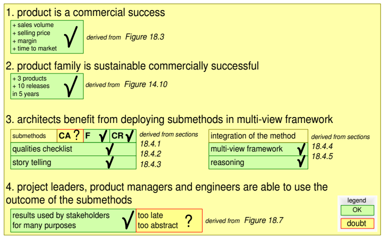
Evaluation of the Architecting Method

by *Gerrit Muller*   Buskerud University College
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

The case study is evaluated: the resulting product and its design and the way the method has been used by the product creation team. The evaluation is done by means of the predefined hypothesis and criteria.

June 23, 2016
status: finished
version: 0.9



Hypothesis and criterions as basis for the evaluation

threads of reasoning

CAFCR submethods
qualities checklists
story telling

CAFCR
quality needles

hypothesis

A rich collection of submethods fitting in a multi-view framework
complemented with reasoning methods enables successful architecting of technology
and software intensive complex systems in heterogeneous environments by means of
generic insights grounded in specific facts

medical
imaging
workstation

small system level
specifications

measurements,
source code, log files
films, images, et cetera

multi-modality
multi-application
multi-release

criteria

1. product is a commercial success
2. product family is sustainable commercially successful
3. architects benefit from deploying submethods in multi-view framework
4. project leaders, product managers and engineers are able to use
   the outcome of the submethods

Evaluation of the product

	Customer objectives / Application	Functional
customer feedback	++ usability film layout ++ film efficiency + operator efficiency printing + ease of auto-printing	+ throughput + image quality + interoperability URF
	- concurrent viewing and auto-printing	- interoperability vascular
operational feedback	+ sales volume + selling price + margin + time to market	+ manufacturability + option handling
	- return on investment	~ network installation

legend

+ good or
++ very good

~ doubt

- problem

Evaluation of the design

Conceptual		Realization
+ notification + Objective-C + standard workstation + X bypass + Unix	+ processing pipeline + graphics + UI toolbox + PMSnet + database engine	+ memory management + DB based communication + SW keys + OIT

~ modularity
~ distance internal and external information model
~ some bloating due to over-genericity
~ property handling

- dependency structure
- interface management

lots of discussions about :
  language choice (why not C++)
  windowing system
  platform re-use

legend

+ good
~ doubt
- problem

based upon technology assessment in "Technology Improvement Plan"

Coverage of submethods

Customer objectives	Application	Functional	Conceptual	Realization
	context diagram	*case descriptions* *commercial decomposition* *service decomposition* *goods flow decomposition* *function and feature specifications* *performance* *external interfaces* *standards*	*construction decomposition* *functional decomposition* *designing with multiple decompositions* *execution architecture* *internal interfaces* *performance* *start up* *shutdown* *integration plan*	*budget* *benchmarking* *performance analysis* *granularity determination*
key drivers **value chain**	**stakeholders and concerns**		**work breakdown** **safety**	**value and cost**
business models suppliers	entity relationship models dynamic models		reliability security	safety analysis reliability analysis security analysis

legend *explicitly addressed* **addressed only implicitly** not addressed

coverage based on documentation status of first product release

Documentation of qualities in 1996

usable

- usability
- attractiveness
- responsiveness
- image quality
- wearability
- storability
- transportability

