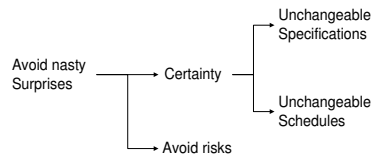


System Architecture: The Silver Bullet?

-



Gerrit Muller

Embedded Systems Institute

Den Dolech 2 (Laplace Building 0.10) P.O. Box 513, 5600 MB Eindhoven The Netherlands

`gerrit.muller@embeddedsystems.nl`

Abstract

Many people jump onto systems architecting methods with the false expectation that these methods will solve most of their problems. A major reason to do this is uncertainty. The real world, unfortunately, is full of uncertainties, and systems architecting can help in dealing with these many uncertainties. However, systems architecting will not make uncertainties disappear, neither will it prevent effort to be spent on unexpected issues.

Silver Bullets do not exist.

The critical success factors for applying system architecture methods are described.

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.

All Gaudí documents are available at:
<http://www.gaudisite.nl/>

1 Introduction

The expectation level with respect to processes in general and the system architecture process in particular can vary from skeptical to blind faith. The skeptics have experienced that horrible specifications and designs can be pursued under the grand name of Architecture. The followers with blind faith are at the opposite end of the spectrum, their believe in processes inhibits them from seeing the limitations and constraints from the processes applied.

The central message of this Intermezzo is:

Silver Bullets do not exist.

This Intermezzo intends to set realistic expectation levels with respect to the System Architecture Process, and describes the ingredients for successful application.

2 Why System Architecture?

System Architecting is a means to create systems efficient and effective, by supplying overview, by guarding consistency and integrity, and by balancing. In other words the System Architect helps the development team to find its way in a rather complex, dynamic and uncertain world.

From psychological point of view people apply their own survival mechanisms, when they perceive a threat. One of the most common survival mechanisms is *The Quest for Certainty*, see subsection 2.1.

Unfortunately System Architecting will never remove all uncertainties, see subsection 2.2. The application of a system architecture process can help in the risk management, amongst others by prevention, and by minimizing impact.

Successful application of system architecture is far from trivial, section 3 describes **how** the System Architecture Process should be applied to meet the goals of efficiency and effectively.

2.1 The Quest for Certainty

This section provides a caricatural view on human behavior based on a free interpretation of the Maslow Hierarchy of Needs, as discussed for instance in [1]. This exaggerated view matches with the security needs in the lower Maslow Hierarchy. Note that less defensive behavior can be triggered by needs in the higher layers, were words such as adventurous en explorative are being used.

The majority of people, including managers and engineers, have a need for certainty. Their ideal is to have stable, unchangeable sets of specifications, schedules et cetera. This (hopefully) isolates them from the nasty surprises of reality see table 1.

- incompetent people
- human mistakes
- lack of collaboration or synergy
- misunderstanding or miscommunication
- changing markets
- fast moving competition
- unforeseen physical, chemical, mechanical properties
- mother nature (illnesses, floods)

Table 1: *Nasty Surprises of Reality*

Unfortunately these nasty surprises are a fact of life. Our human capability to control these phenomena is quite limited.

Risk management can help to be more robust. However risk management certainly does not remove these phenomena and it also does not reduce the consequences to zero. Risk management balances probability, effect, and cost.

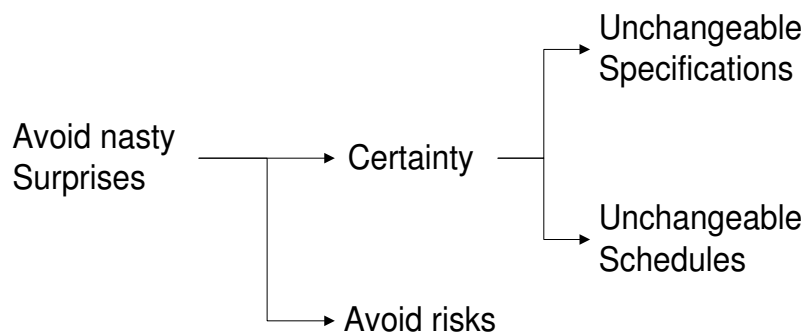


Figure 1: Personal key driver *to avoid nasty surprises*

People with a need for certainty are willing to accept any method or process which promises certainty. In other words *certainty* appears to be their personal key driver. It is better to rephrase this key driver as *to avoid nasty surprises*, which is closer to the internal motivation at the one hand and which gives a handle later on to manage the expectations. Figure 1 visualizes these drivers.

2.2 Disclaimer; Setting the Expectations to a realistic level

The Gaudí Project will not deliver a Plug-and-Play System Architecture Process. System architects which have read all the articles and followed the course will not automatically be successful.

The Gaudí project will deliver a large set of consistent background material for system architects. This material ranges from process and architecture principles, providing insight and understanding, to more specific how-to's which provide more directly applicable guidelines.

