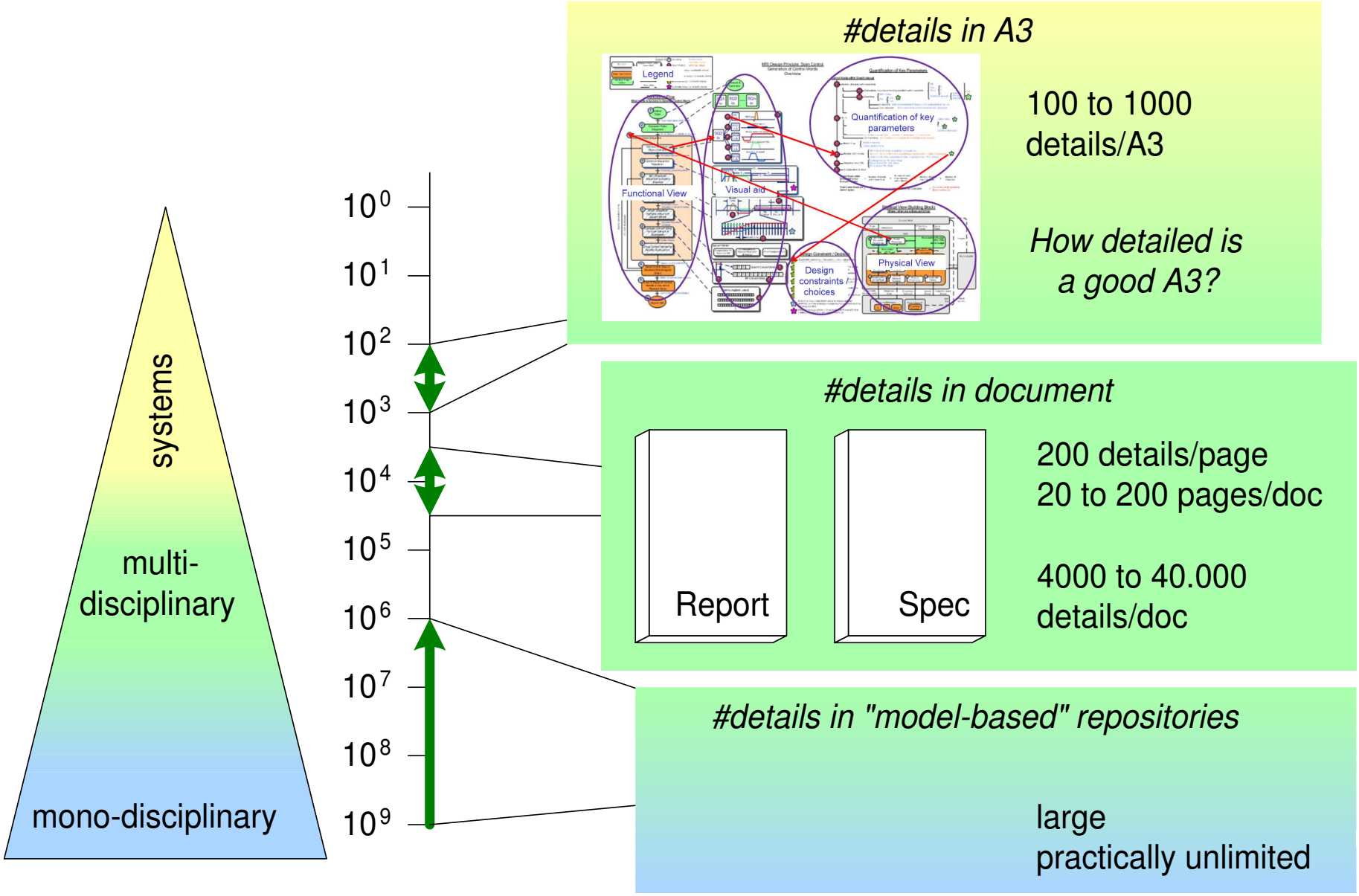
General Response		*Strongly Agree/Agree per Job Title*		*Strongly Agree/Agree per Experience*	
Strongly Agree	0%	Manager/Leader	50%	<5 Years	75%
Agree	29%	Architect	40%	5 <Years< 10	23%
Disagree	40%	Engineer	30%	10 <Years< 20	22%
Strongly Disagree	14%	Designer	0%	Since MR Proton	22%
Don't Know	17%	Domain Expert	50%	(> 20 Years)	
		Other	0%		

