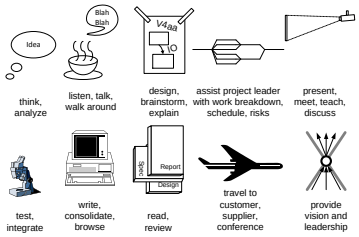
# The Role and Task of the System Architect

-




Gerrit Muller
Buskerud University Collge
Hasbergsvei 36 P.O. Box 235, NO-3603 Kongsberg Norway
gaudisite@gmail.com

*This paper has been integrated in the book "Systems Architecting: A Business Perspective", http://www.gaudisite.nl/SABP.html, published by CRC Press in 2011.*

## Abstract

The role of the system architect is described from three viewpoints: deliverables, responsibilities and activities. This description shows the inherent tension in this role: a small set of hard deliverables, covering a fuzzy set of responsibilities, hiding an enormous amount of barely visible day-to-day work.



version: 2.0 status: concept June 5, 2018

# 1 Introduction

Architects and organizations are often struggling with the role of the system architect (or software architect or any other kind of architect). This struggle is partially caused by the intangible nature of the responsibilities of the architect. At the other hand (good) architects are highly appreciated, even if their quantifiable output is low.

This article starts with specific deliverables, then discusses the more abstract responsibilities and, finally, discusses the day to day activities of an architect.

The role of the software architect is nicely discussed in [1].

# 2 Deliverables of the System Architect

We start at looking for the tangible output that is expected from architects. Project leaders and program managers do expect deliverables to be finished at appropriate milestones. Most Product Creation Processes define the deliverables of a System Architect to be artifacts such as documents or models. These artifacts are symbolized by the stack in Figure 1.

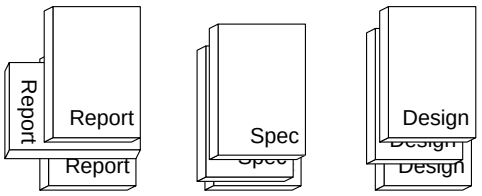



Figure 1: Deliverables of a system architect consists of artifacts forming a stack of paper when printed

Figure 2 shows the main deliverables of a System Architect more specific. Quite often the System Architect does not even produce all deliverables mentioned here, but the architect does take the responsibility for these deliverables by coordinating and integrating contributions of others. Note that some of these deliverables are part of the Policy and Planning Process.

# 3 System Architect Responsibilities

The System Architect has a limited set of primary responsibilities, as visualized in figure 3. The primary responsibilities are:

**Balance** of system properties as well as internal design properties. The system should be balanced: for example, the cost of subsystems should correspond

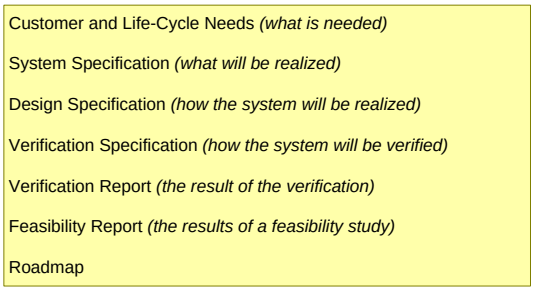



Figure 2: More specific list of deliverables of a System Architect

with its added value in terms of functionality and performance. Architecting is a continuous balancing act in many incomparable dimensions and quantities.

**Consistency** across many organizational and design boundaries; From needs to implementation details, from system level to detailed implementation.

**Decomposition, Integration** Decomposition is the standard answer in dealing with complex and big problems. Decomposing Systems in subsystems, subsystems in modules et cetera is a major responsibility of the architect. In most systems many decomposition dimensions are required: physical, logical, functional, and many more, see [3]. The complementary action of decomposition, however, is integration. The integral functioning and performance of the system is the ultimate goal of product creation, which emphasizes the importance of integration. In practice integration is much more difficult than decomposition, in fact the architect must decompose in such a way that integration is feasible.

**Overview** of the entire system and its context helps to make sensible specification and design decisions. The architect should provide overview to all members of the product creation team. Most of these members have a very limited horizon. The architect should help them by providing proper context information to make local design decisions.

**Elegance, Simplicity** are properties of a "good" architecture. The dangerous aspect of this responsibility is the highly subjective nature of elegance and simplicity. The appreciation of simplicity and elegance should be assessed or acknowledged by others than the architect.

