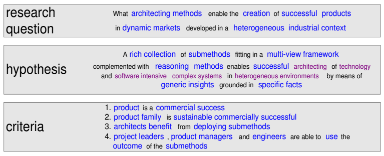
# Research Question and Hypothesis

-





Gerrit Muller

University of South-Eastern Norway-NISE

Hasbergsvei 36 P.O. Box 235, NO-3603 Kongsberg Norway

gaudisite@gmail.com



## Abstract

The research question and objectives with respect to architecting methods are defined. The scope is product-family and product level primarily. An hypothesis is formulated, that is based on a multi-view framework allowing many submethods to be used. Criteria for architecting methods are articulated.





version: 4.5 status: finished September 9, 2018

# 1 Introduction

The starting point for the investigation of architecting methods in this thesis is the research question, articulated in Section 2. An hypothesis is formulated in Section 3. The criteria to validate the hypothesis are defined in Section 4. Section 5 summarizes all three aspects in a single overview, and shows how the different parts of the thesis fit together in this thesis.

# 2 Research Question

Figure 1 shows the annotated research question. The core research question is *what architecting methods enable the creation of successful products*. This core question is more focused by adding a more specific environment: *a dynamic market* and a *heterogeneous industrial context*. The product category is also narrowed down by looking only at *technology and software intensive* products.

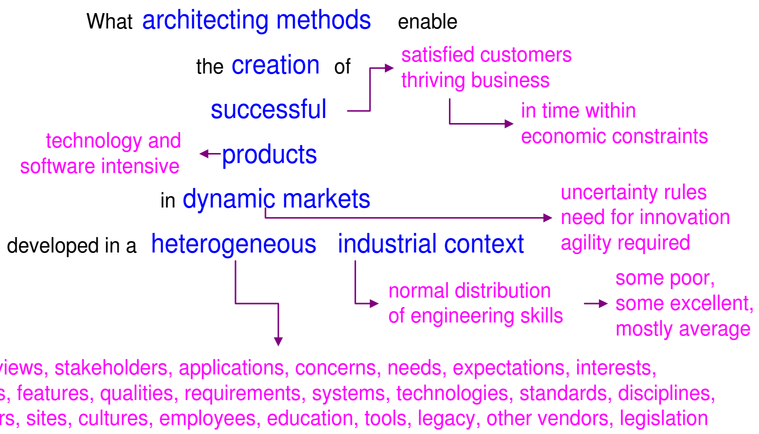



Figure 1: Research question

*Successful products* are products that satisfy the customers and result in a thriving business. In present-day economy this means that time plays a dominant role, products must be *in time*. The economic reality and the hefty competition force economic constraints on the created product and its timing. This economic reality requires pragmatism in the architecting methods. Many academic methods, however, suffer from a mismatch with these economic and time constraints. For instance formal verification systems work well for small well-defined problems, but require too much time and skills to be useful in larger, more uncertain, problems.

Most product areas become more and more dynamic: customers, competitors, and other stakeholders interact in complex ways, with a lot of uncertainty. Active commercial product lifetimes have decreased from years to months. Globalization enables unexpected competitors to enter the market, based on low-cost labor and huge work-forces. Constant innovation is required to stay competitive. To follow the rapid market changes agile procedures and organizations are required.

The context in which we operate is characterized by an increasing variability and complexity. Products should fit in a very heterogeneous world. The heterogeneity is present in aspects such as: *views, stakeholders, applications, concerns, needs, expectations, interests, functions, features, qualities, requirements, systems, technologies, standards, disciplines, suppliers, sites, cultures, employees, education, tools, legacy, other vendors, legislation*.

The industrial context in which the methods have to be used has a population of engineers with a normal distribution of engineering skills and intellect. Some have poor skills, some have excellent skills, but most engineers have average skills. This is a severe constraint on the architecting methods. Some very nice methods are too difficult to apply in practical organizations. Note that the research question is what methods enable the product creation in the *industrial context*. This does not imply that the constraint is that they should fit entirely in the existing crew. Crew and method should be matched, but the degrees of freedom in composing a PCP team in an industrial context are quite limited.

## 3 Hypothesis

The variability of products being created is so large, that **one** all encompassing method is impossible. The dynamic range in requirements spans many orders of magnitude. For example requirements for power consumption, weight, and processing needs differ a factor of 1000 for products such as televisions and GSM cellphones. It is more feasible to grow a rich collection of submethods than to develop a single all encompassing method. Submethods are methods that address a smaller part of the problem. This step moves the problem to another area: how to combine multiple submethods in a useful way?

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**

Figure 2: Hypothesis

The hypothesis, as shown in Figure 2 formulates the need for a *multi-view framework*. The submethods must fit in the multi-view framework. *Reasoning* *methods* are needed to cope with multiple submethods.

The claim is that this combination of a *rich collection of submethods*, *multi-view framework*, and *reasoning methods* enables *the successful architecting of technology and software intensive complex systems in heterogeneous environments*.

