

# Evaluation of the Architecting Method

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

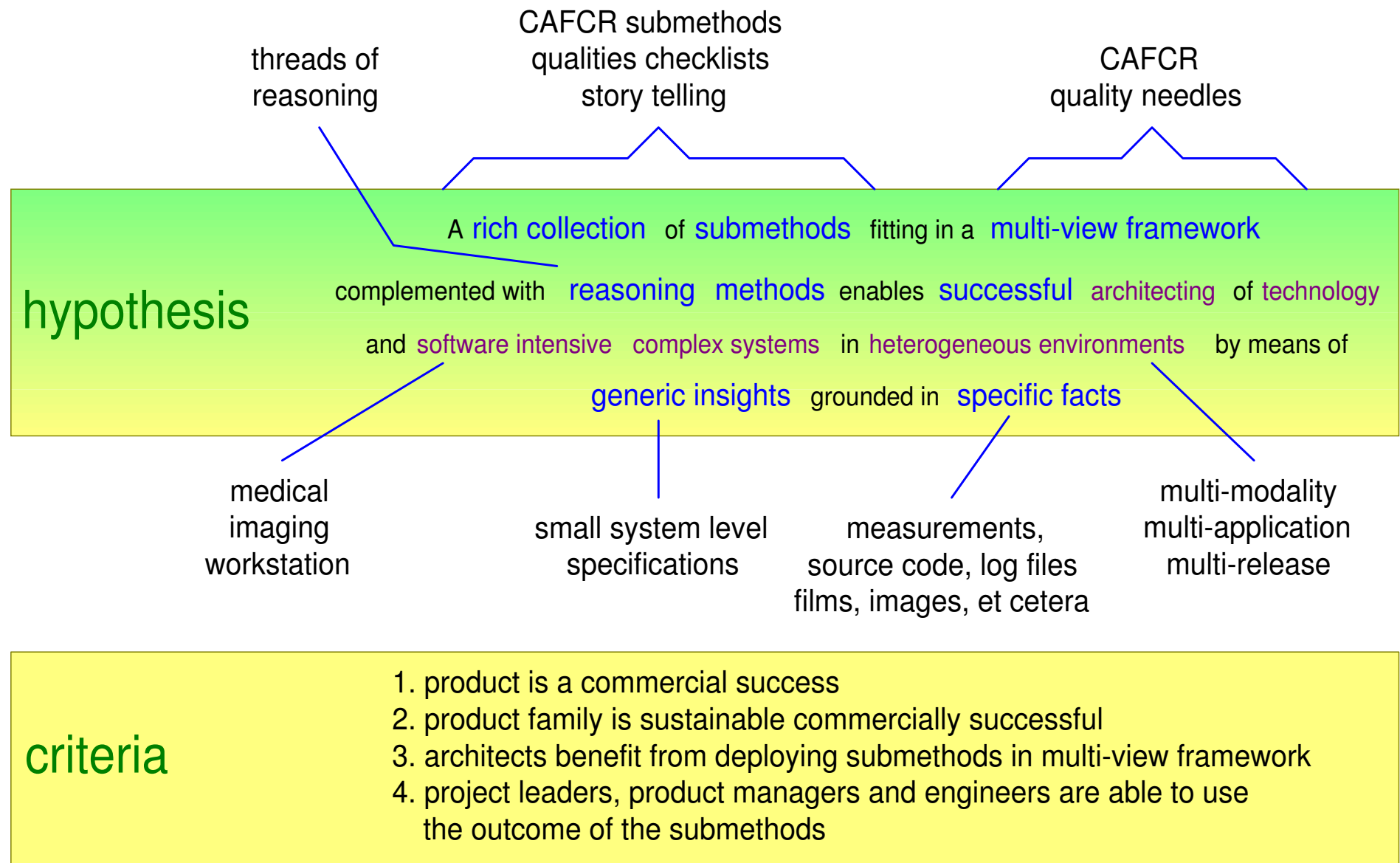
## Distribution

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: finished  
version: 0.9

1. product is a commercial success			
□ sales revenue			
+ selling price	✓	derived from	Figure 18.3
+ margin			
↳ firm to market			
2. product family is sustainable commercially successful			
† 3 products	✓	derived from	Figure 14.10
↳ 10 releases			
↳ 5 years			
3. architects benefit from deploying submethods in multi-view framework			
submethods	CA ?	F ✓	CR ✓
qualifies checklist	✓	18.4.1	integration of the method
story telling	✓	18.4.2	multi-view framework
		18.4.3	reasoning
			18.4.5
4. project leaders, product managers and engineers are able to use the outcome of the submethods			
results used by stakeholders	✓	too late	?
for many purposes		too abstract	
			derived from Figure 18.7
			Legend
			OK
			doubt

