Architecture Definition vs. Design Definition

Outcome of the SSSE Round Table

in Systems Engineering on 2018-10-24

Version 1.0

Sarah Bostock, Werner Gerritsen, Andreas Köhli, Thomas Leuzinger,

Andreas Mettauer, Frauke Regge, Jörg Sekler, Julian Yeandel,

Members of the Swiss chapter of INCOSE

Members of the "Systems Engineering Bern" Meetup

Contact: Sarah Bostock, Sarah@ssse.ch

Abstract: The outdated standard ISO 15288:2008 did not distinguish the Design Definition process from the Architecture Definition process. The newer edition of the standard, ISO15288:2015, however makes this distinction. The boundaries between Architecture Definition and Design definition remain unsharp. What are the criteria for distinguishing the two processes? This was the topic of the Bern round table in Systems Engineering on October 24th, 2018 and the outcome is presented in this paper.

Introduction

The attendees were presented with the above abstract, together with the SeBok definitions of both the System Architecture and Design Definition processes. The authors then split into two discussion groups focusing on the following questions:

1. What are the criteria for distinguishing the two processes?
2. Do you distinguish between design and architecture processes in your company and if so, how? What are examples of architecture or design activities and how do each relate back to requirements engineering?

The majority of the subsequent discussions focussed on question 1.

Criteria for distinguishing between the two processes

Both groups concluded that there is no definitive answer to the first question. Group 1 stated that "it depends" and can and should be interpreted differently, depending on the individual organisation and its needs. While they felt it was not possible to identify a clear and universally applicable line between architecture and design, they did propose factors that could affect where the line might choose to be drawn.

These were:

- System complexity
- Internal development of the system / subsystem versus external development (sub-contractors).
- Organisational structure
- Abstraction levels – system, subsystem etc.

Group 2 agreed that there is "no black and white" in terms of where to make a split, and that both system architecture and design must be considered together. They also asserted that critical system features should be taken into consideration during the architecture definition process.

Group 2 also looked at the ownership of the processes, as a potential differentiating factor, and said that:

- System Architecture is the responsibility of a system architect (a more systems-engineering role),
- While the Design Definition would fall under the remit of a (design-) engineer (a more technical implementation role).

SeBok definition

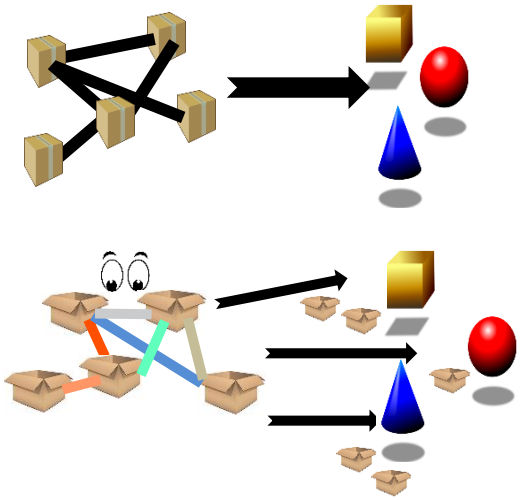
There was general agreement with the SeBok definition that Architecture is more abstract, while Design is more detailed.

One participant said that the way he visualises this difference is as follows:

- He sees system architecture as boxes and the connections between them.

  The boxes in this case are the functional or physical elements, while the connections would be the allocation of functions to physical elements and interfaces between functions or between physical elements. In terms of interfaces at architecture level he would include only **whether** an interface exists and **not** what exactly it does.

- System Design on the other hand would involve looking inside the boxes describing in detail what happens inside them, which bit from each box is done by which physical element and describing as well as what the interfaces carry and how they do so.



This leads to another proposed indicator: that because it is more abstract, architecture is more likely to be the same or similar between devices with many common use cases, while the design is likely to be more different.

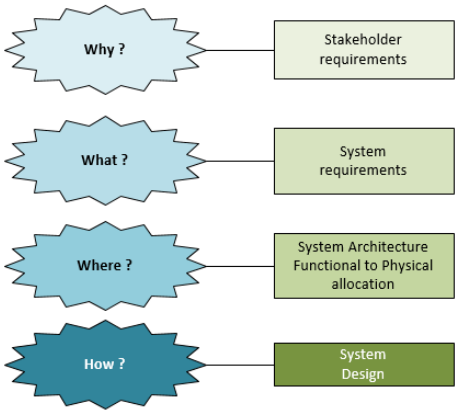
The issue of Abstraction Levels

In one group the question came up as to where to document design and architecture and to what level of detail across multiple abstraction levels. The consensus was that this depends on whether subsystems need complete and independent documentation, for instance for regulatory reasons, or whether the subsystems only make sense as part of a larger system context. The issue of whether subsystems are internally developed or subcontracted out should also be taken into consideration.