The generic term *product* in the research question is replaced by the more focused notion of *technology and software intensive systems*. The research described in this thesis has been limited to embedded systems. Embedded systems are systems with embedded computing hardware and software that have interaction with the physical world. This interaction with the physical world is technology intensive, for example actuator technology and sensor technology.

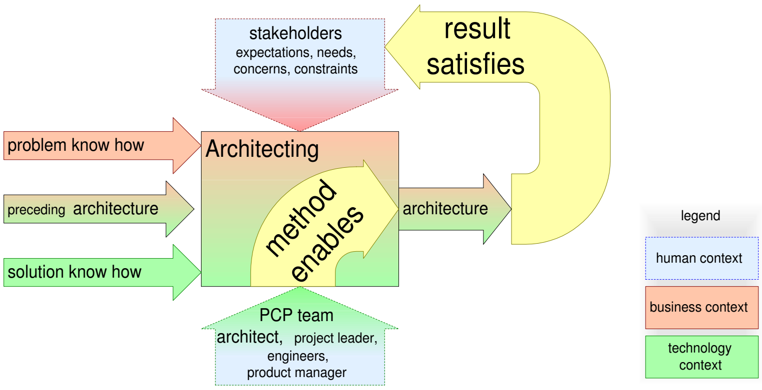



Figure 3: The hypothesis is valid if *successful* architecting is *enabled*.

In addition two crucial characteristics of architecting work are added to the claim: the use of *generic insights* grounded in *specific facts*. These two characteristics seem to be contradictory: generic insights are often interpreted as ignoring the details (=specific facts); some details, however, are often needed to appreciate the essence captured in the generic insight.

Architecting methods need sufficient genericity to have impact. Architects will lose overview when they have to specify every product detail. The challenge is to extract the essence from specific facts in such a way that powerful and trustworthy generic insights are created.

Many product developments fail in combining the specific facts and the generic insights. Discussions during the SARCH courses [1] often show organizations inside and outside Philips where the PCP teams spend all their time in details. The policy makers in these same organizations are disconnected from the rest of the PCP team. The PCP team is working on *specific* details, while the policy makers are working on *generic* insights, but the two worlds are disconnected. The consequence of the disconnection is that product innovations fail. Small improvements are made by the PCP team, but the larger changes fail because important details have not been taken into account.

Figure 3 addresses the term *successful* in the hypothesis. This is done in a two-step approach from PCP team to stakeholders. The PCP team is successfully enabled by an architecting method if the use of the method resulted in the creation of a successful architecture. An architecture is successful if the stakeholders are satisfied with the result.

# 4 Criteria

Figure 4 shows the criteria to be used, based on the two step approach shown in Figure 3.

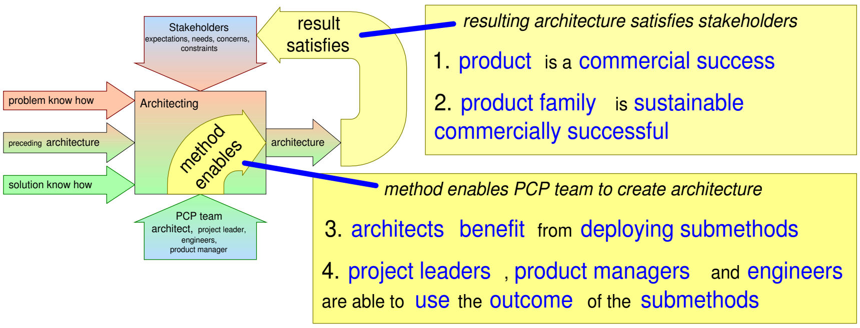



Figure 4: From hypothesis to criteria

*The resulting architecture satisfies the stakeholders* is indirectly verified by measuring the *1. short term commercial success of the product* and the *2. sustainability of this commercial success in the following product family*. The underlying assumption is that satisfied customers buy more products and motivate other customers to buy this satisfactory product. Dissatisfied customers have a negative impact on the sales.

In order to exclude incidental success, the long term commercial success is also required. This long term success can only be measured by means of follow-on products. The active commercial lifetime of products (1 to 2 years) is too short to measure long term commercial success with the product itself. The follow-on product family, based on the same architecture and architecting method, is used instead.

*The architecting method enables the PCP team to create a successful architecture* is sharpened by defining two criteria. The first criterion is for the architect as primary user: the architect(s) must be able to use the submethods to achieve a good architecture. Note that the hypothesis mentions *submethods complemented with reasoning methods*. The criterion for success is not that all submethods are useable, but the criterion is that *3. the architect benefits from the collection of submethods*.

The second criterion to assess is the enabling of the product creation team, especially the (non-architect) members of the PCP team: *4. the outcome of the architecting method must be usable for the other members of the Product Creation Team (or PCP Team): project leaders, product managers and engineers.*

The quality of the design of the product contributes to the sustainability of the product. The quality of design is also one of the measures for the support that the method provides to architects. Another measure for the support is the integration: How well was integration supported by the chosen integration?