Statement 5: The SDS delivers what you expect from a system specification

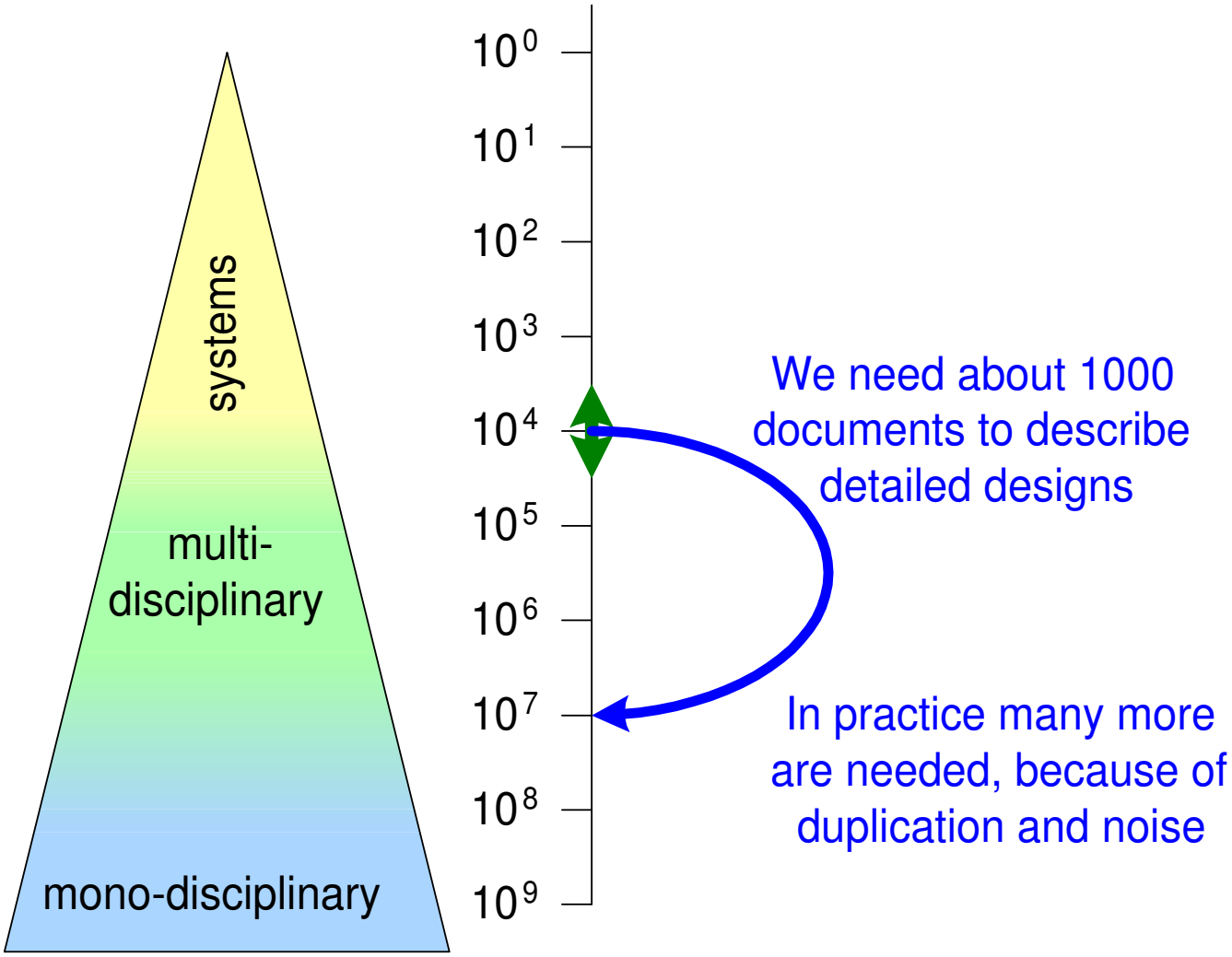
General Response		*Strongly Agree/Agree per Job Title*		*Strongly Agree/Agree per Experience*	
Strongly Agree	0%	Manager/Leader	25%	<5 Years	50%
Agree	26%	Architect	20%	5 <Years< 10	31%
Disagree	49%	Engineer	40%	10 <Years< 20	11%
Strongly Disagree	6%	Designer	0%	Since MR Proton	22%
Don't Know	20%	Domain Expert	50%	(> 20 Years)	
		Other	33%		