The competent system architect will use the material by customizing it to the specific problem to be addressed. At the same time the system architect will have to interact with the environment to share this customized way of working.

Whenever the material is applied literal, this is a strong indication that the organization and the system architect do not work explicit enough on the way of working.

3 How: Critical Success Factors

Ingredients for an effective application of a system architecture process are shown in table 2.

- Know-How
- Common Sense
- Pragmatics
- Critical attitude
- Social skills
- Drive
- Vision

Table 2: *Critical Success Factors for an effective application of a System Architecture Process*

No method or process will function without these critical success factors. A process can not be used as substitute for know how or common sense.

3.1 Know-How

The core of the system architecture work is know-how, ranging from pure technology know-how to application and business know-how. Active control on a broad basis is a prerequisite for a system architect.

3.2 Common Sense

Most problems encountered during Product Creation require common sense to solve them. Mechanistic approaches severely limit the solution space, resulting in complex solutions. System architects are capable of "lateral" thinking, allowing solutions in previously unexpected directions.

3.3 Pragmatics

The holistic approach can easily derail in a sea of seemingly conflicting requirements and viewpoints. The system architect needs a significant amount of pragmatism to be selective and focused, while being holistic in the back of his mind.

3.4 Critical attitude

Clear diagrams, tables with facts and smooth presentations give the impression of high quality and increase the confidence. However these same diagrams, tables and presentations conceal the forgotten, misinterpreted, or underestimated facts. The system architect must always be alert, for instance by asking questions as shown in table 3.

- Do we address the right problem or requirement?
- Is the customer/user on-board?
- Is this design adequate?
- Consists the input data from facts, wishes or ideas?
- Do we need so many people for the implementation?
- Does this process or organization fit the problem?

Table 3: *Critical Attitude: Examples of questions to be asked by the System Architect*

3.5 Drive

A good system architect has a passion for his architecture, it has an emotional value.

An architect which is working entirely according to the book, obediently going through the motions, will produce a clinical architecture without drive or ownership.

Good architectures have an identity of themselves, which originate in the drive of the architect. Such an architecture is an evolving entity, which is appreciated by the stakeholders.

3.6 Vision

The system architect needs to have a vision to be able to provide direction. A vision enables an evolution of existing architectures to desired architectures. Having vision is not trivial, it requires a good understanding of needs (the problem) and means (the solution) plus the trends (opportunities and threats) in both needs and means.

4 Summary

The one sentence summary of this intermezzo is: *Silver bullets do not exist*. Table 4 gives a bullet-wise summary.

- Most people want to avoid nasty surprises
- Most people are looking for certainty
- Silver Bullets do not exist
- System Architecture is not a golden bullet
- Critical Success Factors: Know-How, Common Sense, Pragmatics, Critical attitude, Drive and Vision

Table 4: *Summary*

5 Acknowledgements

Hans Gieles suggested improvements to increase the cohesion and the red line in this Intermezzo.

Henk Obbink and Angelo Hulshout and many others pointed out that "The Golden Bullet" should have been "The Silver Bullet", which has been changed finally. Adriaan van den Brand pointed out the forgotten change of golden into silver.

Eugene Ivanov pointed out that evolution aspects were missing. The result is the addition of vision as critical success factor.

Steve R. Nanning indicated unclarity in the abstract and criticized the too negative phrasing of the *Quest for Certainty*. Stephen Boggess added a question to the list in Section "Critical Attitude". Ning Lu added more surprises and critical success factors.

References

- [1] W. Huitt. Maslow's hierarchy of needs. *Educational Psychology Interactive*. Valdosta, GA: Valdosta State University., 2004.
- [2] Gerrit Muller. The system architecture homepage. <http://www.gaudisite.nl/index.html>, 1999.

History

Version: 1.8, date: May 23, 2006 changed by: Gerrit Muller

- added a few more "nasty surprises".
- added social skills to critical success factors

Version: 1.7, date: April 21, 2006 changed by: Gerrit Muller

- added a question to "Critical Attitude".

Version: 1.6, date: April 19, 2006 changed by: Gerrit Muller

- reworded the abstract
- added a paragraph and a reference to "Quest for Certainty"

Version: 1.5, date: February 19, 2003 changed by: Gerrit Muller

- changed forgotten "golden" in "silver"

Version: 1.4, date: August 5, 2002 changed by: Gerrit Muller

- small updates

Version: 1.2, date: June 20, 2001 changed by: Gerrit Muller

- Added abstract
- Added "vision" as critical success factor

Version: 1.1, date: March 15, 2001 changed by: Gerrit Muller

- Renamed and propagated new layout

Version: 1.0, date: April 7, 2000 changed by: Gerrit Muller

- Created, no changelog yet

Version: 0, date: March 24, 2000 changed by: Gerrit Muller

- Created, no changelog yet