

Modeling and Analysis: Reasoning

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

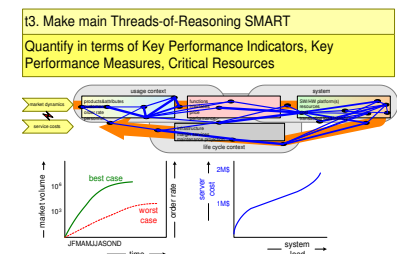
Abstract

Models are made to facilitate decision making. These decisions range from business decisions, such as Service Level Agreements, to requirements, and to detailed design decisions. The space of decisions is huge and heterogeneous. The proposed modeling approach is to use multiple small and simple models. In this paper we discuss how to reason by means of multiple models.

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.

February 10, 2011
status: preliminary
draft
version: 0.6



content

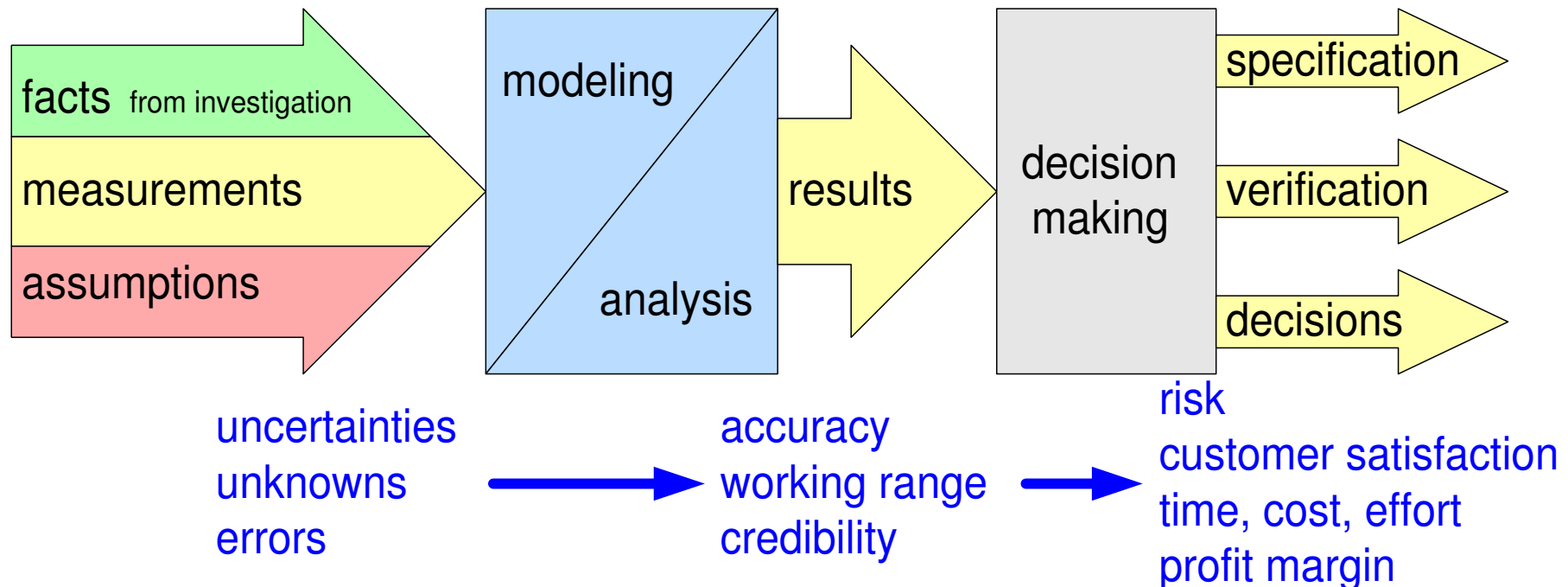
From chaos to order: inputs, assumptions, models and decisions

Reasoning approach: stepwise top-down and bottom-up

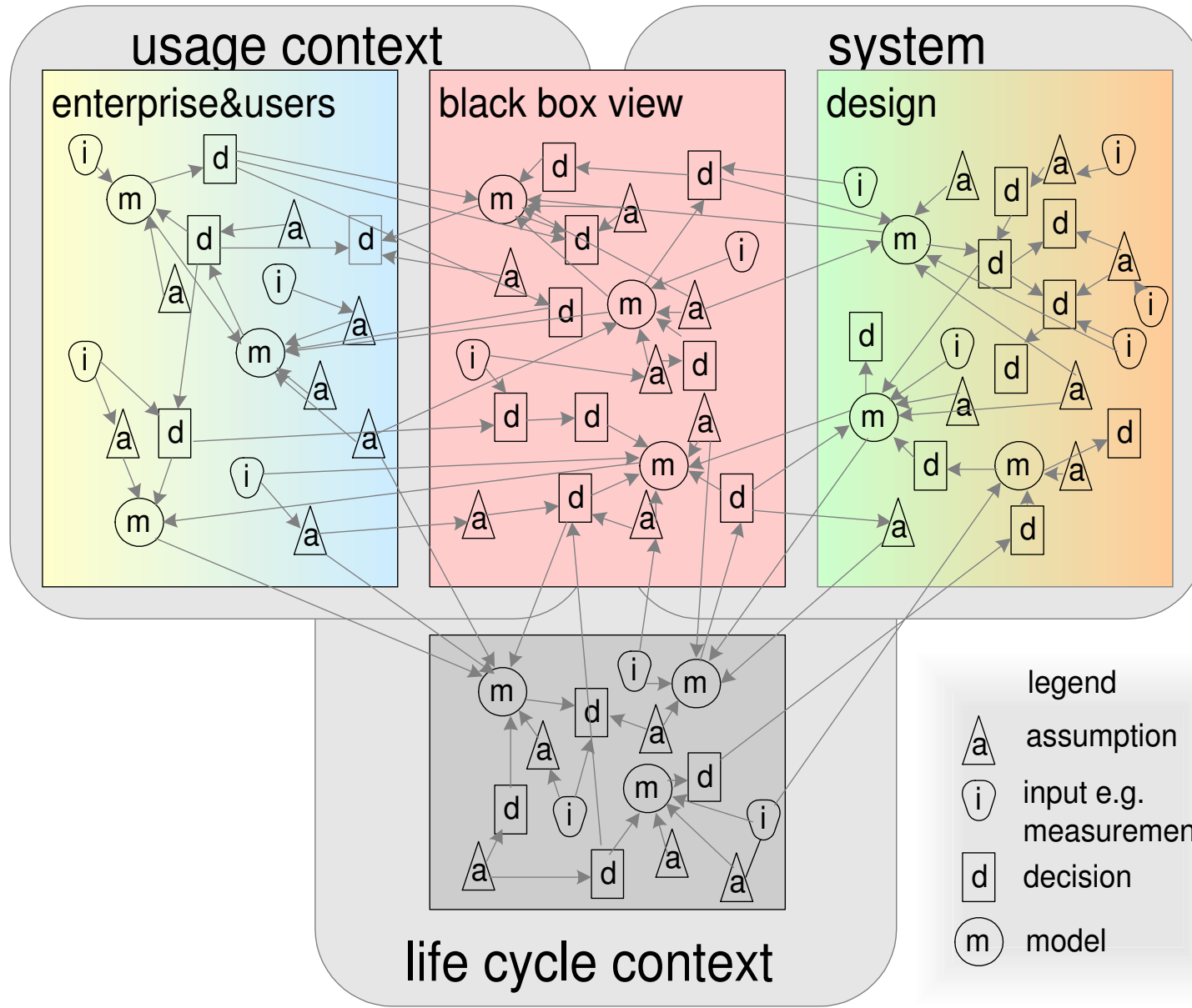
Life cycles of models in relation to project life cycle

