

Basic Working Methods of a System Architect

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

Abstract

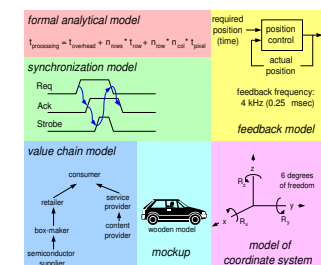
The challenge for the architect is to cover a wide range of subjects, with many unknowns and uncertainties, while decisions are required all the time.

The basic working methods, such as viewpoint hopping, modelling, handling uncertainties and WWHWWW questions are described.

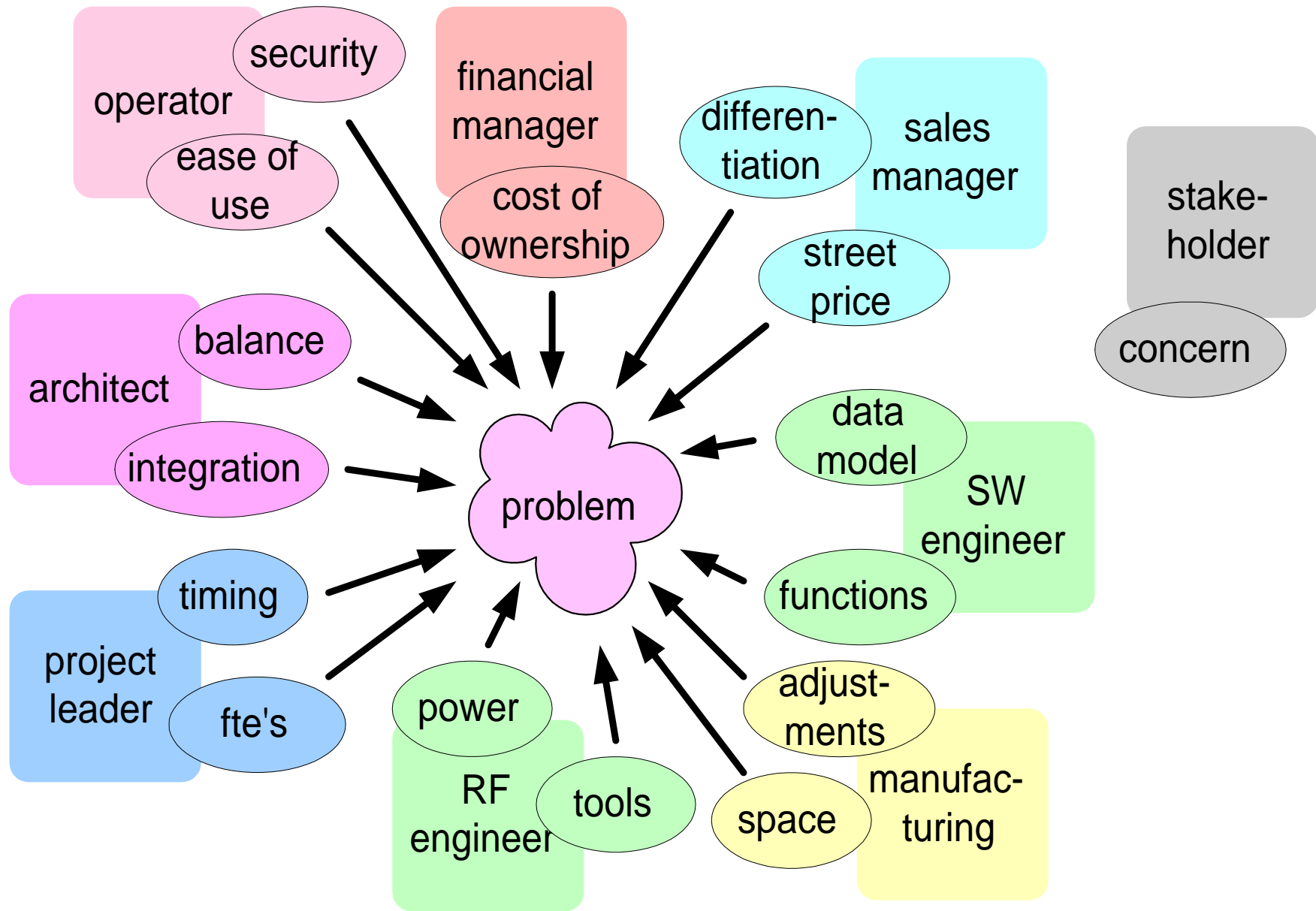
Distribution

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.

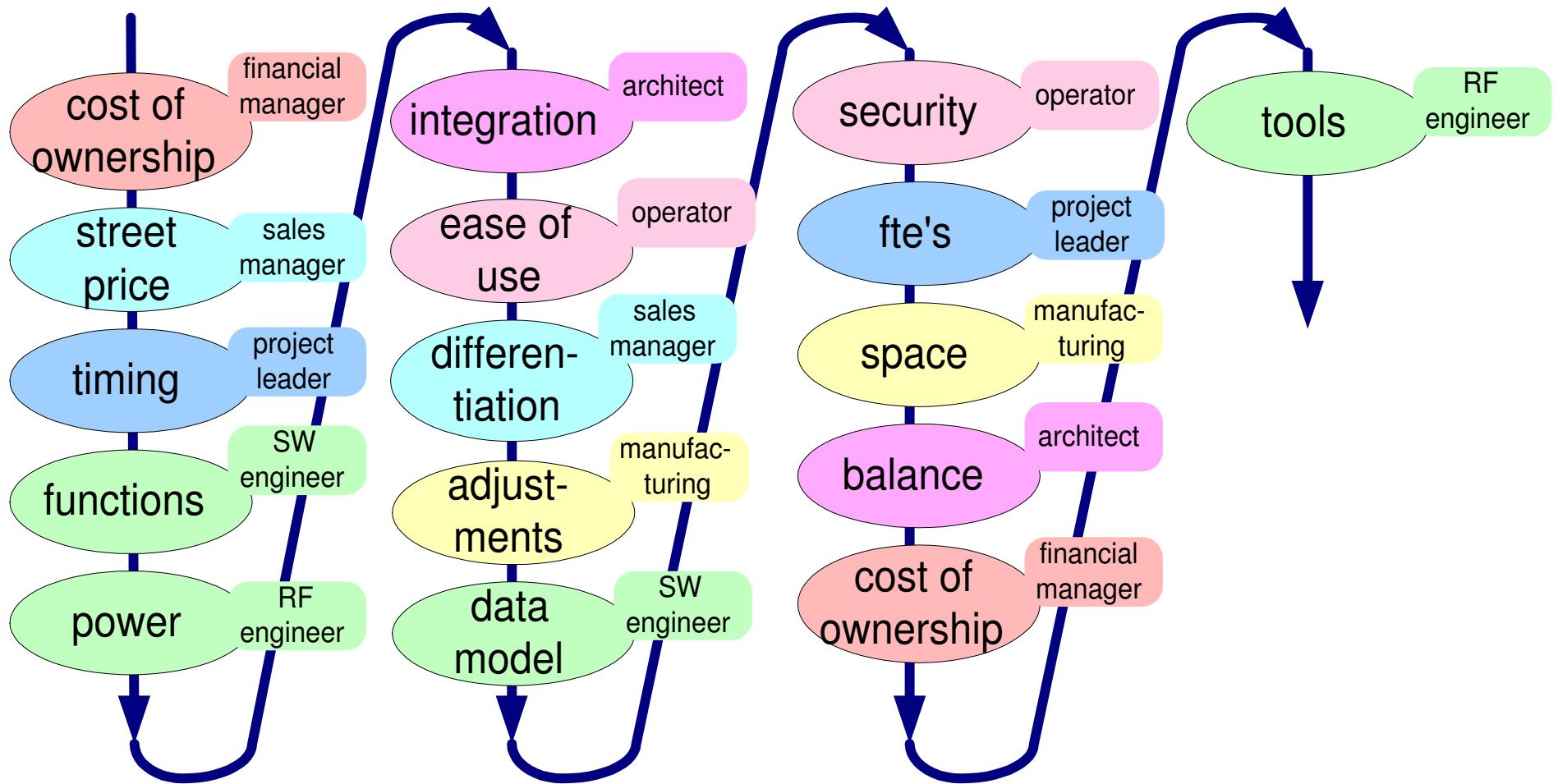
July 1, 2011
status: concept
version: 1.5