# Hypothesis and criteria as basis for the evaluation



# Evaluation of the product

	<b>C</b> ustomer objectives	<b>A</b> pplication	<b>F</b> unctional	
customer feedback	++ usability film layout ++ film efficiency + operator efficiency printing + ease of auto-printing - concurrent viewing and auto-printing		+ throughput + image quality + interoperability URF - interoperability vascular	<b>legend</b> + good or ++ very good ~ doubt - problem
operational feedback	+ sales volume + selling price + margin + time to market - return on investment		+ manufacturability + option handling ~ network installation	

# Evaluation of the design

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

<b>C</b> ustomer objectives	<b>A</b> pplication	<b>F</b> unctional	<b>C</b> onceptual	<b>R</b> ealization
<p><b>key drivers</b> <b>value chain</b></p> <p>business models suppliers</p>	<p><i>context diagram</i></p> <p><b>stakeholders and concerns</b></p> <p>entity relationship models dynamic models</p>	<p><i>case descriptions</i> <i>commercial decomposition</i> <i>service decomposition</i> <i>goods flow decomposition</i> <i>function and feature specifications</i> <i>performance external interfaces</i> <i>standards</i></p>	<p><i>construction decomposition</i> <i>functional decomposition</i> <i>designing with multiple decompositions</i> <i>execution architecture</i> <i>internal interfaces</i> <i>performance</i> <i>start up</i> <i>shutdown</i> <i>integration plan</i></p> <p><b>work breakdown</b> <b>safety</b></p> <p>reliability security</p>	<p><i>budget</i> <i>benchmarking</i> <i>performance analysis</i> <i>granularity</i> <i>determination</i></p> <p><b>value and cost</b></p> <p>safety analysis reliability analysis security analysis</p>

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  
3<sup>rd</sup> 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

