The Future of Systems Engineering

Outcome of the SSSE Round Table in Systems Engineering on 2016-01-28 Version 1.1*, dated 2016-02-08

Sarah Bostock, Pierfelice Ciancia, Marcel Frikart, Werner Gerritsen, Stefan Hänggi, Jesko G. Lamm, Andreas Mettauer, Frauke Regge, Markus Walker, Jörg Sekler, Daniel Weberhofer

Members of the German, Italian and Swiss chapter of INCOSE
Contact: Jesko Lamm, jesko@ssse.ch

Abstract: How will a holistic Systems Engineering approach look in future, to overcome silos in organizations? This was the topic of the round table in Systems Engineering on January 28th, 2016. This paper presents the outcome.

Introduction

Systems Engineering (SE) has the ambition to overcome the formation of "silos" in the organization by applying systems thinking. Nevertheless, Systems Engineering itself creates new silos: the requirements silo, the architecture silo, the human/machine interaction silo, the safety silo, the verification silo.

What will a holistic Systems Engineering approach look like in future, to overcome both kinds of silos, so the ones created in engineering domains, but also the ones created by Systems Engineering itself?

These were key questions discussed at SSSE's round table in Systems Engineering during the meeting held at Aarbergerhof Bern on January 28th, 2016. This paper presents the method and the outcome of the discussion.

Method

The authors met and split into two groups. They spent two hours to approach the topic "the future of systems engineering", based on the lead questions below that had been prepared in advance by some of the authors:

- What will a holistic Systems Engineering approach look like in future, to overcome forming silos around different systems engineering processes?
- How will requirements, architecture and verification integrate seamlessly? How will they link to specifications and models in the engineering domains?

---

* Corrects some typos compared to the version 1.0 that was dated 2016-01-28

- How will a stakeholder quickly find the relevant information? For example, embedded software engineers in the need to know specifications affecting the current piece of code they are working on.
- How will Systems Engineering facilitate fast impact analysis?
- Is model-based Systems Engineering (MBSE) really the universal answer to all this?
- How do we manage the data that is needed for all the above?

Results

Silos in Systems Engineering and in Engineering Domains

Some of the authors can indeed confirm the existence of silos in Systems Engineering from their practical experience (Figure 1) – but there are also experiences underlining that the engineering domains behave like silos towards Systems Engineering, e.g. not aligning architecture on different levels (Figure 2). However, it was also stated that silos are not bad per se, because they may also help separating different domains, keeping each of them on a certain level of autonomy, which can again lead to more efficiency in both making progress and maintaining existing work products.

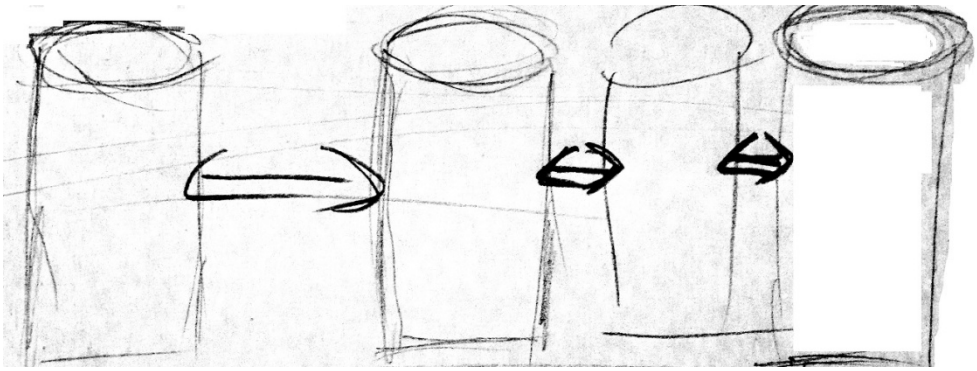

Figure 1: Interacting Silos in Systems Engineering

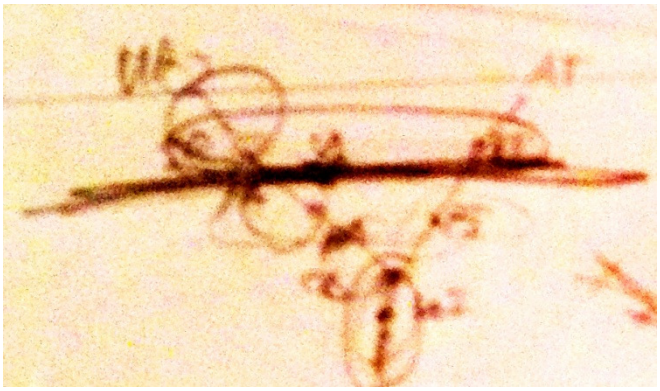

Figure 2: Silos between different abstraction levels of architecture, depicted by means of the V model (here: System Architecture, above the thick line and subsystem architecture, below the thick line)

Where silos are considered a problem they could be overcome by building more cross-systems engineering knowledge. Systems Engineers with a broad knowledge of various systems engineering processes can bridge silos inside systems engineering. This will enable systems engineering to become an end-to-end service in product development, starting at the start of a project and ending at the end of the lifecycle.

In providing the said end-to-end service, different systems engineering processes will have their time of high relevance in different phases of the project (Figure 3).