Purpose of Modeling

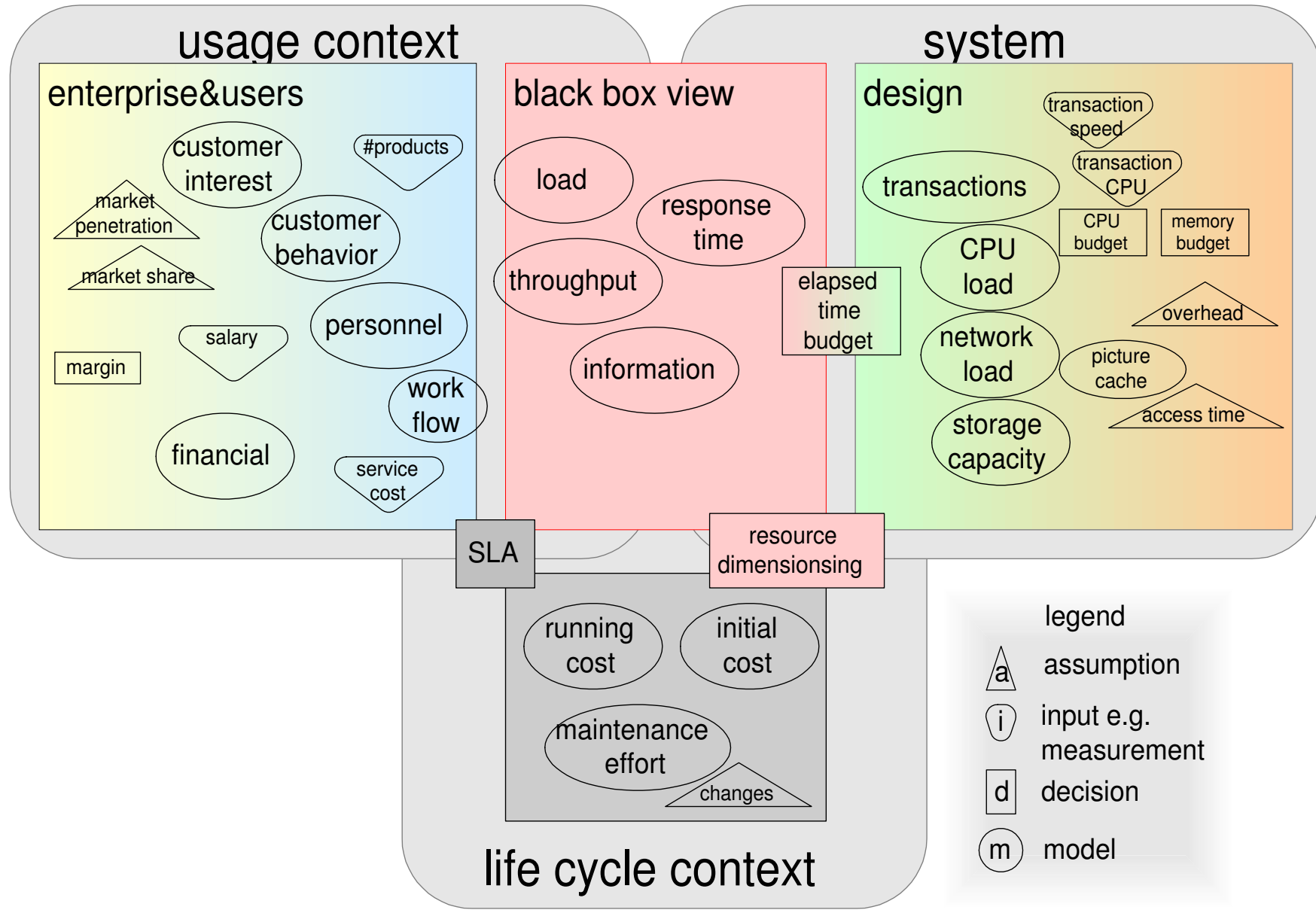
*How to use multiple models to facilitate decisions?
How to get from many fragments to integral insight?
How many models do we need?
At what quality and complexity levels ?*