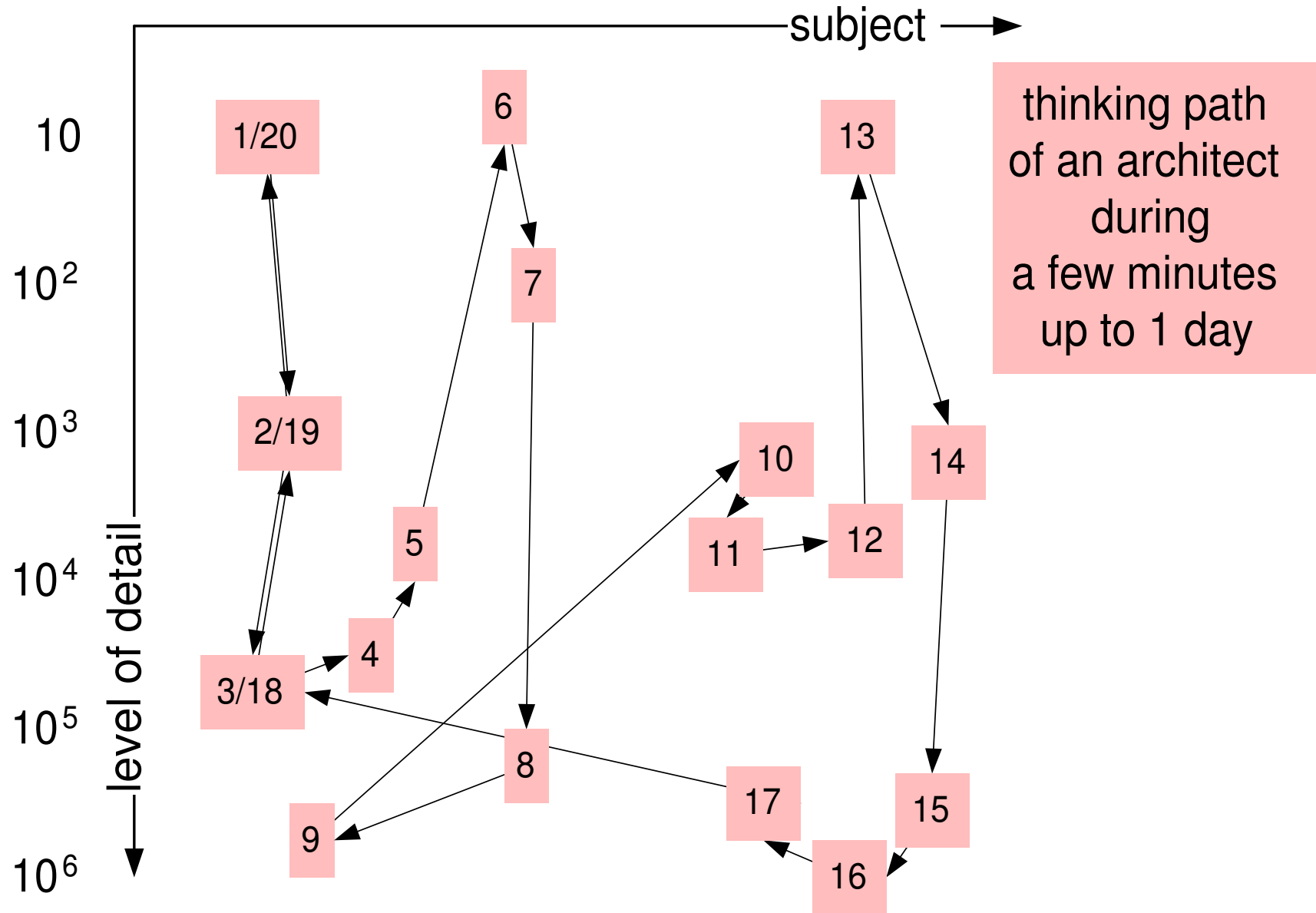
Many viewpoints



Viewpoint Hopping



The seemingly random exploration path



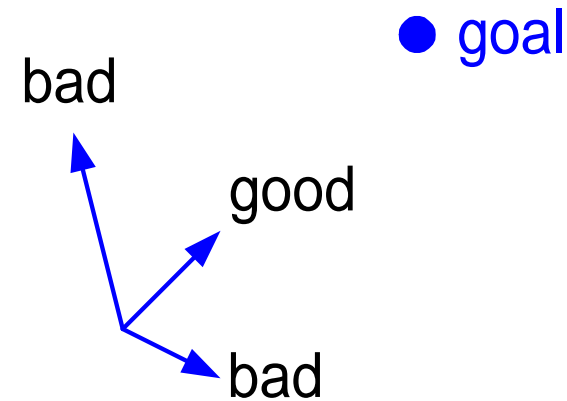