Source: PhD thesis Daniel Borches <http://doc.utwente.nl/75284/>

Amount of Data per Medium



Number of Conventional Documents

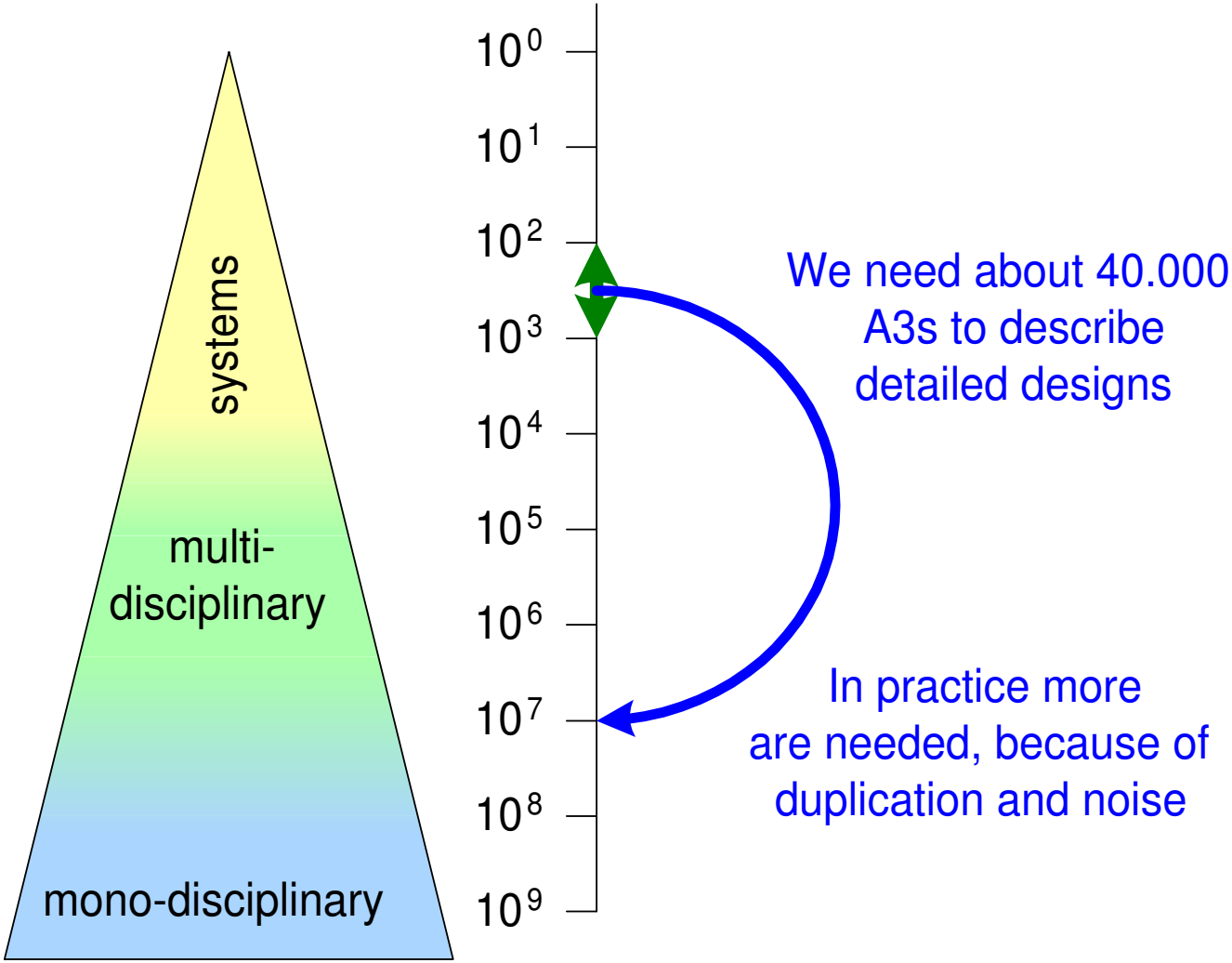


#details in document

200 details/page
20 to 200 pages/doc