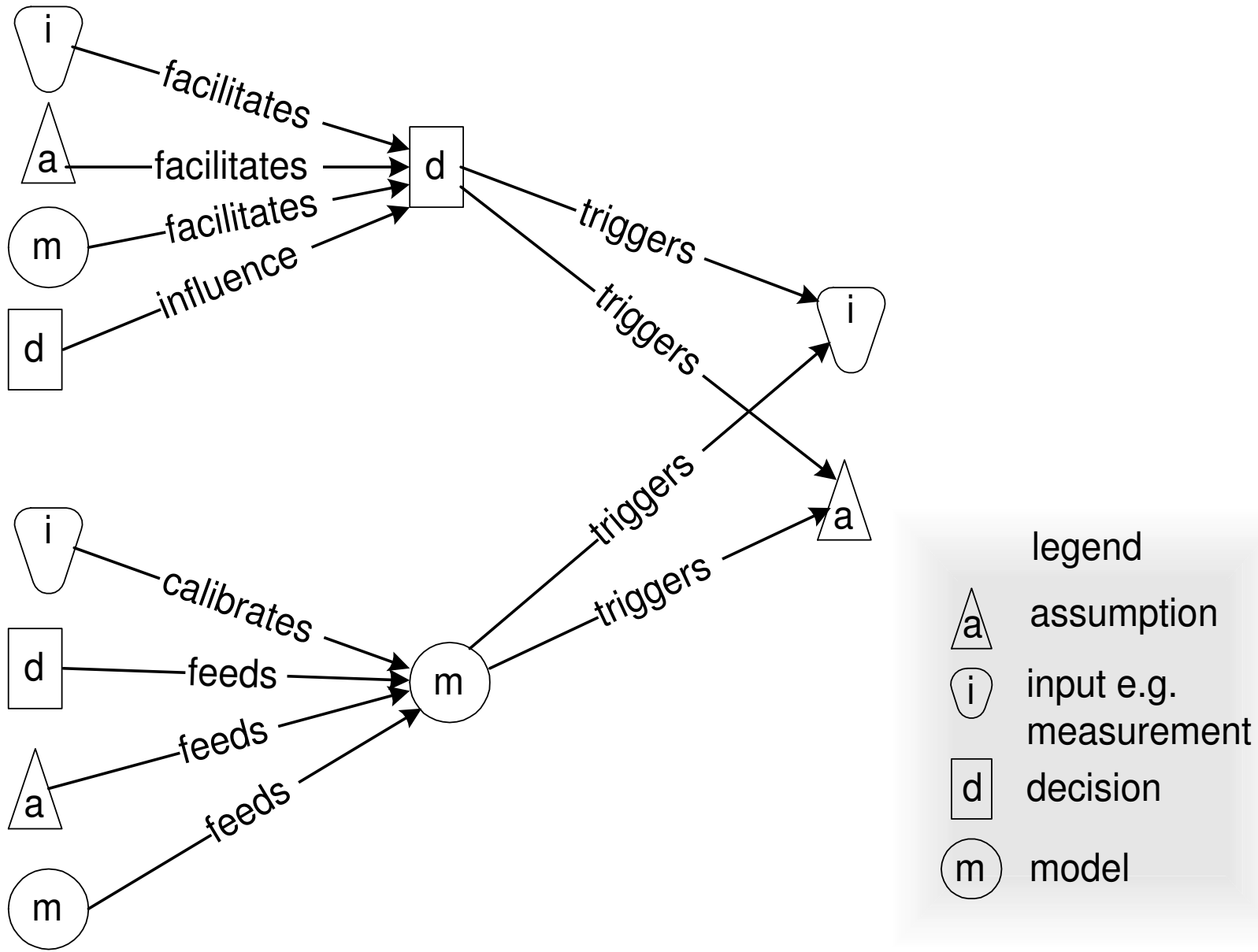
Graph of Decisions and Models



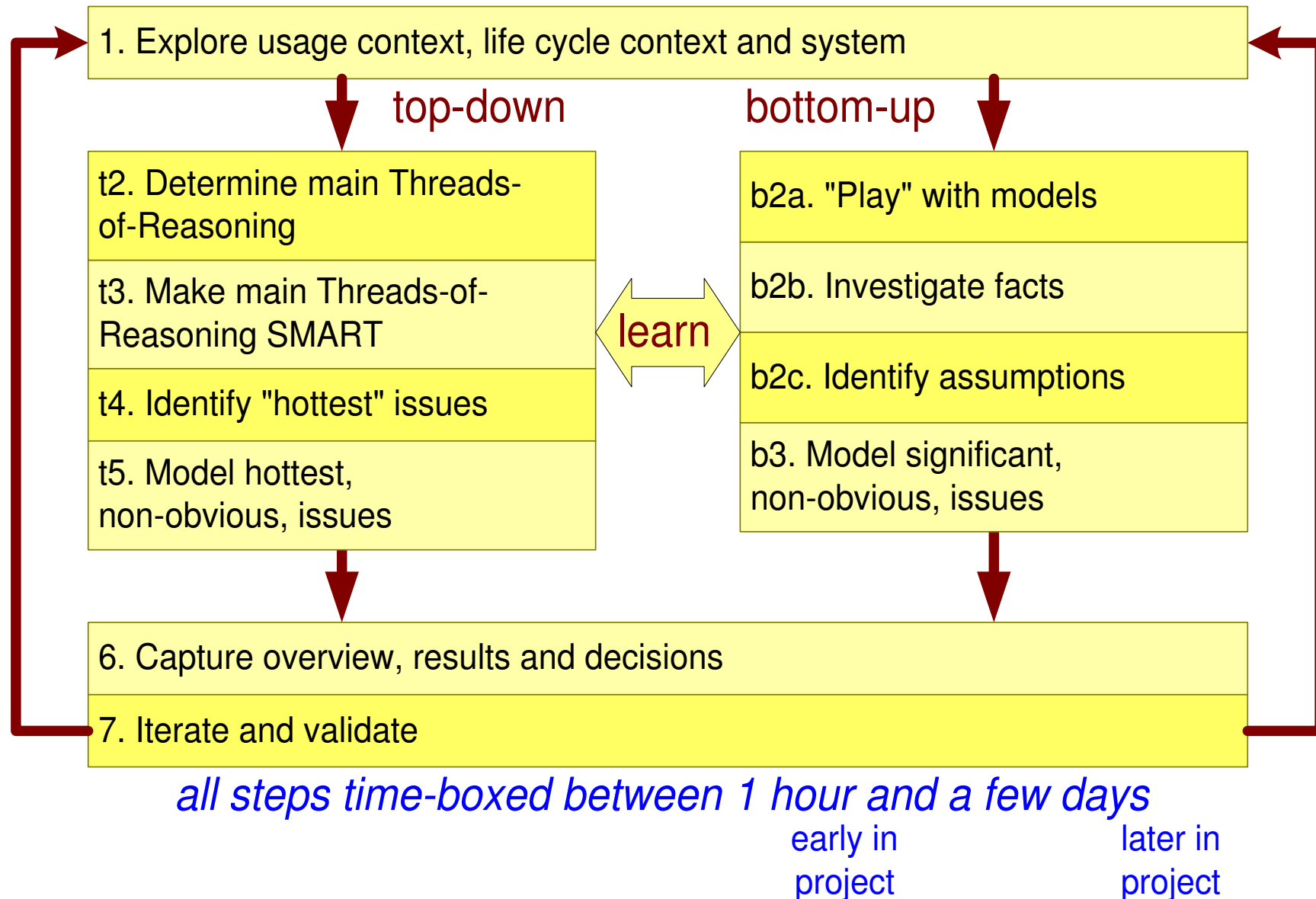
Example Graph for Web Shop



Relations: Decisions, Models, Inputs and Assumptions



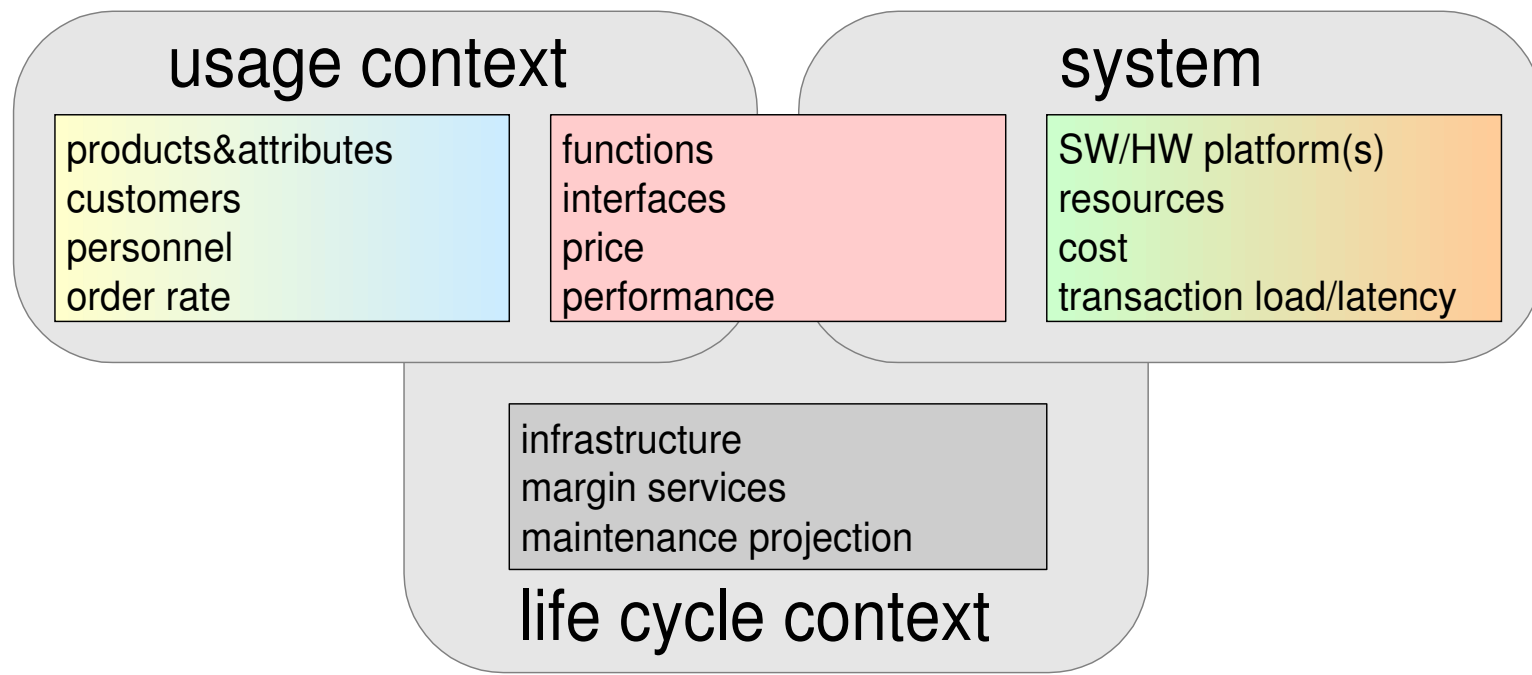
Reasoning Approach



1. Explore

1. Explore usage context, life cycle context and system

Populate with "known" facts, numbers, issues from preceeding projects, available work products and stakeholders

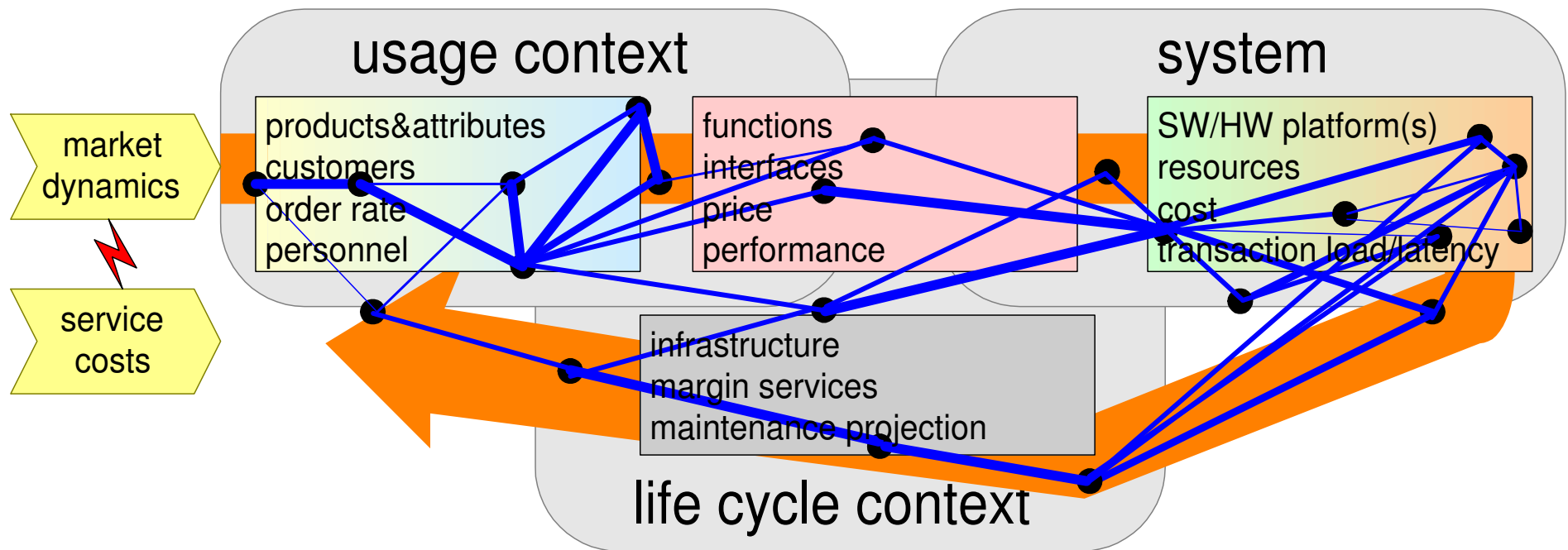


t2. Thread-of-Reasoning

t2. Determine main Threads-of-Reasoning

Architecting and System Design

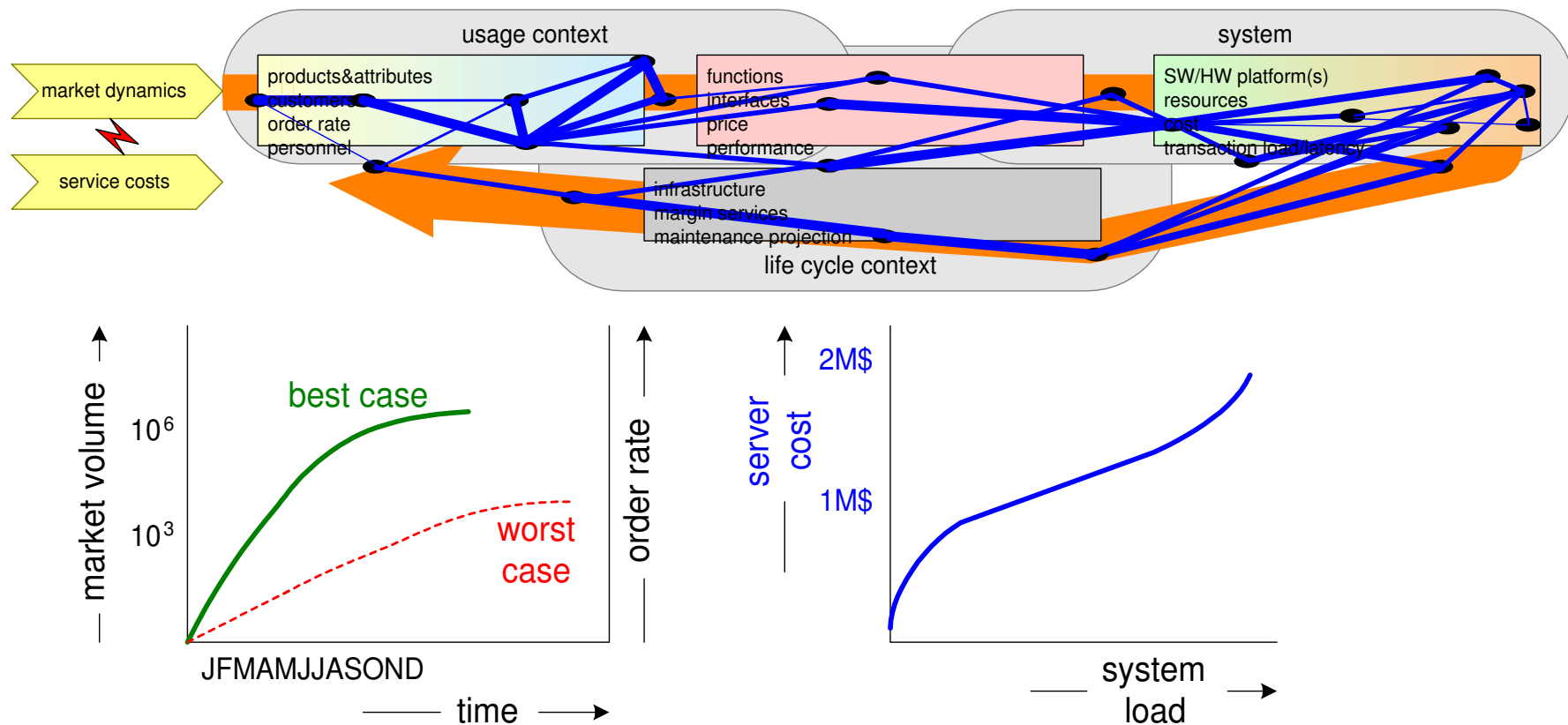
e.g. <http://www.gaudisite.nl/ModuleTORSides.pdf>