Scanning modes of the architect

open
perceptive
scanning



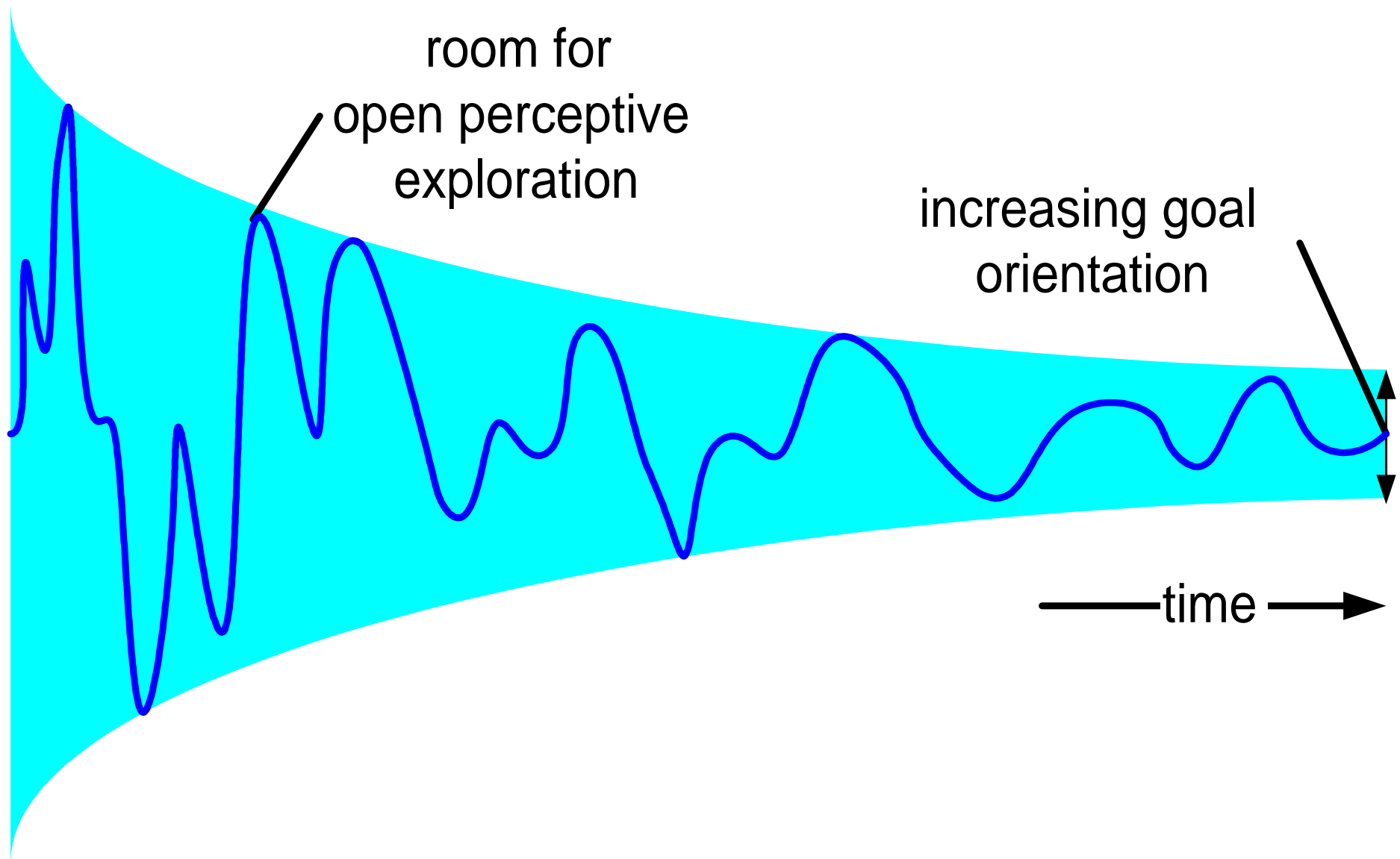
drunkard's walk
the world is full
of interesting
needs, technologies, ...