**Integrity** of the system specification and design over time. The focus of a development team is often wandering over time, sometimes it depends on the hype of the week. The architect is responsible for maintaining a balanced

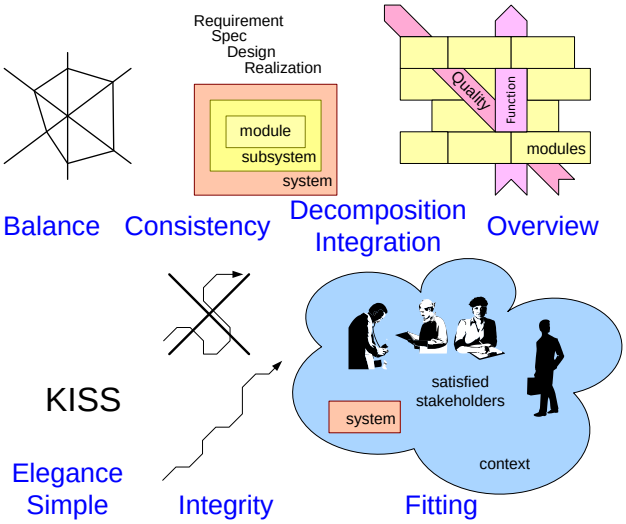



Figure 3: The primary responsibilities of the system architect are not tangible and easily measurable

and focused development over time. For instance, when cost price reduction is required then the architect should keep performance and reliability on the agenda.

**Fitting** in stakeholder needs and system context, during the entire life cycle, is one of the core responsibilities of the architect. The architect must connect depth knowledge with breadth knowledge.

We can condense the primary responsibility of the System Architect as: to ensure the good functioning of the System Architecting Process. In practice, this responsibility is often shared by a team of System Architects, with one chief architect taking the overall responsibility.

The list of primary responsibilities as discussed above is suffering from a lack of measurability and is rather intangible. Systems Architects also have secondary responsibilities, where these are primarily owned by other persons. Most other roles in product creation are much sharper defined, as shown in Figure 4. For instance the business manager is responsible for the business plan and the financial results. The project leader is responsible for the schedule and hence for completing the project in time and within budget. The marketing manager is responsible for addressing the relevant markets and hence for market share and salability of the product. The technology manager is responsible for the timely availability of technologies and related tools. The line manager is responsible for the availability of the right people, with skills and processes to do their job. Final example are the engineers who are responsible for the design of their component or module.

| responsibility | primary owner |
|---|---|
| business plan, profit | business manager |
| schedule, resources | project leader |
| market, saleability | marketing manager |
| technology | technology manager |
| process, people | line manager |
| detailed designs | engineers |

Figure 4: (Incomplete) list of secondary responsibilities of the system architect and the related primary owner

# 4 What does the System Architect do?

Figure 5 shows the variety of activities of the day to day work of a system architect. A large amount of time is spent in gathering, filtering, processing and discussing detailed data in an informal setting. These activities are complemented by more formal activities like meetings, visits, reviews et cetera.

The system architect is rapidly switching between specific detailed views and abstract higher level views. The concurrent development of these views is a key characteristic of the way a system architect works.

*Abstractions only exist for concrete facts*

System Architects which stay too long at "high" abstraction levels drift away from reality, by creating their own virtual reality.

Figure 6 shows the bottom up elicitation of higher level views. A system architect sees a tremendous amount of details, most of these details are skipped, a smaller amount is analyzed or discussed. A small subset of these discussed details is shared as an issue with a broader team of designers and architects. Finally, the system architect consolidates the outcome in a limited set of views. The order of magnitude numbers cover the activities in one year.

The opposite flow in 6 is the implementation of many of the responsibilities of the system architect. By providing overview, insight and fact-based direction a simple, elegant, balanced and consistent design will crystalize, where the integrity of designs goals and solutions are maintained during the project.

A lot of time spent by the architect serves the purpose of communication between many project members. The architect not only responsible for the system integration, but has also an integrating role in the project itself. The architect has to interact a lot with all the people mentioned in Figure 4, in order to fulfil the architect's responsibilities.

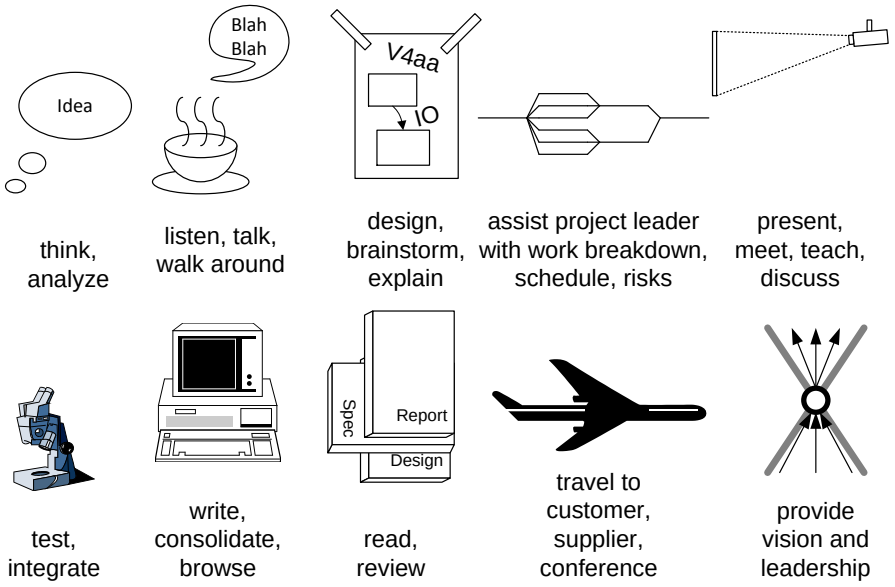



Figure 5: The System Architect performs a large amount of activities, where most of the activities are barely visible for the environment, while they are crucial for the functioning of architects

# 5 Task versus Role

The task of the system architect is to generate the agreed deliverables, see section 2 This measurable output is requested and tracked by the related managers: project leaders and the line managers. Many managers appreciate their architects only for this visible subset of their work.