reliable

- safety
- security
- reliability
- robustness
- integrity

effective

- throughput or productivity

interoperable

- connectivity
- 3rd party extendible

liable

- liability
- testability
- traceability
- standards compliance

efficient

- resource utilisation
- cost of ownership

consistent

- reproducibility
- predictability

serviceable

- serviceability
- configurability
- installability

future proof

- evolvability
- portability
- upgradeability
- extendibility
- maintainability

logistics friendly

- manufacturability
- logistics flexibility
- lead time

ecological

- ecological footprint
- contamination
- noise
- disposability

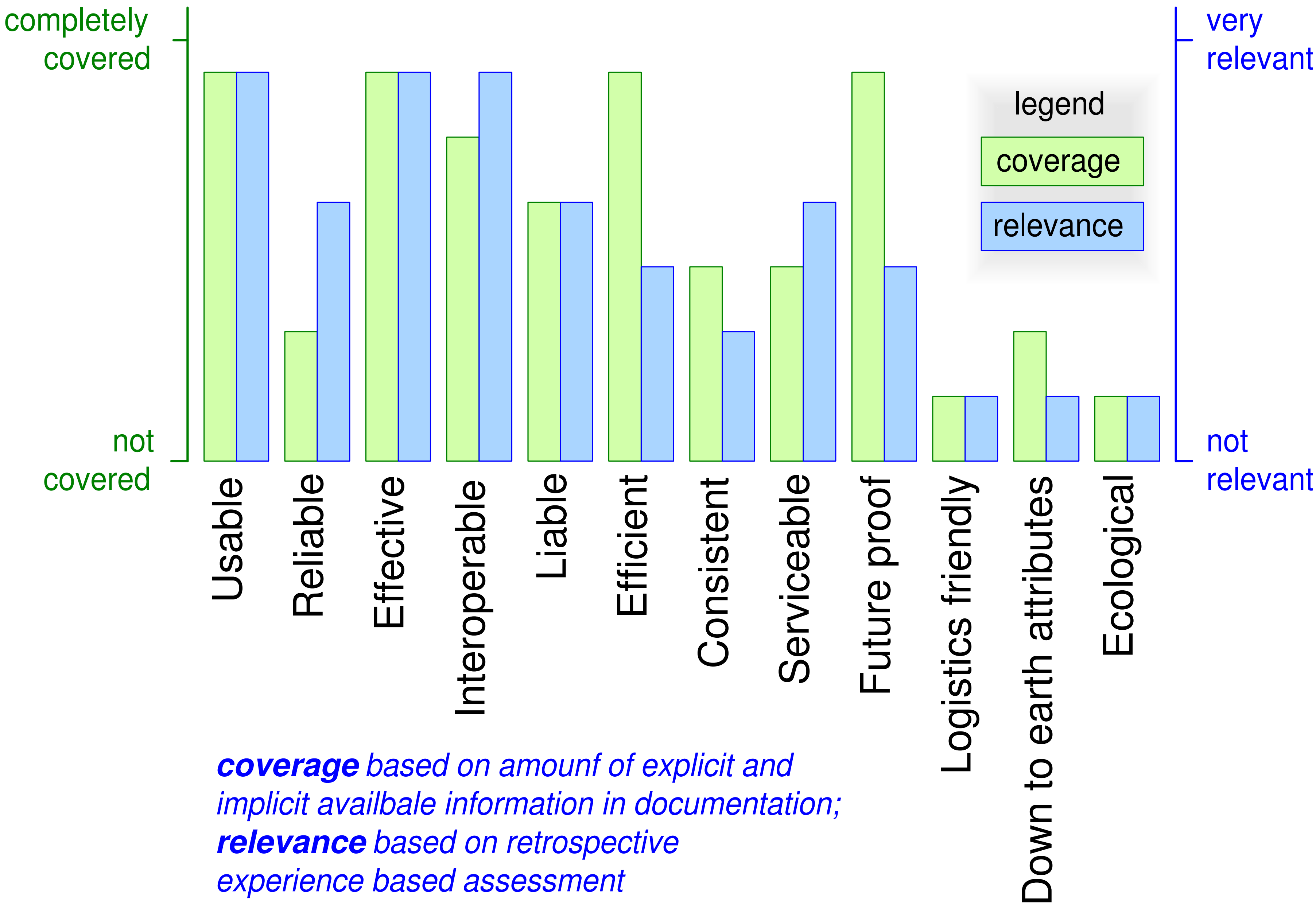
down to earth attributes

- cost price
- power consumption
- consumption rate (water, air, chemicals, et cetera)
- size, weight
- accuracy

legend

- *in separate document*
- *implicit in other documents*

Coverage profile of qualities



***coverage** based on amounf of explicit and implicit availbale information in documentation;*
***relevance** based on retrospective experience based assessment*

Users and usage of the results of the architecting method

results used by:

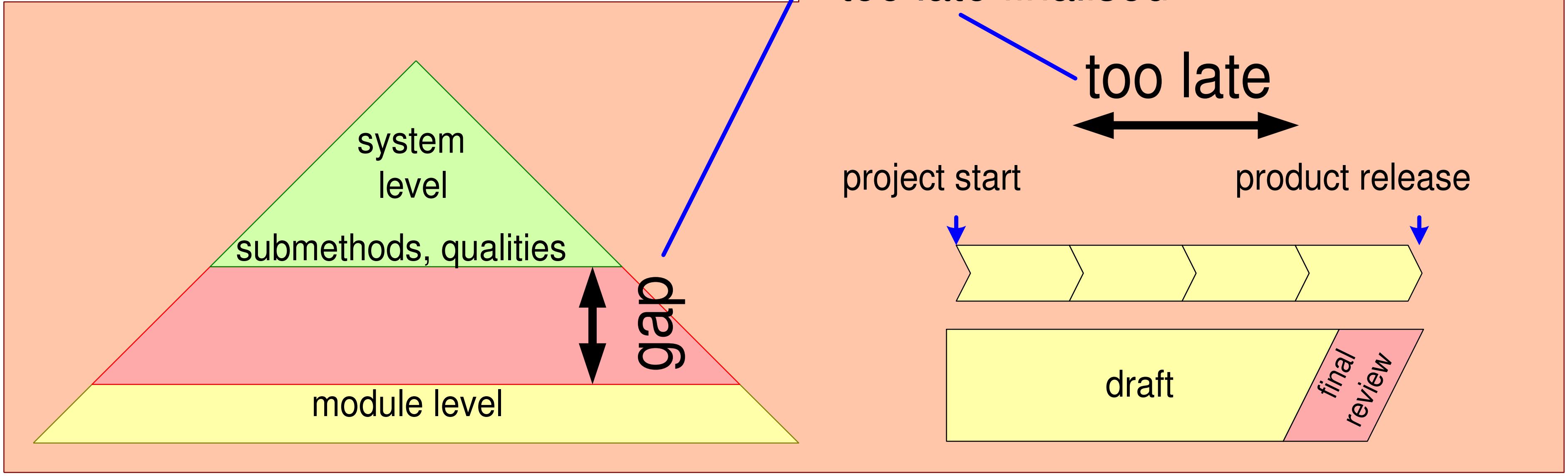
- product management
- application
- project leaders
- engineers
- test engineers
- purchasing
- manufacturing
- suppliers

results used for:

- detailed specifications
- testing
- communication
- derived documentation (manuals)
- used for succeeding products

engineers critics

- too abstract
- too late finalised



The conclusion of the case evaluation

1. product is a commercial success

+ sales volume + selling price + margin + time to market	√

derived from *Figure 18.3*

2. product family is sustainable commercially successful

+ 3 products + 10 releases in 5 years	√

derived from *Figure 14.10*

3. architects benefit from deploying submethods in multi-view framework

submethods	CA ?	F √	CR √
qualities checklist √			
story telling √			

derived from sections
18.4.1
18.4.2
18.4.3

integration of the method	
multi-view framework	√
reasoning	√

derived from sections
18.4.4
18.4.5

4. project leaders, product managers and engineers are able to use the outcome of the submethods

results used by stakeholders for many purposes √	too late too abstract ?

derived from *Figure 18.7*

legend
OK
doubt