4000 to 40.000 details/doc

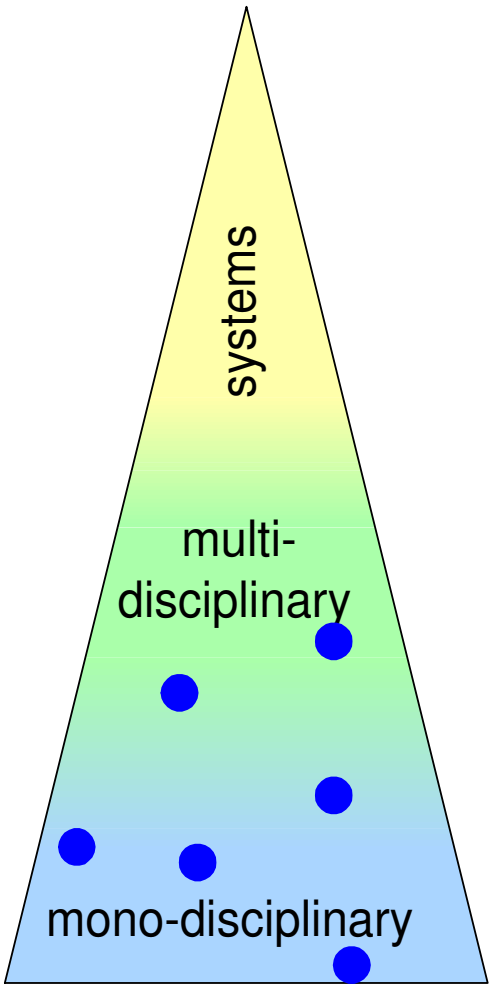
What If we Use A3s for all Detailed Designs?

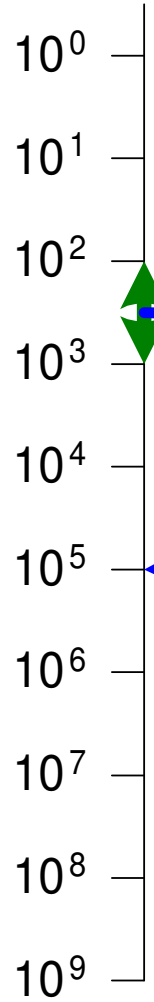


#details in A3

100 to 1000
details/A3

What If we Use A3s for System Design?