t3. SMART'en Thread-of-Reasoning

t3. Make main Threads-of-Reasoning SMART

Quantify in terms of Key Performance Indicators, Key Performance Measures, Critical Resources



Intermezzo: the acronym SMART

- Specific

quantified

- Measurable

verifiable

acronym consensus

- Assignable (Achievable, Attainable, Action oriented, Acceptable, Agreed-upon, Accountable)

- Realistic (Relevant, Result-Oriented)

- Time-related (Timely, Time-bound, Tangible, Traceable)

variation of meaning

t4: Identify Hottest

t4. Identify "hottest" issues

assess explored landscape:

highest (perceived) risk
most important/valuable
most discussed
historic evidence
urgency

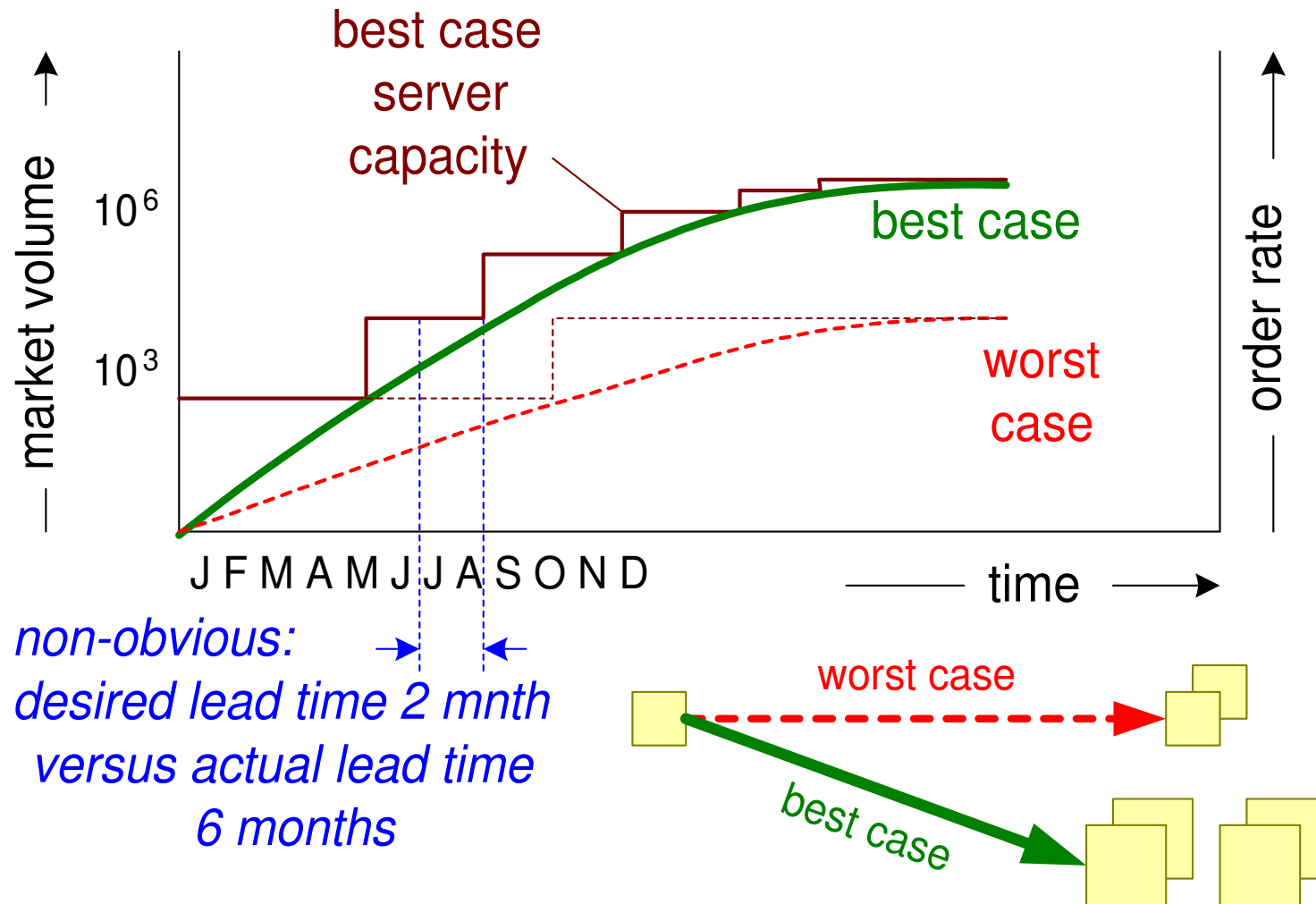
1..5 scale,
1 = low risk
5 = high risk
et cetera

rank issues according to aggregated assessment

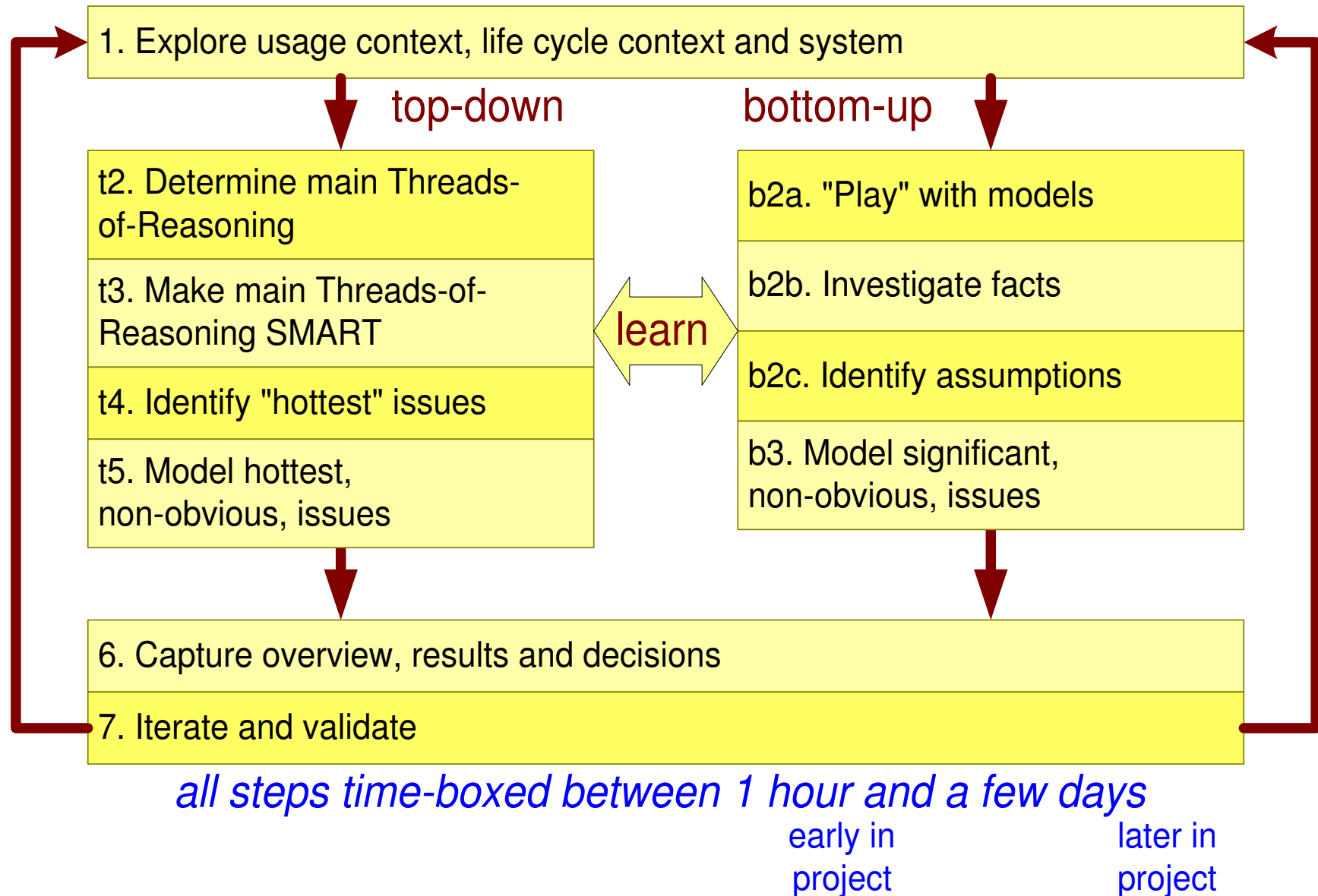
	risk	value	discussion	history	urgency	ranking
server cost	2	3	2	1	3	
order rate	4	5	5	3	5	1
transactions	3	3	3	4	2	3
response time	3	5	1	4	2	2
availability	2	5	1	3	3	4
network bandwidth	1	1	3	1	3	
storage capacity	1	1	1	2	3	