scanning
while
structuring
and judging

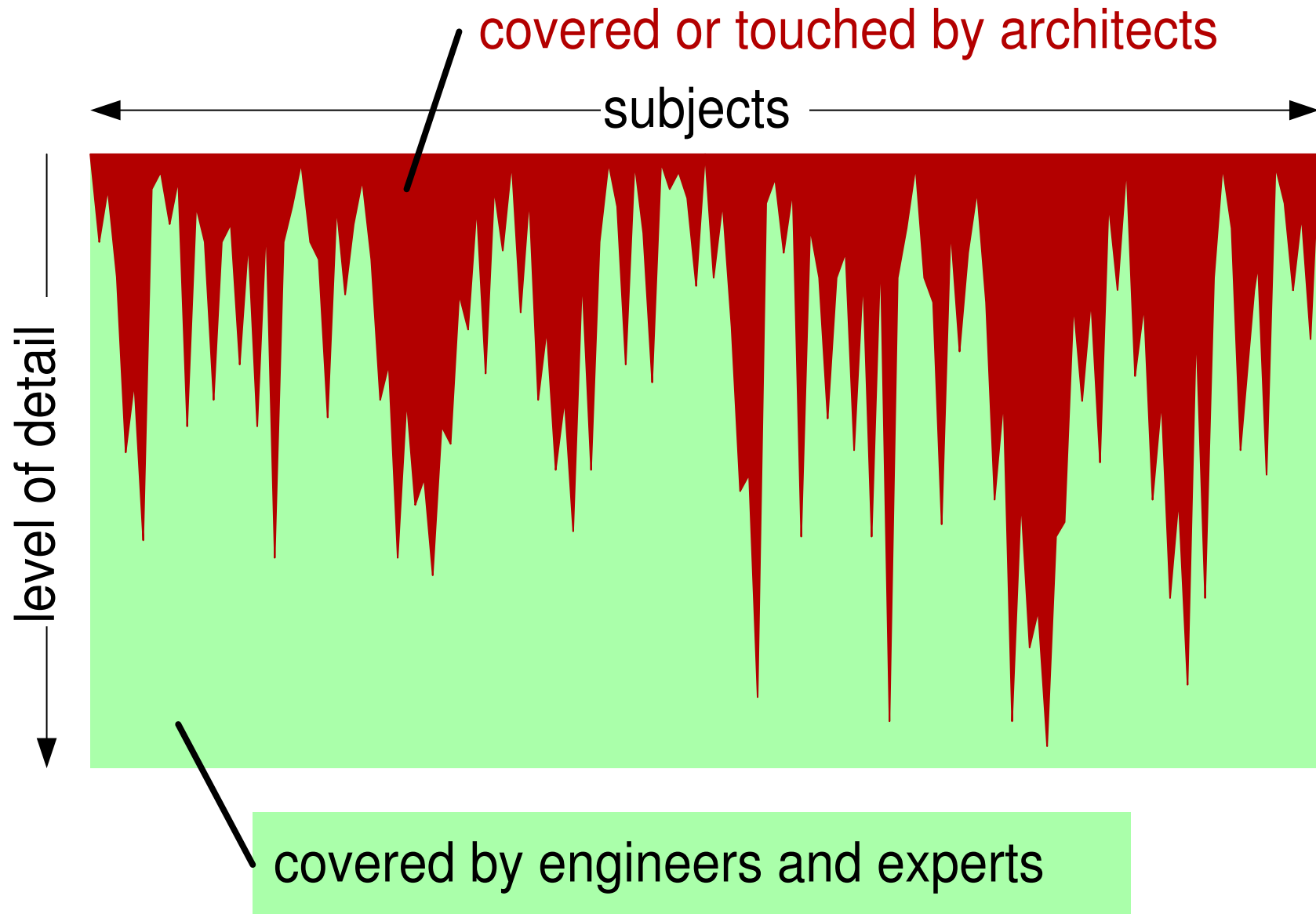


straight for the goal
ignore everything
that is not contributing
directly to the goal

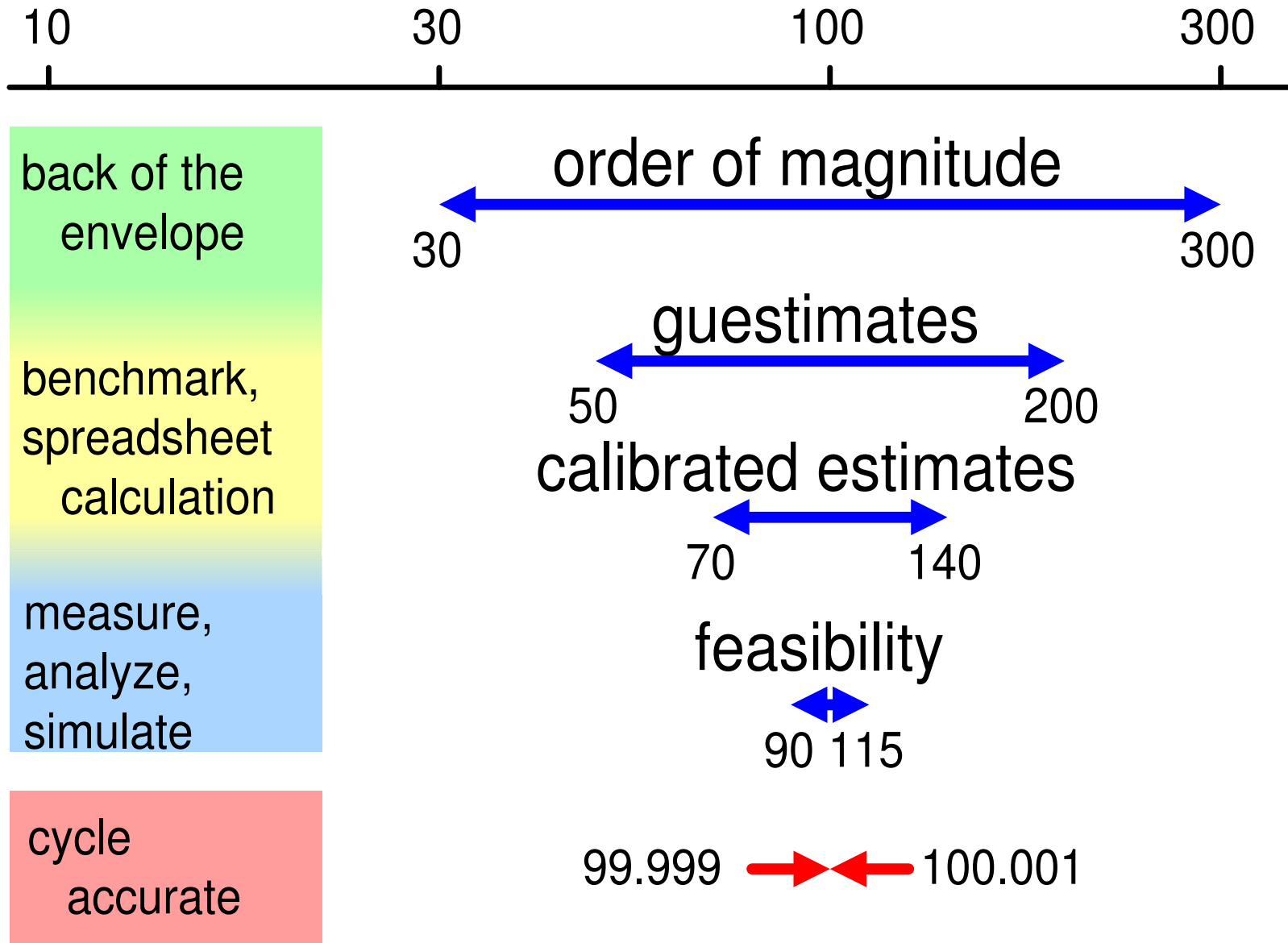
Combined open perceptive and goal oriented scanning