#details in A3

100 to 1000 details/A3

We need about 400 A3s to describe system design

In practice some more are needed, because of duplication and noise

A3s are useful to focus on specific issues there might be many more focused A3s

Conclusions A3s

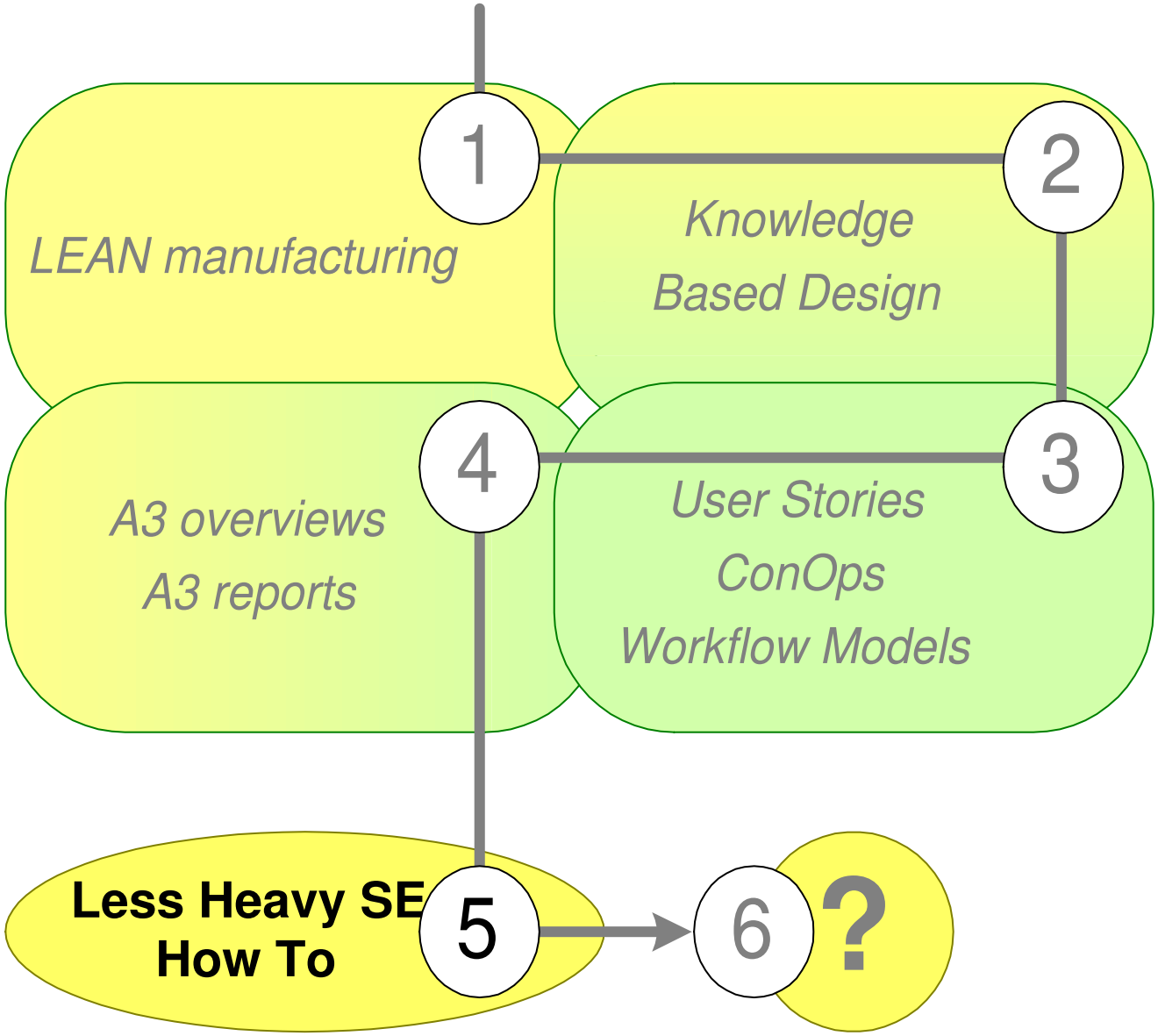
We need documents **and** A3s **and** data bases

We need to design documentation structure

We need conventions for use naming, meta information, structure, storage

A3s fit in broader context

A3s are practical and work well


1
LEAN manufacturing
2
Knowledge
Based Design
4
A3 overviews
A3 reports
3
User Stories
ConOps
Workflow Models
Less Heavy SE
How To
5
6
?

Light Weight How To

1. Reduce the rule set to the (business) essential

Understand
- your customer
- your customer's customer

etcetera

ConOps
user stories
work flows

weight(architecture) = $\sum_{\text{all rules}}$

f (level of **enforcement** ,

minimize number of mandatory rules

agile

scope (impact) ,

empower, delegate

LEAN
A3

size,

minimize implementation details
focus on essential concepts

level of **coupling** or number of dependencies)

Apply design principles on architecture
and documentation
Multi-view architecting

systems thinking
A3