t5: Model Hottest Issues

t5. Model hottest, non-obvious, issues



From *top-down* to *bottom-up*



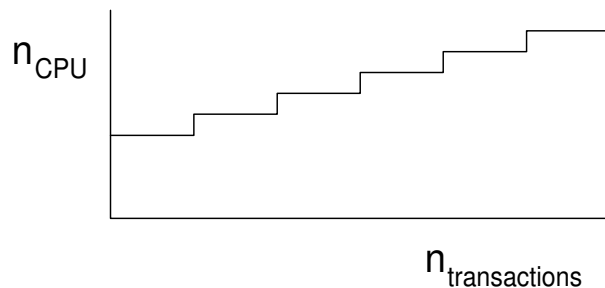
b2abc: Bottom-up

b2a. "Play" with models	b2b. Investigate facts	b2c. Identify assumptions
vary inputs vary model structure to understand <i>model applicability</i> , <i>design quality</i> and <i>specification feasibility</i>	market research measurements preceeding systems micro benchmarks literature, supplier info	What is the basis for model structure, design decision, specification, quantification et cetera? <i>Most assumptions are implicit and hidden!</i>

$$n_{\text{CPU}} = t_{\text{required total}} / t_{1 \text{ CPU}}$$

$$t_{\text{required total}} = n_{\text{transactions}} * t_{1 \text{ transaction}} + t_{\text{other}}$$

$$t_{1 \text{ transaction}} = 1 \text{ ms (on 1 CPU)}$$



http://www.tpc.org/tpcc/results/tpcc_perf_results.asp

IBM System p5 595
 TPC-C Throughput 4,033,378
 Total # of Processors: 32
 Total # of Cores: 64
 $1/t_{1 \text{ transaction}} = 4 * 10^6 / 60 / 64$
min to sec / # cores
 $t_{1 \text{ transaction}} \approx 1 \text{ ms}$

server load dominated by transactions
 transaction load scales linear
 TPC-C is representative
 what is the effect of other TPC-C workload?

Bottom-up, more detailed steps

Make a list of technologies, components and resources to be used

transactions, data base engine, memory, disk

Make a list of important qualities

performance, reliability, security, maintainability

Make a characterization matrix of technologies, components and resources versus qualities

1..5 scale,
1 = low risk
5 = high risk
et cetera

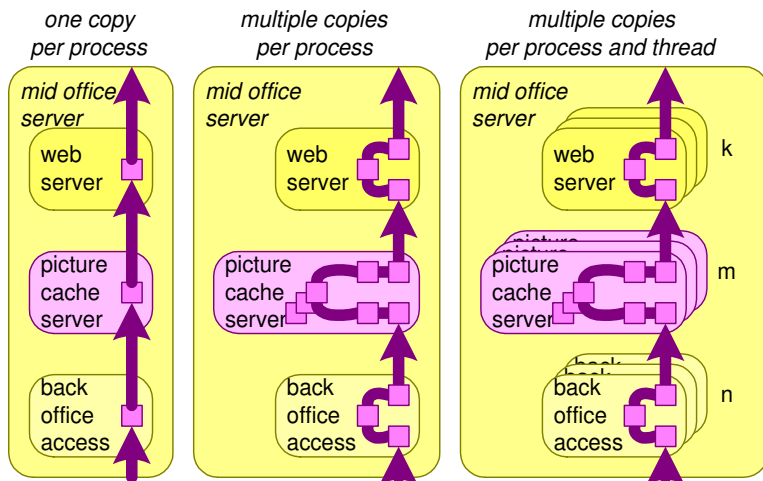
Perform step 2abc on most critical

class 4 and 5 risks

b3: Model Significant Issues

b3. Model significant, non-obvious, issues

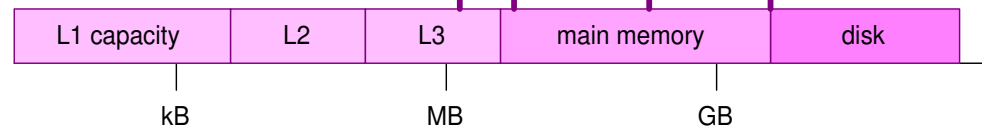
for example, memory use in server(s) for picture transfers and buffering



n	m	k	s	c	MB
1	1	1	1.E+05	10	1.5
2	4	10	1.E+05	20	5.3
2	4	1000	1.E+05	100	296.2
2	4	1000	1.E+06	100	2962.1

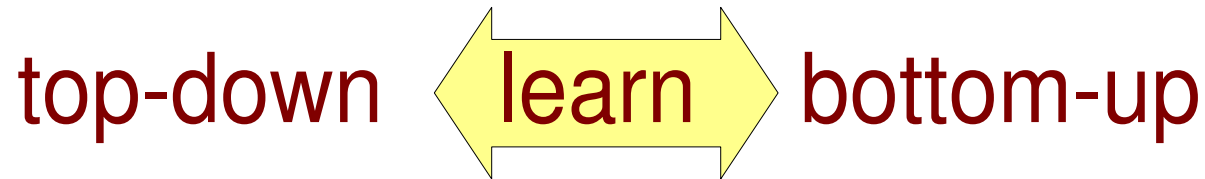
memory use
product browsing only
pictures only
single server

picture memory =
 $3 * n * s +$
 $5 * m * s + c * s +$
 $3 * k * s$
 where
 n = # back office access threads
 m = # picture cache threads
 k = # web server threads
 s = picture size in bytes
 c = in memory cache capacity
 in # pictures



What is the performance impact of memory use on other processing?

Learning Concurrent Bottom-up and Top-down

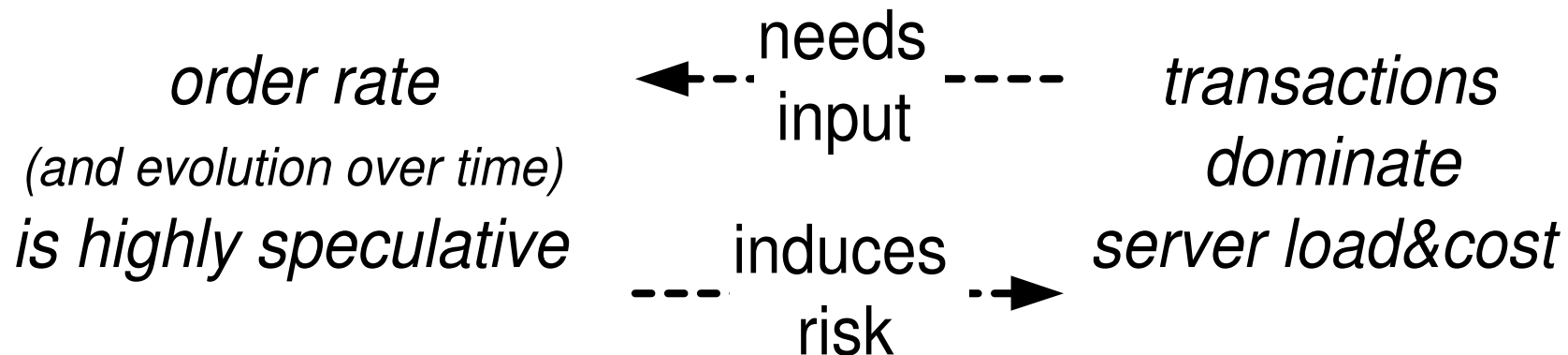


top-down: what is *hidden* in details?

top-down: do we address the *relevant* decomposition?

bottom-up: do we address relevant details?

bottom-up: what details have *significant* impact?