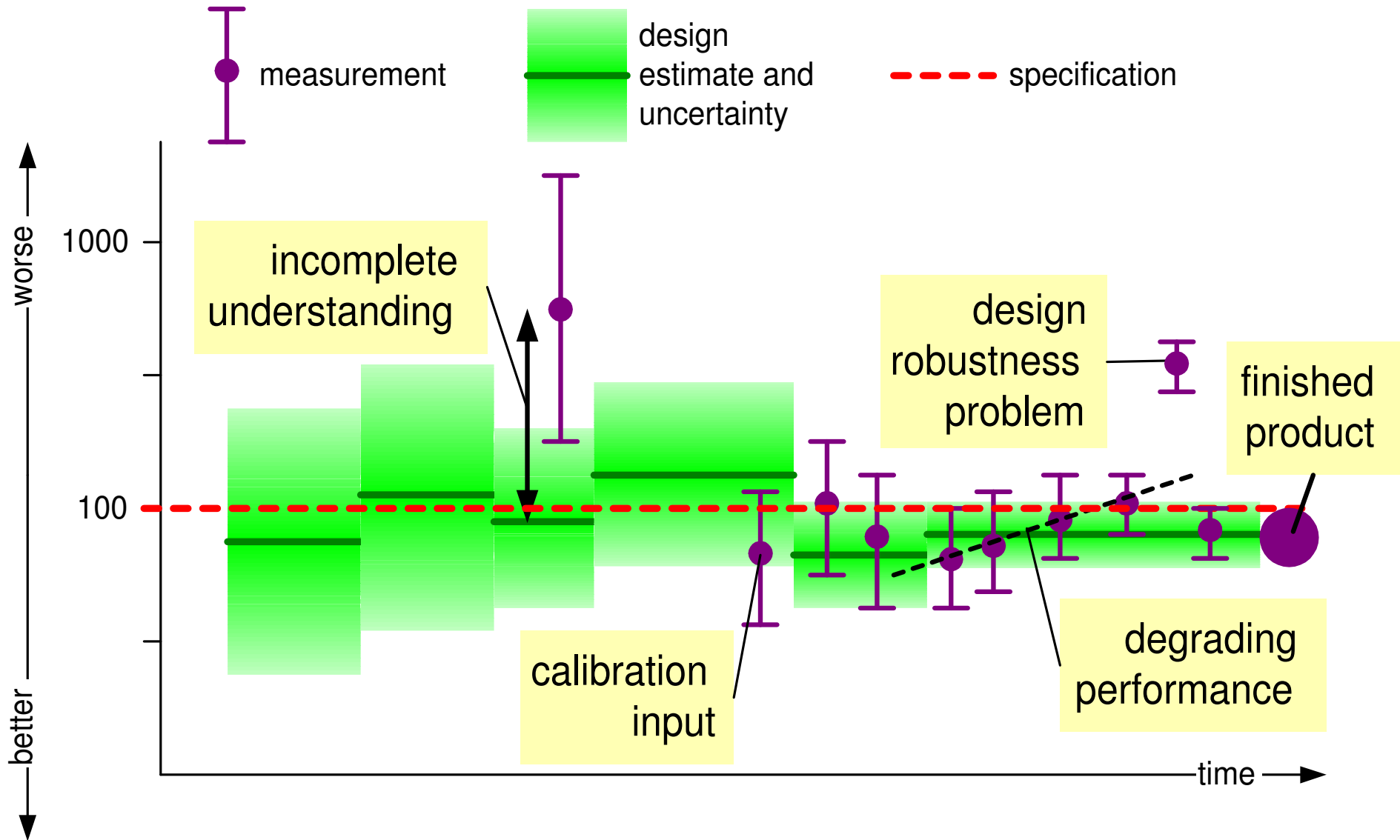
Coverage of problem and solution space



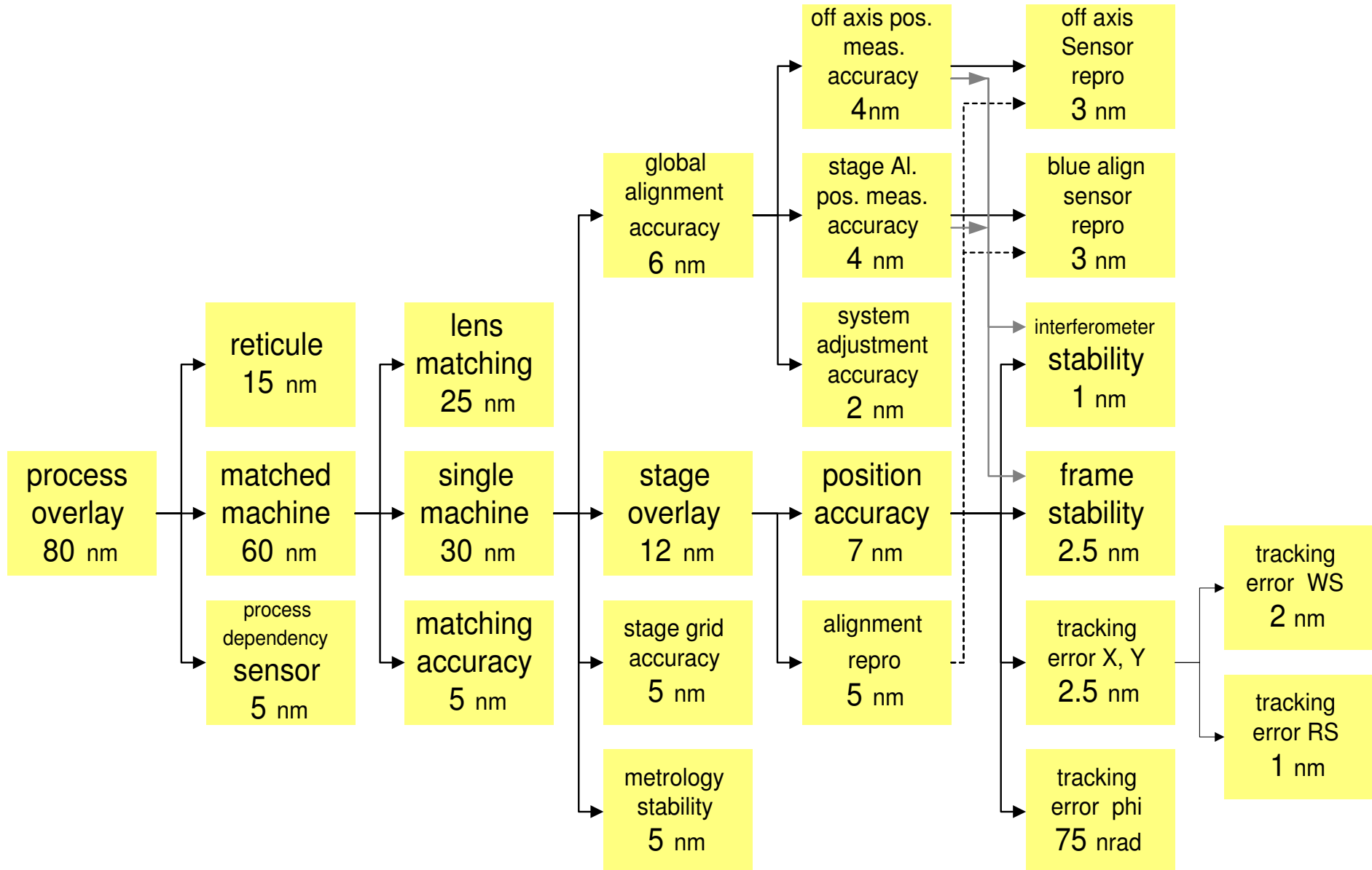
Successive quantification refinement



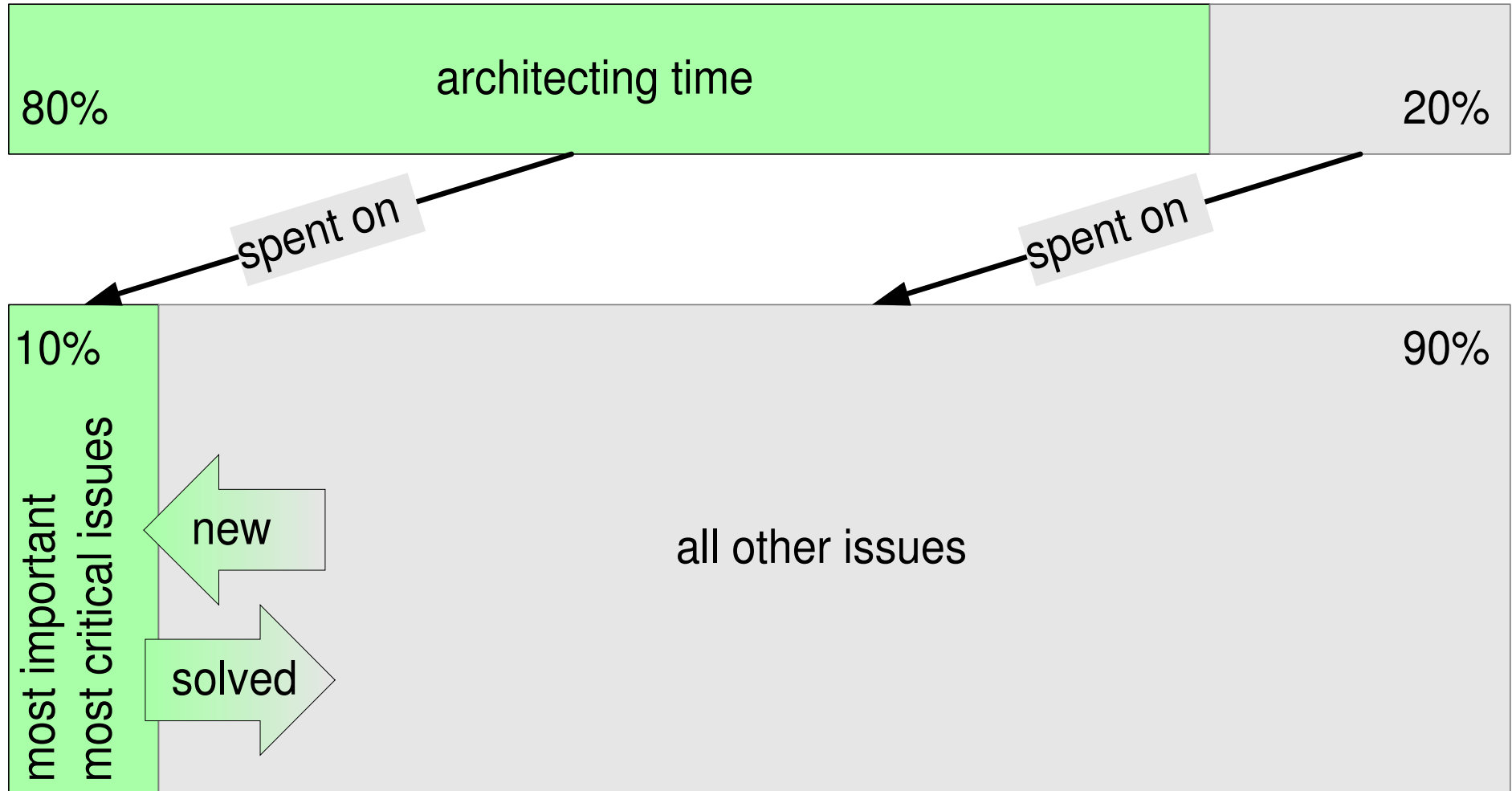
Example evolution of quantification



Quantified understanding of waferstepper overlay



Architect focus on important issues



Architect “worry” list

1. response time

from key press
until first image
on display

2. cost price

resource budgets

3. layering to separate

separation of concerns
self sustained
life-cycle separation
robust: paranoia validations

4. reliability of storage

5. database redesign

6. integration schedule