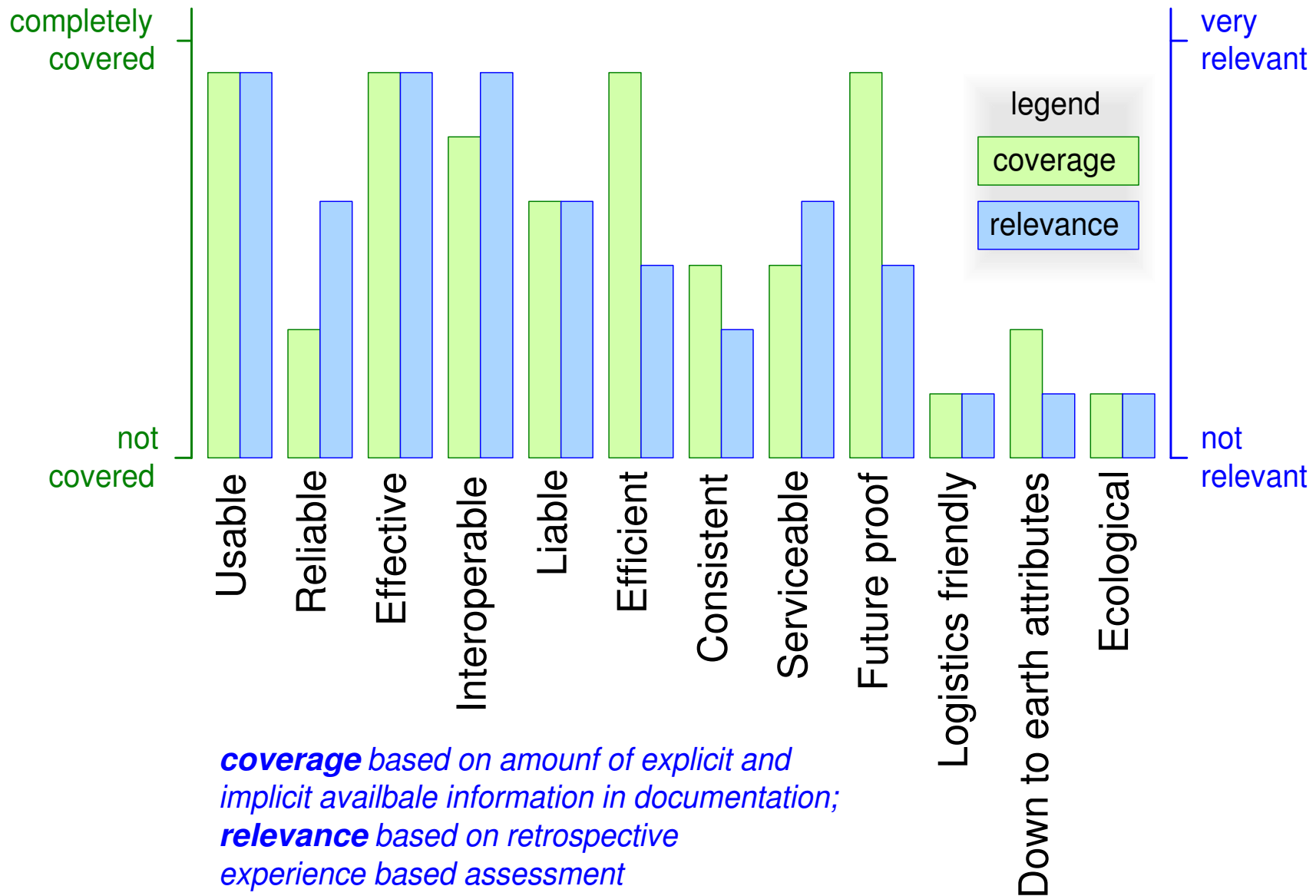
## down to earth attributes

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



# Users and usage of the results of the architecting method

*results used by:*

product management  
application  
project leaders  
engineers  
test engineers  
purchasing  
manufacturing  
suppliers



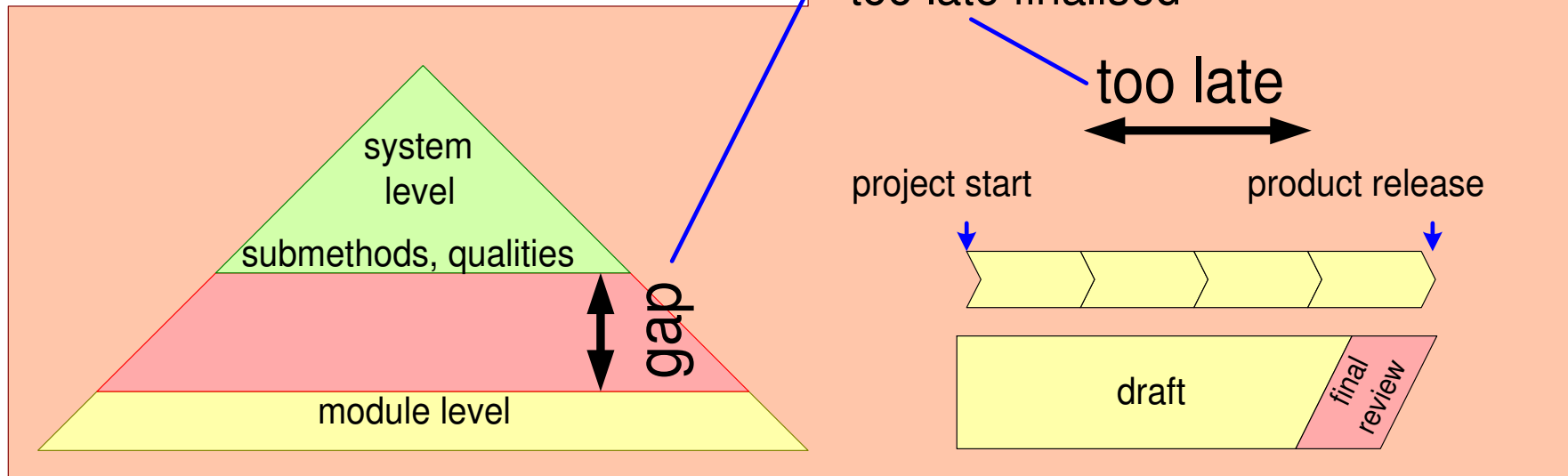
*results used for:*

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



derived from sections

integration of the method

derived from sections

qualities checklist



18.4.1

multi-view framework



18.4.4

story telling



18.4.2

reasoning



18.4.5

18.4.3

## 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