Example top-down and bottom-up

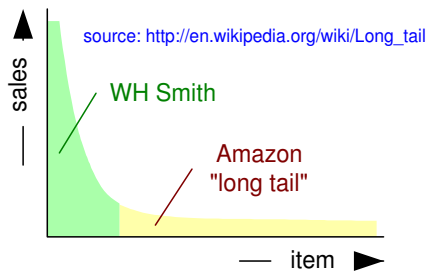
top-down:
what is impact of
catalogue size and changes?

bottom-up:
what is relevant concurrency (k), cache size (c),
or picture size (s)?

new books per year

UK (1)	206k (2005)	107k (1996)
USA(2)	172k (2005)	68k (1996)
China(3)		101k (1994)
India(21)		12k (1996)

source: http://en.wikipedia.org/wiki/Books_published_per_country_per_year



	n	m	k	s	c	MB	storage type
	1	1	1	100,000	10	1.5	L3
	2	4	10	100,000	20	5.3	main
highly concurrent	2	4	1000	100,000	100	296	main
large pictures	2	4	1000	1,000,000	100	2,962	main+disk
many pictures	2	4	10	100,000	100,000	9,540	main+disk
all at once	2	4	1000	1,000,000	100,000	98,234	disk

$$\text{picture memory} = 3 * n * s + 5 * m * s + c * s + 3 * k * s$$

where

n = # back office access threads

m = # picture cache threads

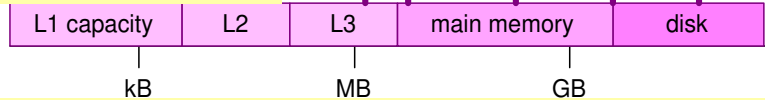
k = # web server threads

s = picture size in bytes

c = in memory cache

capacity in # pictures

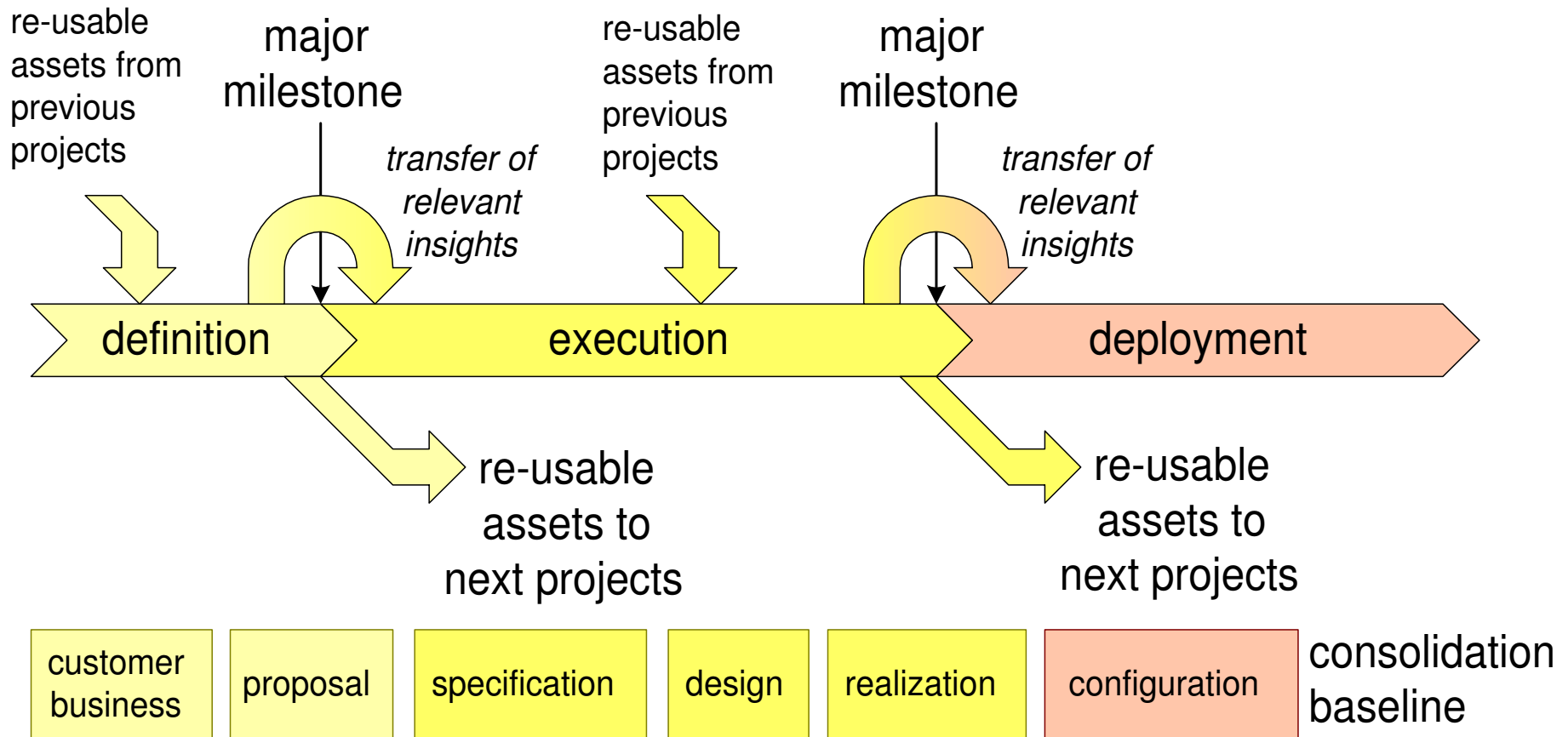
memory use
product browsing only
pictures only
single server



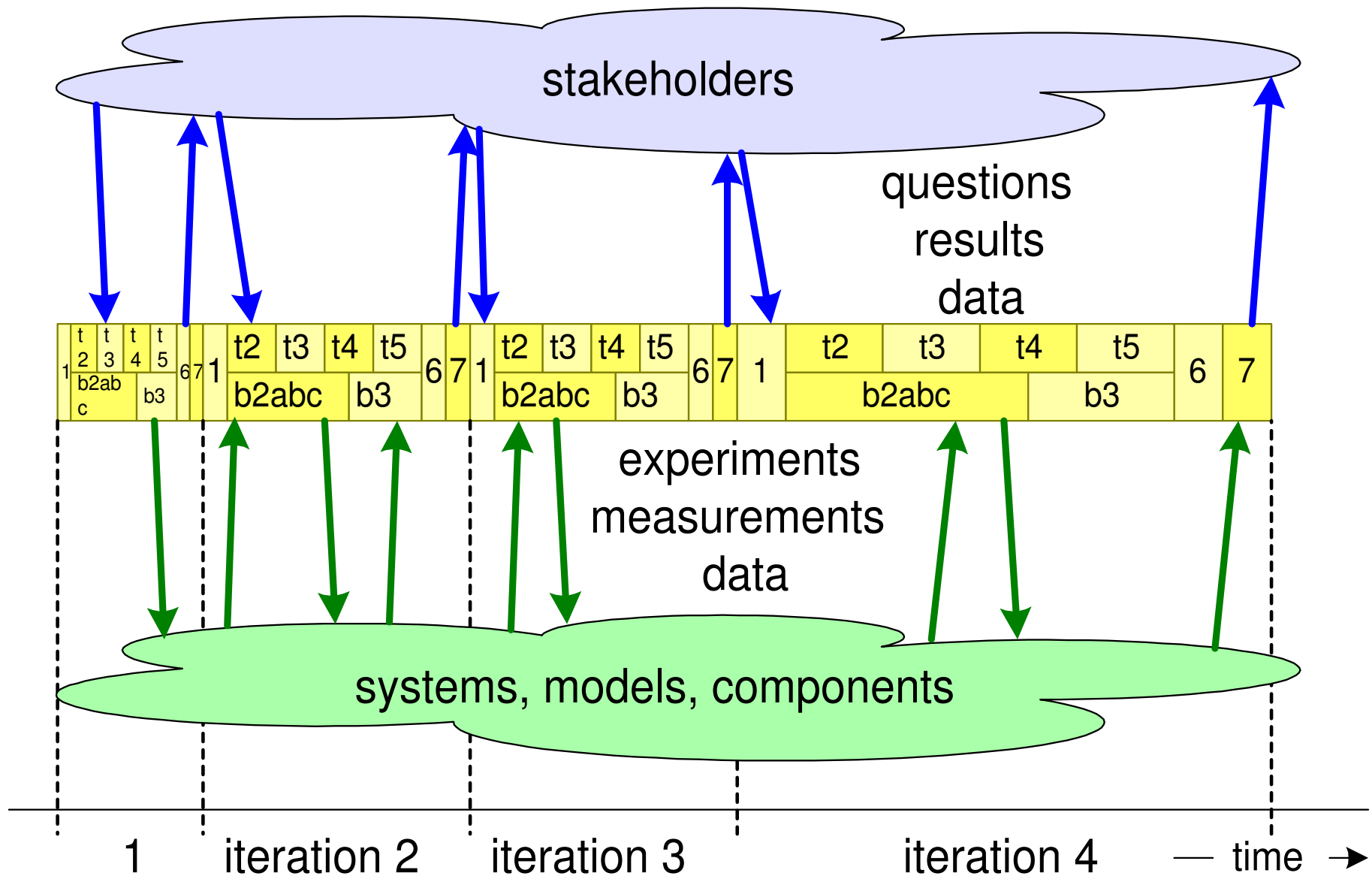
What is the performance impact of memory use on other processing?