7. movement artefact

8. standby power

9. weak signal handling

10. location-based twiddle

A **model** is
a **simplified** representation of
part of the **real world** used for:

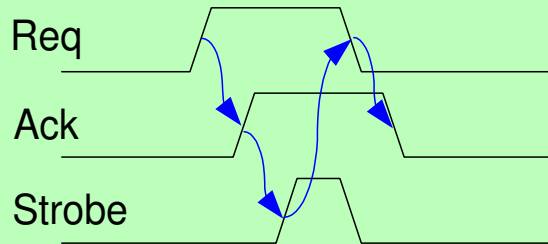
communication, documentation
analysis, simulation,
decision making, verification

Some examples of models

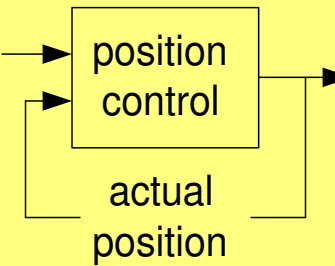
formal analytical model

$$t_{\text{processing}} = t_{\text{overhead}} + n_{\text{rows}} * t_{\text{row}} + n_{\text{row}} * n_{\text{col}} * t_{\text{pixel}}$$

synchronization model



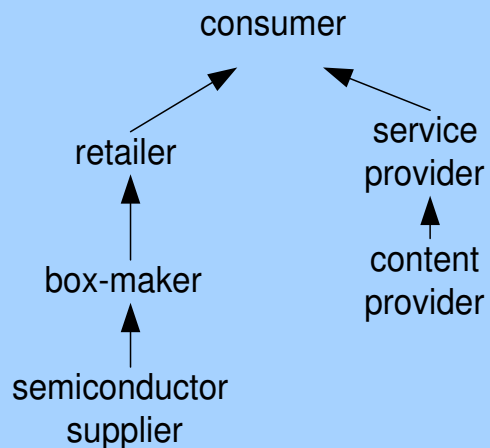
required
position
(time)



feedback frequency:
4 kHz (0.25 msec)