# 5 Summary

In this chapter we have discussed the research question, the objectives of the architecting method, the hypothesis about the architecting method improvements, and the criteria to evaluate the method. Figure 5 shows the summary of this chapter: research question, hypothesis and criteria.

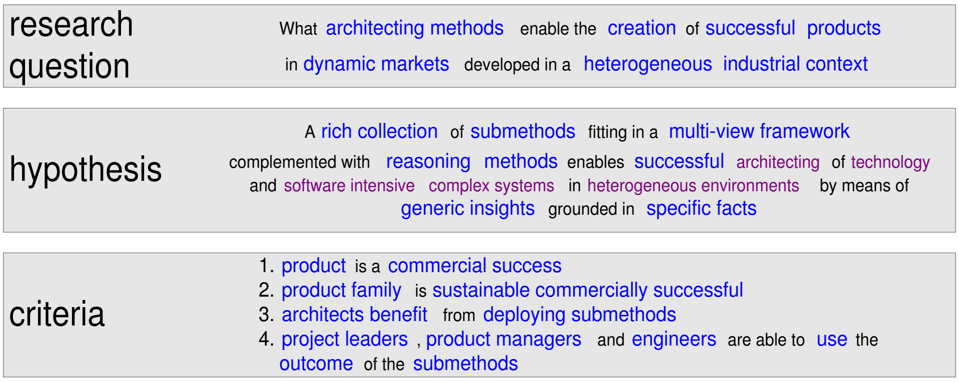



Figure 5: Overview of research question, hypothesis and criteria.

Part II will show the theory of the architecting method. Part III describes the case that is used for the evaluation. In Part IV, in chapters **??** and **??**, the hypothesis and criteria are used for the evaluation.

# 6 Acknowledgements "Criteria for architecting methods"

Wim Vree challenged me to articulate the research question, objective, hypothesis and criteria and gave lots of valuable feedback during the creation. Lia Muller-Charité helped me by discussing the criteria. Martin Rem provided a lot of feedback.

## References


[1] Paul de Witte. CTT course SARCH. `http://www.extra.research.philips.com/ctt/ctt_it/sarch.htm`, 1999.

[2] Gerrit Muller. The system architecture homepage. `http://www.gaudisite.nl/index.html`, 1999.


### History

**Version: 4.5, date: April 7, 2004 changed by: Gerrit Muller**

- removed citation to light weight architecting
- changed title from “criteria for architecting methods” into “Research Question and Hypothesis”

**Version: 4.4, date: April 6, 2004 changed by: Gerrit Muller**

- changed Criterions into Criteria
- added reference to evaluation chapter
- refactored the paragraph: "Structure of this thesis". Part of it moved to the summary, the figure has been moved to the Thesis introduction.
- changed status into finished

**Version: 4.3, date: February 27, 2004 changed by: Gerrit Muller**

- updated figure to improve greylevel representation

**Version: 4.2, date: January 13, 2004 changed by: Gerrit Muller**

- Added design and integration as measures of the criterions.
- few small textual improvements
- changed status to concept

**Version: 4.1, date: December 2, 2003 changed by: Gerrit Muller**

- appended Section *Structure of this thesis* to this chapter.
- many small textual improvements

**Version: 4.0, date: October 2, 2003 changed by: Gerrit Muller**

- changed status into *draft*.
- removed many language errors and rephrased sentences

**Version: 3.0, date: July 28, 2003 changed by: Gerrit Muller**

- removed section ”Objectives”
- removed section ”Scope of architecting method”
- created summary section, moved overview figure from introduction to summary.

**Version: 2.0, date: May 13, 2003 changed by: Gerrit Muller**

- removed some text and diagrams from section ”Criterions”
- added text and diagram from hypothesis to criterions to section ”Criterions”
- added text to section ”Research question”
- added text to section ”Hypothesis”
- updated overview figure
- added text to section ”Scope of architecting method”
- set status to preliminary draft

**Version: 1.4, date: March 24, 2003 changed by: Gerrit Muller**

- added overview diagram to introduction

**Version: 1.3, date: March 24, 2003 changed by: Gerrit Muller**

- updated research question
- removed annotated objective
- updated more detailed objectives
- added success criterion to hypothesis

**Version: 1.2, date: March 21, 2003 changed by: Gerrit Muller**

- updated scope figure
- added objectives in relation with research question
- moved scope section backwards

**Version: 1.1, date: February 11, 2003 changed by: Gerrit Muller**

- updated research question (”normal” engineers -> industrial context)
- updated objective (added ”complex”, added annotation for ”heterogeneous”)

**Version: 1.0, date: February 11, 2003 changed by: Gerrit Muller**

- removed figures responsibilities, what is architecting, architecture how
- added research question
- updated objective
- added hypothesis
- changed abstract

**Version: 0.1, date: January 14, 2003 changed by: Gerrit Muller**

- Added objective and objectives diagram

**Version: 0, date: November 11, 2002 changed by: Gerrit Muller**

- Created, no changelog yet