6. Capture overview, results and decisions

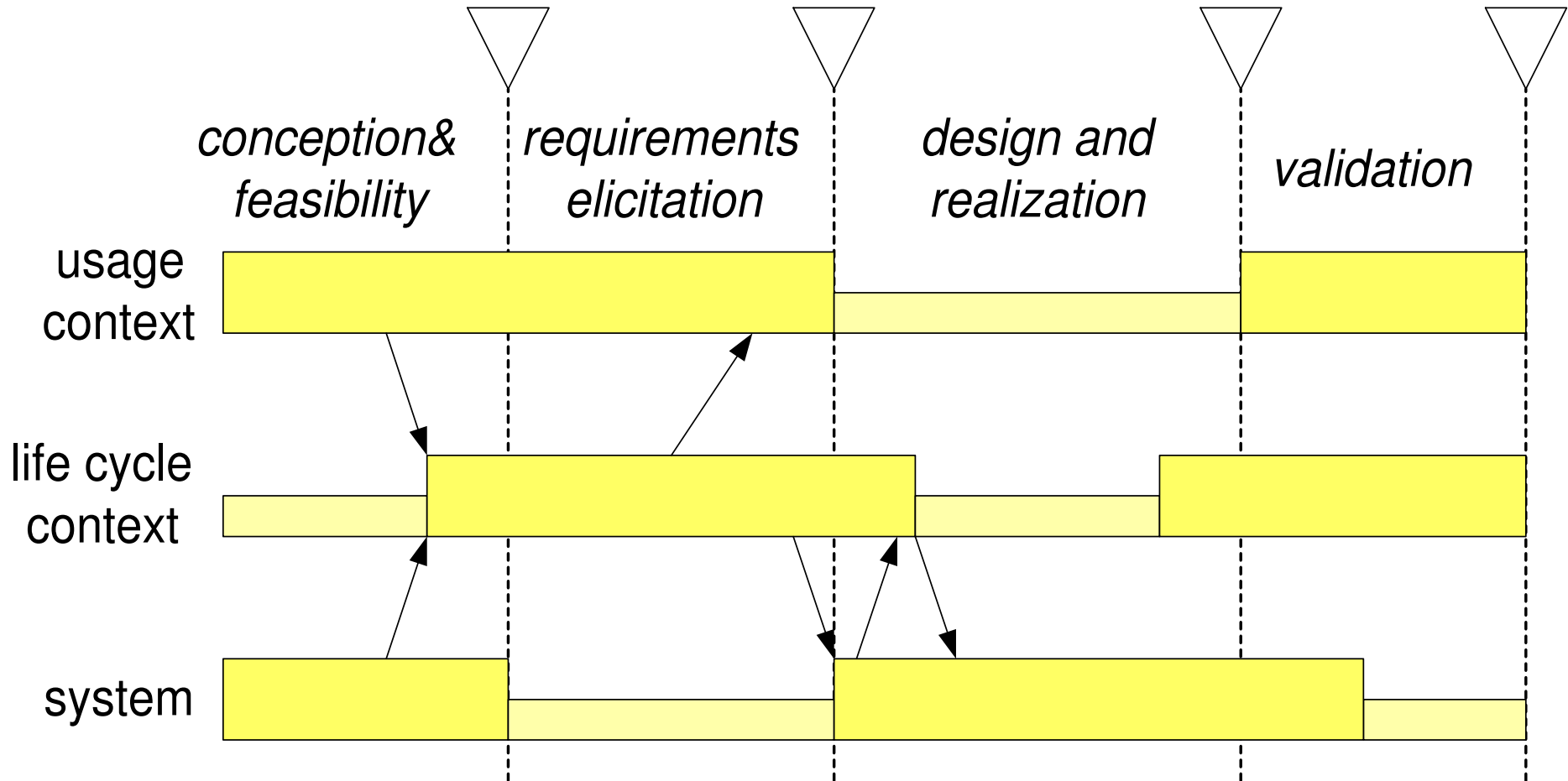
6. Capture overview, results and decisions