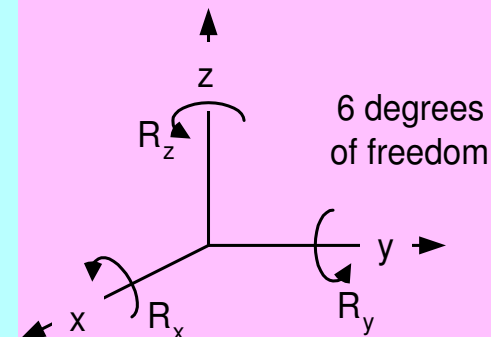
feedback model

value chain model



wooden model

mockup



model of coordinate system

Types of models

mathematical

linguistic

visual

formal

informal

quantitative

qualitative

detailed

global

concrete

abstract

accurate

approximate

executable

read only

← rational —

— intuitive →

Why

Who

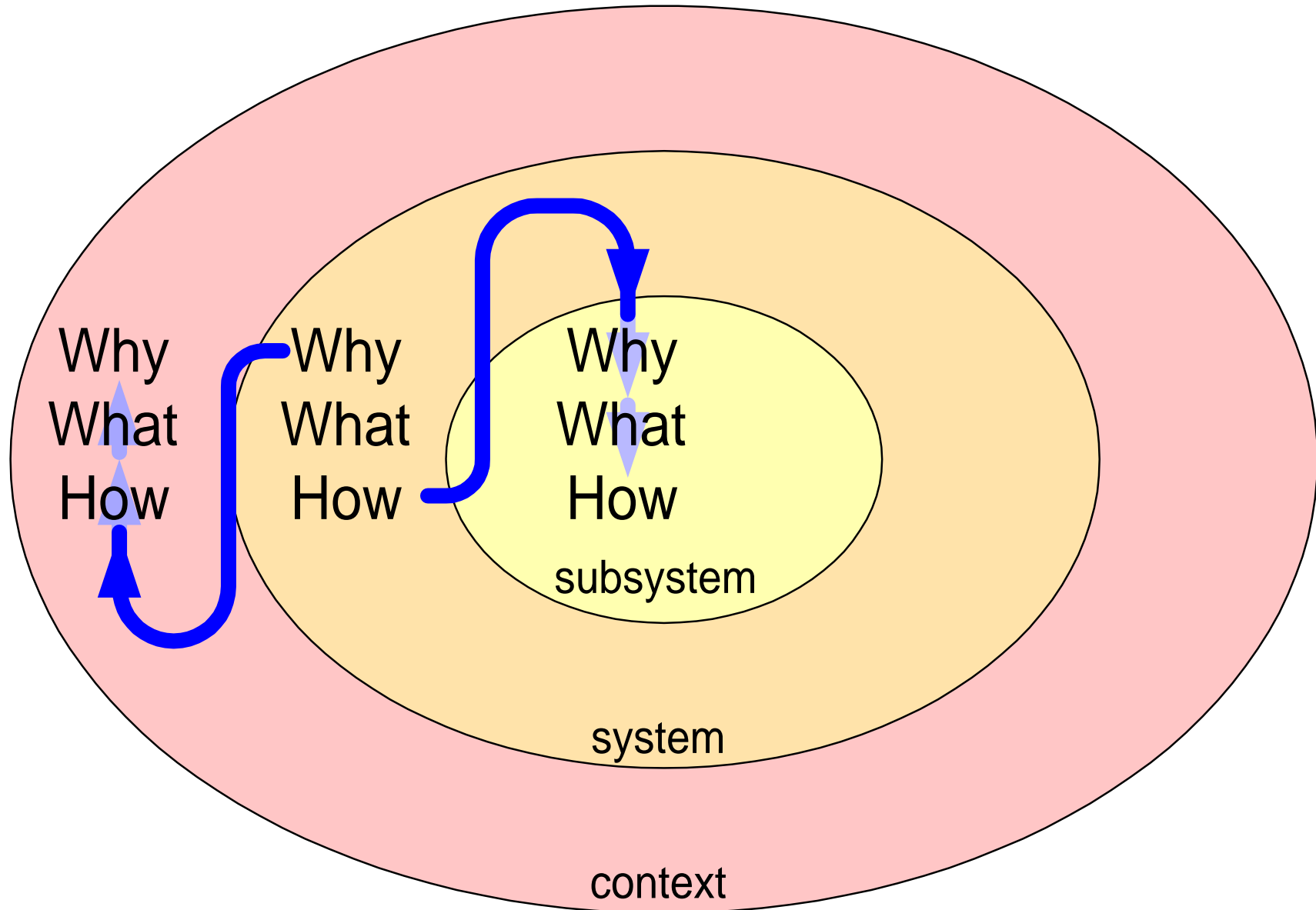
What

When

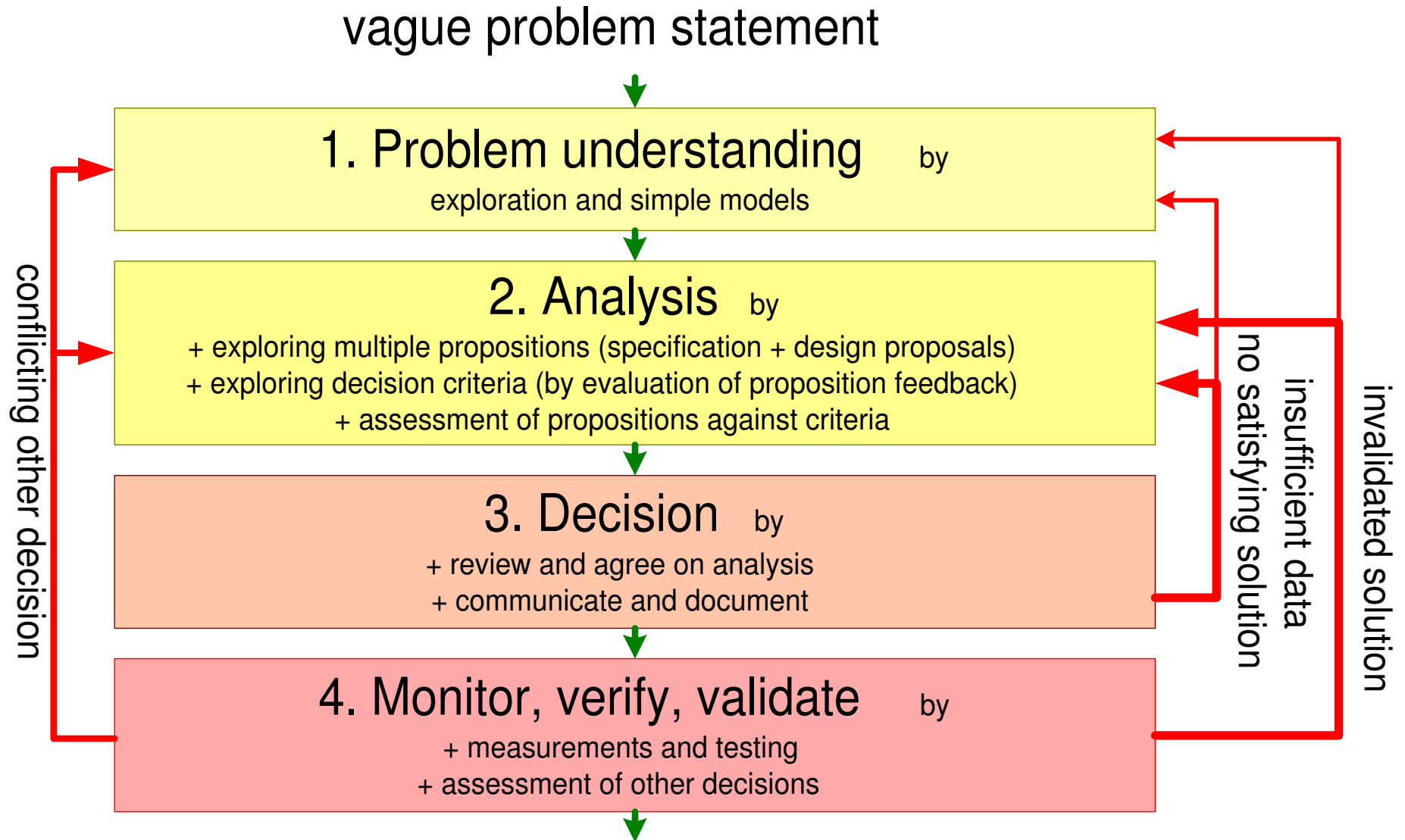
How

Where

Why broadens scope, How opens details



Flow from problem to solution



Multiple propositions

throughput	20 p/m	high-performance sensor	350 ns
cost	5 k\$	high-speed moves	9 m/s
safety		additional pipelining	

low cost and performance 1