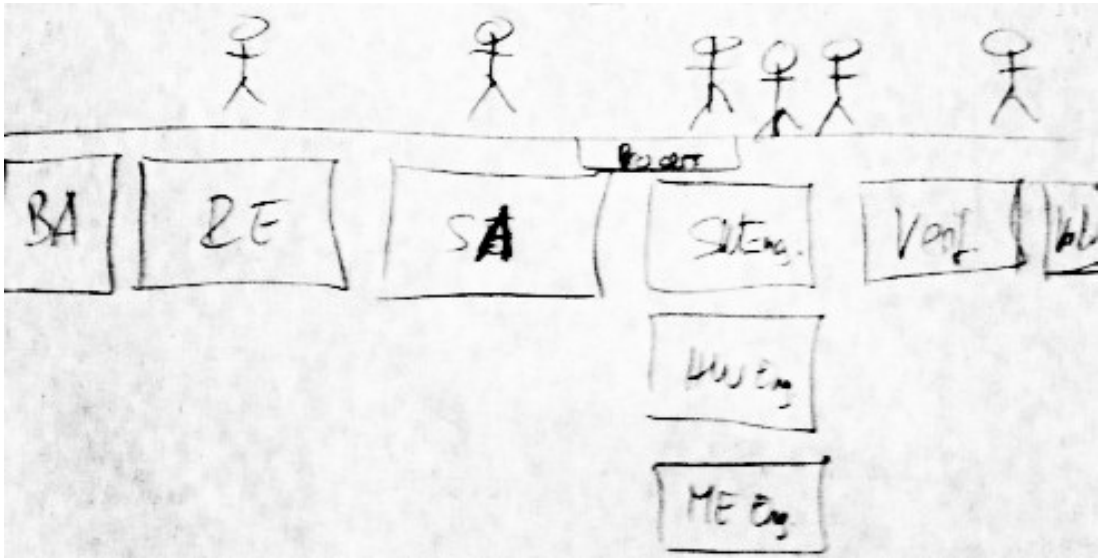

Figure 3: Phases of high relevance for different Systems Engineering processes across the timeline of a project (simplified).

Stakeholders

Many stakeholders have to be served in many different languages. The translation or transformation between the languages is a hassle we have to overcome. Stakeholders also create different structures. Some of the authors have observed that stakeholders blame formalities of languages (e.g. in the case of using UML or SysML) if they are addressed in the wrong language.

An Architecture framework may help to guide how to provide what, but there are concerns regarding the size of typical architecture frameworks like DoDAF. A way out may be the lightweight architecture framework [1].

Stakeholders we like to serve with information have too much information (overflow of information). They receive emails with data they do not need, partly from human origin and partly from automatic systems that trigger notifications by email. This brings us to the next topic, tools.

Tools

There seem to be too many tools involved in Systems Engineering (Figure 4), and "a fool with a tool makes the disaster faster" (quoted from unknown source). Web-based solutions were discussed as a

solution for providing easier initial access to the systems engineering information in the suitable language for the stakeholders. They have the advantage of taking the look and feel of being one single tool even if the IT landscape behind them is heterogeneous.

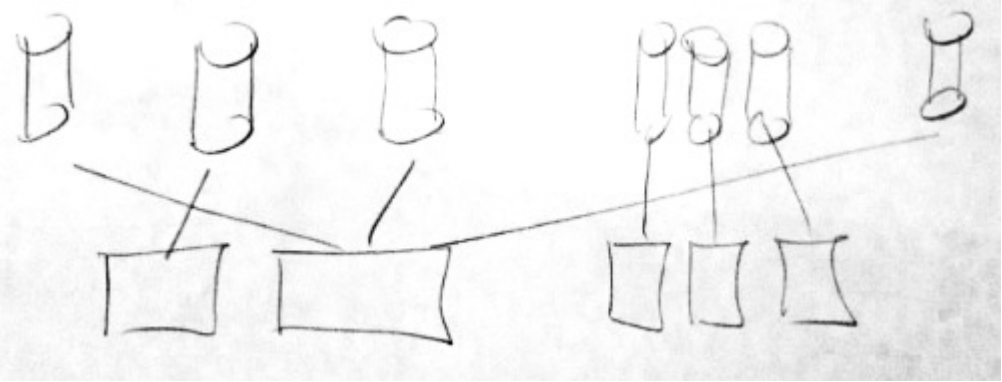

Figure 4: Different repositories (top) for different departments (bottom)

Model-Based Systems Engineering (MBSE)

The question whether MBSE is an answer to all the problems can of course not be answered with a clear yes, at least as of today. We have already discussed that stakeholders need to be addressed in their respective language and this may be the key challenge to master, no matter whether the MBSE frameworks are ready for it or not.

Other Topics

Questions from the “Method” section that were not explicitly answered above could not be answered during the 2 hour time frame that was used for the work. They may be more hard to answer.

Conclusion & Outlook

Within 2 hours, a few relevant future aspects of systems engineering could be found. For example the need to translate to the languages of different stakeholders was identified as a key success criterion for Systems Engineering.

The unanswered questions should be subject for further investigation. Especially “How will a stakeholder quickly find the relevant information?” is a key question to answer.

References

[1] Harold 'Bud' Lawson, A Journey Through the Systems Landscape, College Publications, 2010