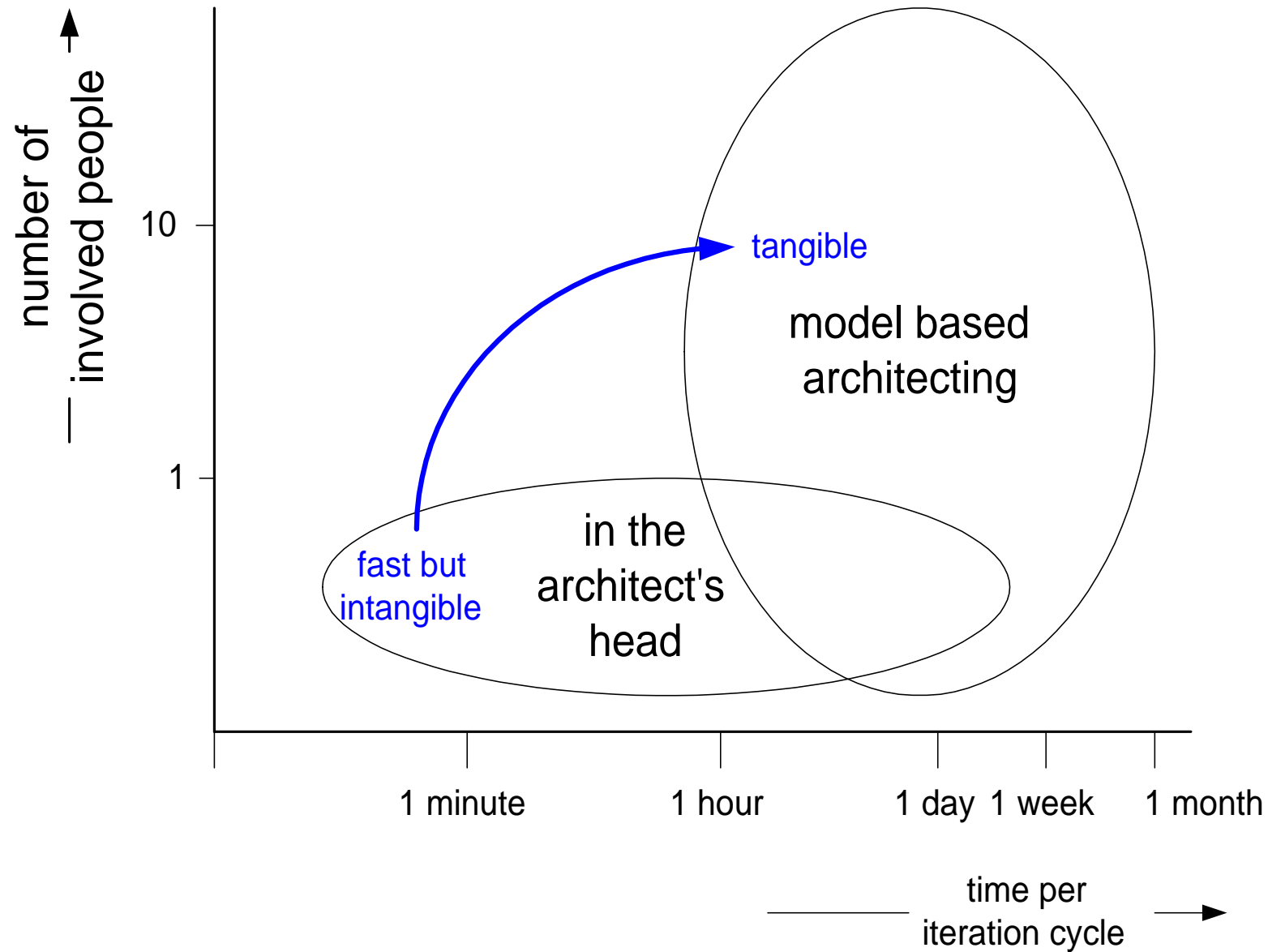
7. Iterate and Validate



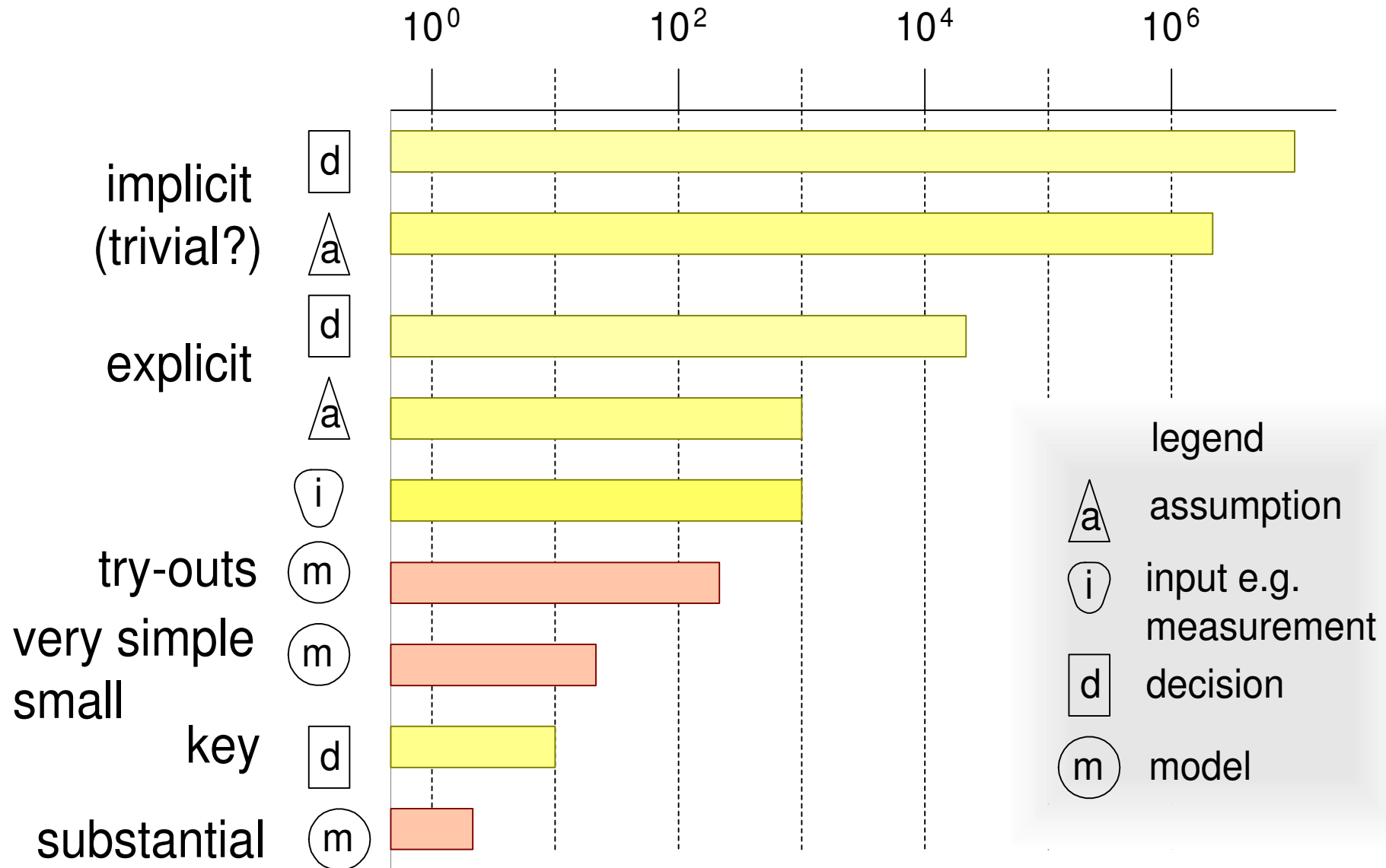
Focus is Shifting during Project



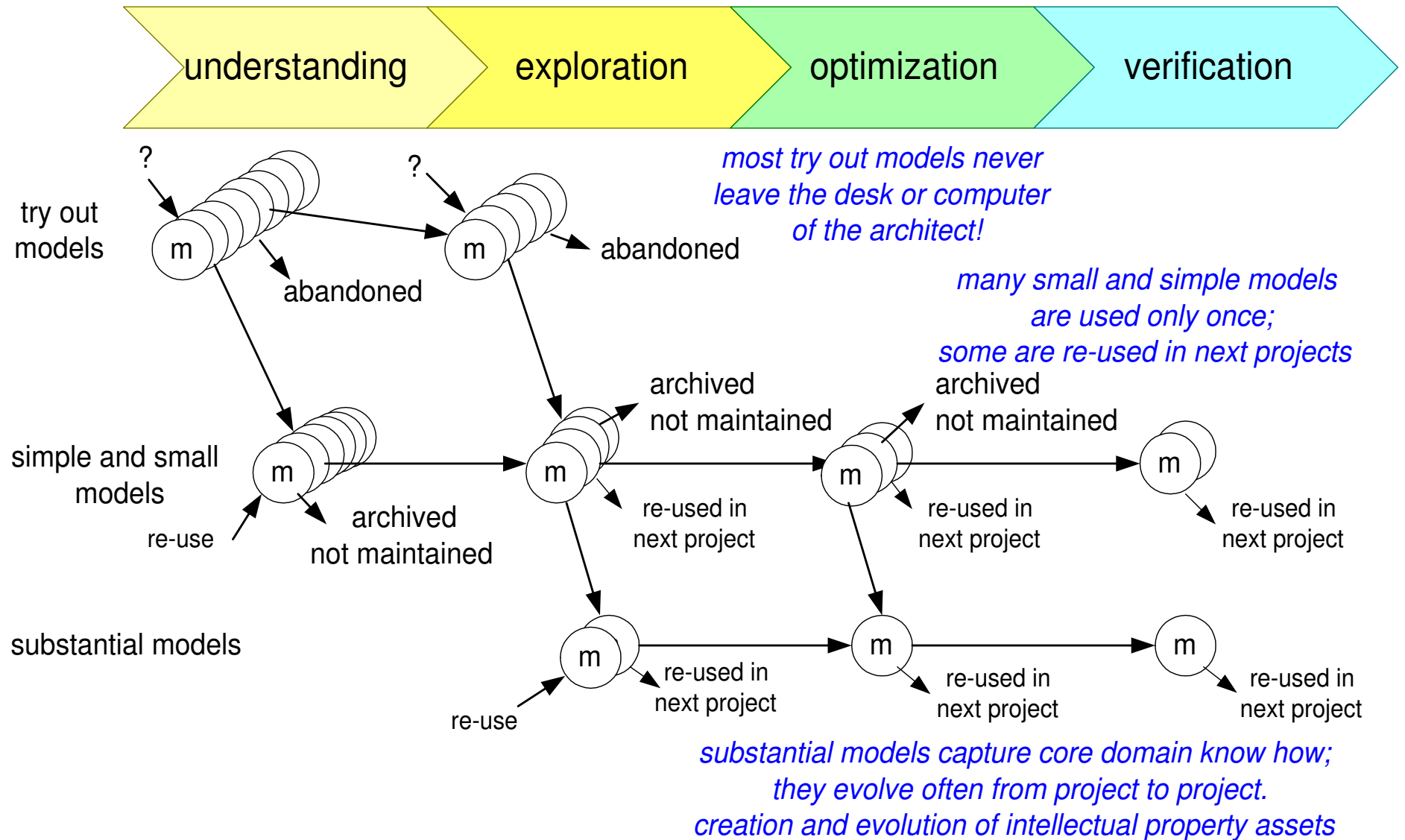
Models Support Communication



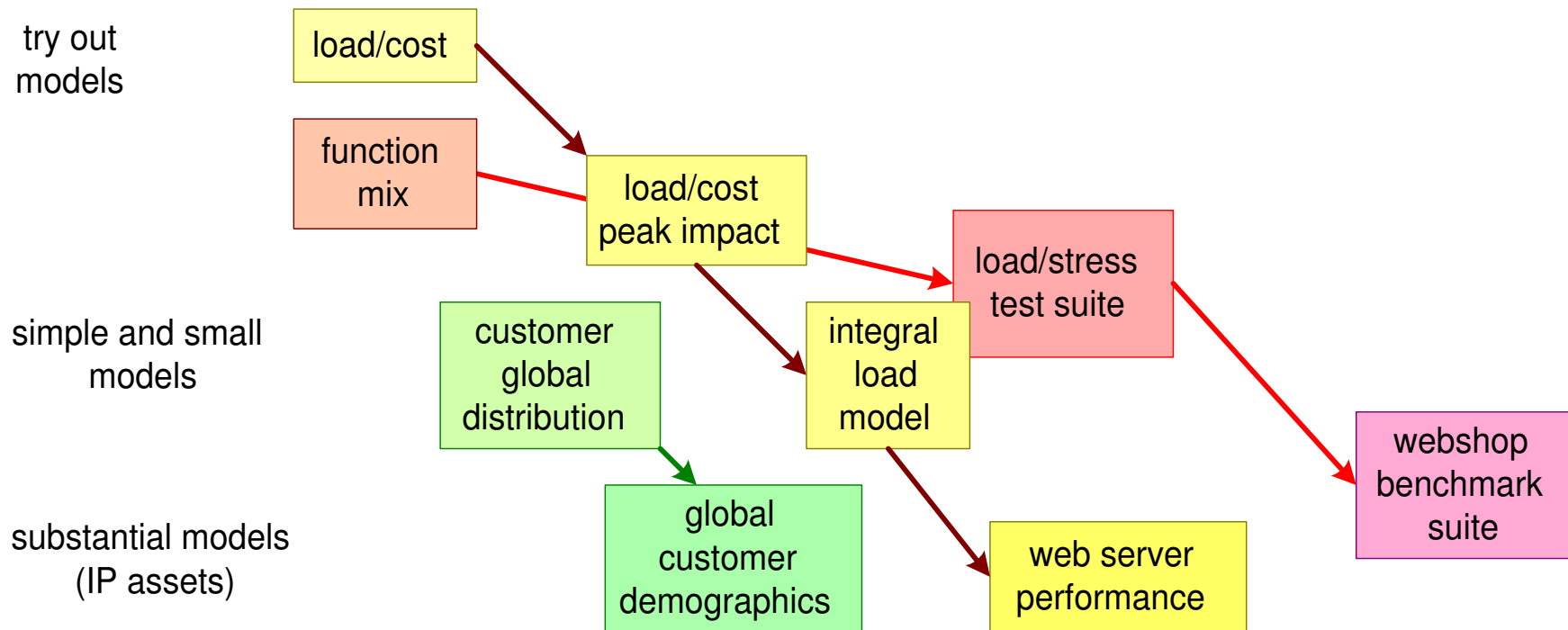
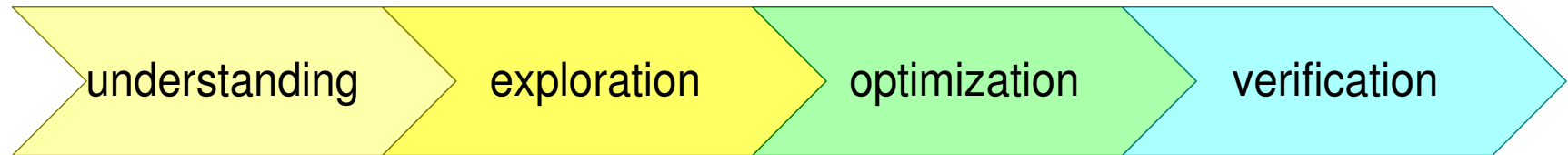
Frequency of Assumptions, Decisions and Modeling



Life Cycle of Models



Examples of Life Cycle of Models



Conclusions

Top-down and bottom-up provide complementary insights

Key words for selection: hottest, non-obvious, significant, relevant

Multiple small models are used in combination

Some models evolve from very simple to more substantial

Techniques, Models, Heuristics of this module

Threads-of-reasoning

SMART

Key Performance Indicators, Key Performance Measures, Critical Resources

Ranking matrices