The deliverables are only one of the means to fulfil the System Architect Responsibilities, as described in section 3. The system architect is doing a lot of nearly invisible work to achieve the system level goals, his primary responsibility. This work is described in section 4. Figure 7 shows this as a pyramid or iceberg: the top is clearly visible, the majority of the work is hidden in the bottom.

# 6 Acknowledgements

Nicolette Yovanof pointed out that the text belonging to Figure 2 and Table 2 was rather incomplete. She also mentioned that some more attention for the interaction with non-architects would be helpful. Chuck Kilmer provided feedback on ”The Awakening of a System Architect”, which resulted also in an update of this paper. Byeong Ho Gong suggested a better coverage of the interfacing with

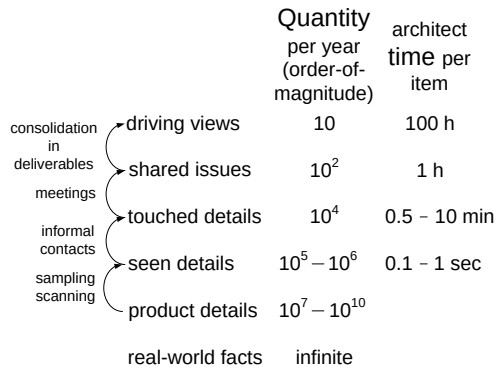



Figure 6: Bottom up elicitation of high level views

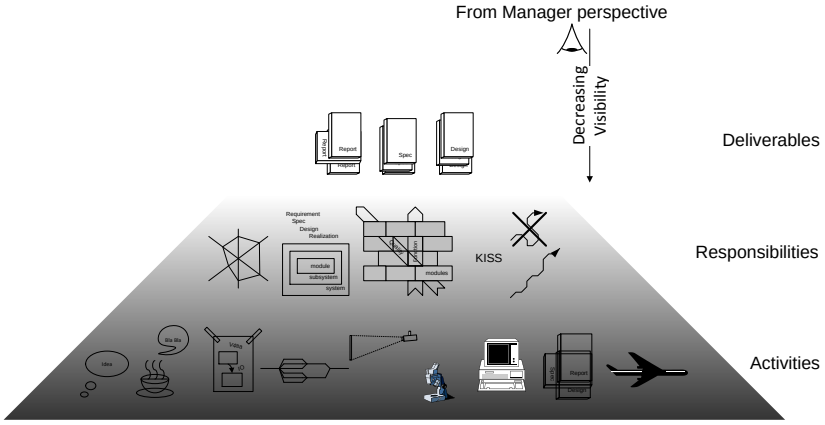



Figure 7: The visible outputs versus the (nearly) invisible work at the bottom

customers/stakeholders. Pierre van de Laar provided textual improvements.

# References

[1] Dana Bredemeyer and Ruth Malan. Role of the software architect. `http://www.bredemeyer.com/pdf_files/role.pdf`, 1999.

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

[3] Gerrit Muller. Architectural reasoning explained. `http://www.gaudisite.nl/ArchitecturalReasoningBook.pdf`, 2002.

## History

## Version: 2.0, date: June 27, 2010 changed by: Gerrit Muller

- changed status to concept

**Version: 1.4, date: June 7, 2010 changed by: Gerrit Muller**
- changed tables with deliverables and secondary responsibilities into figures

**Version: 1.3, date: June 5, 2008 changed by: Gerrit Muller**
- textual improvements

**Version: 1.2, date: May 23, 2006 changed by: Gerrit Muller**
- extended acknowledgements section

**Version: 1.1, date: April 25, 2006 changed by: Gerrit Muller**
- added providing vision and leadership to activities

**Version: 1.0, date: April 21, 2006 changed by: Gerrit Muller**
- updated Figure 2
- added more text for Figure 2 and Table 2
- added text about the interaction between architect and others
- changed status to draft

**Version: 0.4, date: December 8, 2005 changed by: Gerrit Muller**
- updated reference to Bredemeyer paper

**Version: 0.3, date: August 5, 2002 changed by: Gerrit Muller**
- Added introduction

**Version: 0.1, date: September 13, 2000 changed by: Gerrit Muller, Pierre America**
- Small editorial changes only

**Version: 0, date: October 10, 2000 changed by: Gerrit Muller**
- Created, no changelog yet