throughput	20 p/m	high-performance sensor	300 ns
cost	5 k\$	high-speed moves	10 m/s
safety			

low cost and performance 2

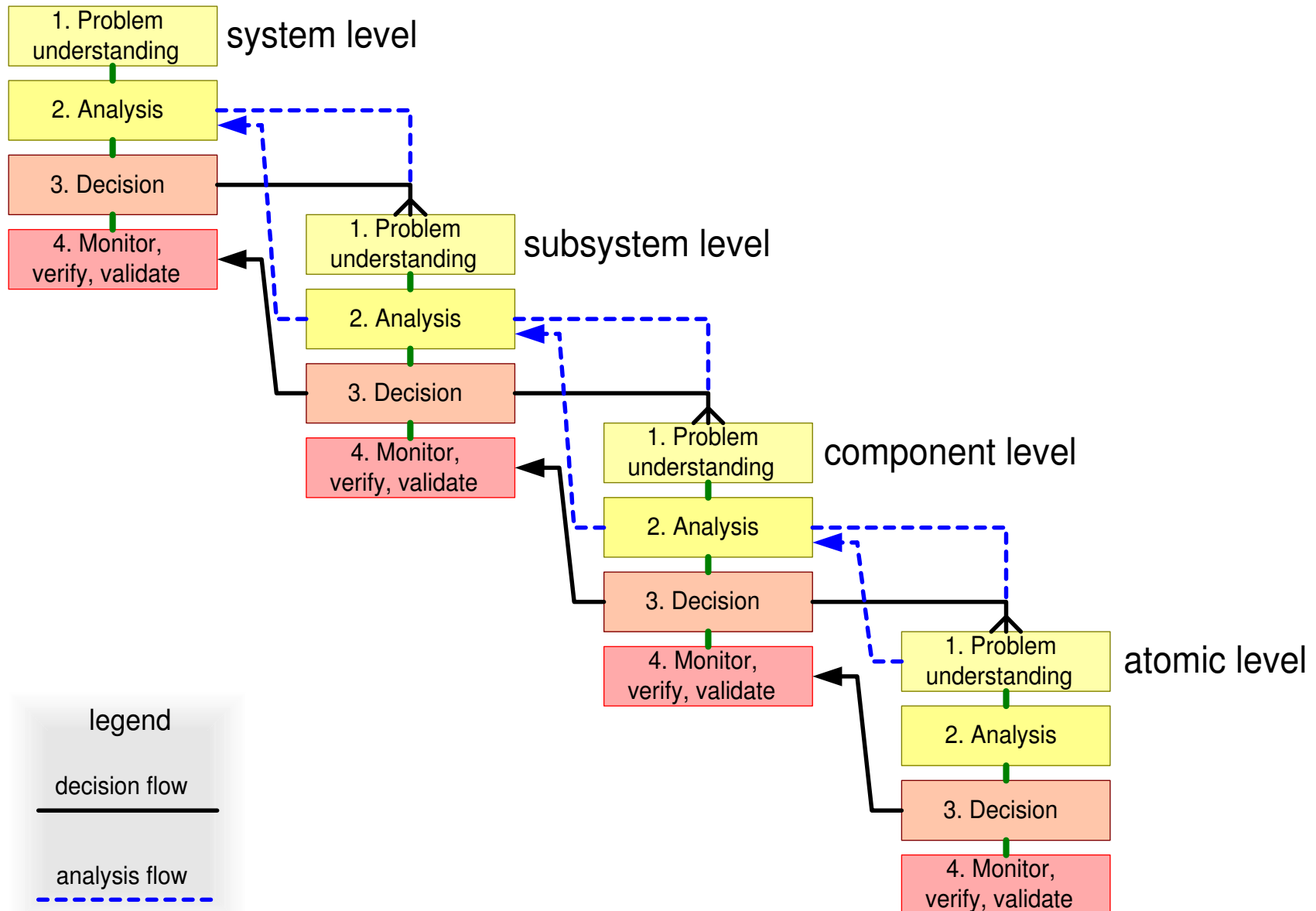
throughput	25 p/m	highperformance sensor	200 ns
cost	7 k\$	high-speed moves	12 m/s
safety		additional collision detector	

high cost and performance

Assessment of propositions

<i>criteria</i>	<i>weight</i>	<i>low cost and performance 1</i>	<i>low cost and performance 2</i>	<i>high cost and performance</i>
throughput	5	2	2	3
cost	5	3	3	2
safety	5	5	5	5
future proof	2	2	3	3
effort	4	5	4	4
dev. time	5	5	4	4
risk	4	4	3	3
maintenance	3	2	3	3

Recursive and concurrent application of flow



Exploration by rapid iteration