There was general agreement that it is more important to write the design decisions and details down somewhere, and move them later if necessary, than to worry too much about exactly what level they belong on at the beginning.

The iterative nature of development was discussed in terms of systems and subsystems, and it was hypothesised that at each level of abstraction we go through the following four questions:

- "**Why** is this system needed?"
- "**What** should the system do?"
- "**Where** should the system do the **what**?"
- "**How** should the system do the **what**?"



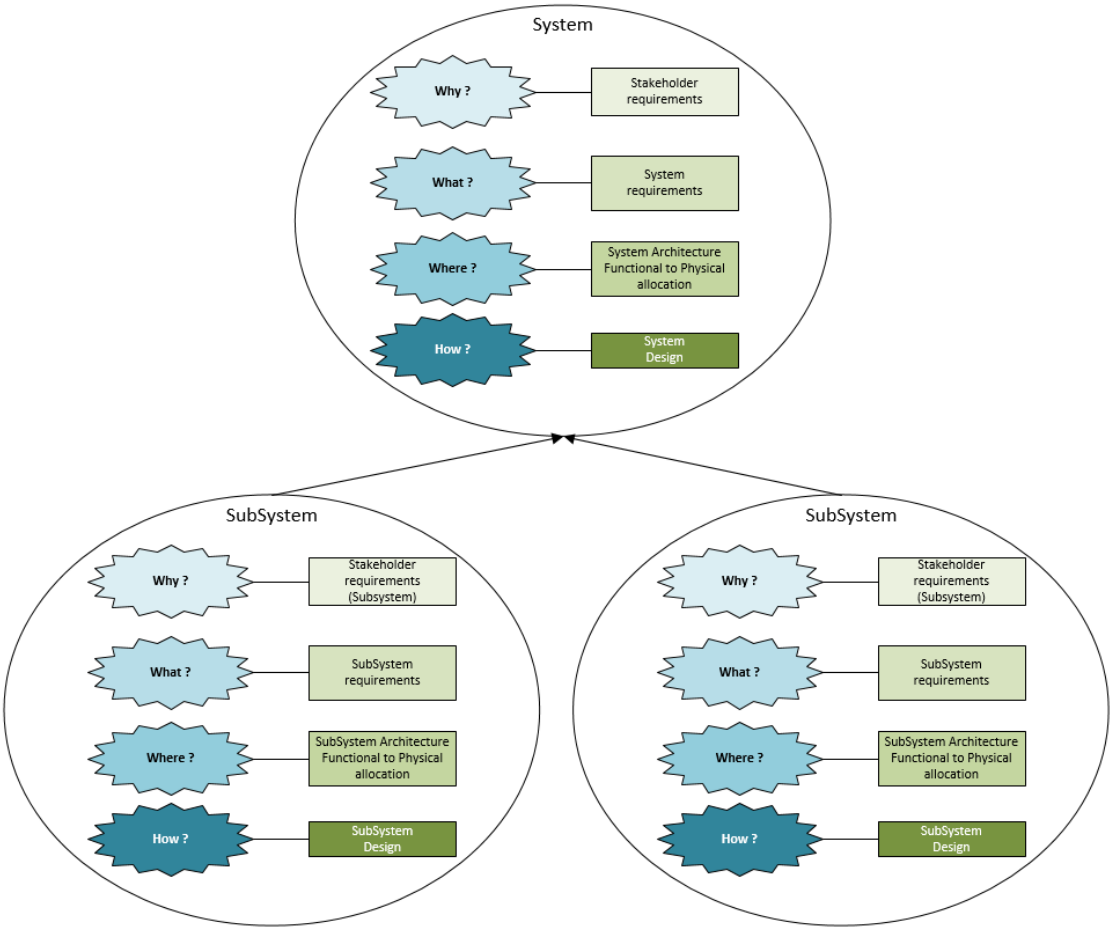
Then, armed with:

what the system should do (System requirements) together with
where it will do it (System Architecture including Functional to physical architecture mapping)
and **how** it should do it (System Design)
then we can proceed iteratively to the next abstraction level.

The diagram to the right shows this suggested mapping, giving another hint as to how some participants would distinguish between the system architecture and design definition processes.

Following this theory through results in an iterative thought process, within each abstraction level, starting each level with "Why?"

In the diagram the green boxes indicate the type of content that the questions might generate. It is of course not necessarily the case that every organisation would create formal artefacts for each level, but the proposal is that at each system abstraction level we should ask ourselves all four questions.



Examples of architecture or design activities within organisations

In response to question 2, some participants gave System FMEAs as an example of a system architecture activity in their organisation, while the Design FMEA would be an example of a design definition activity.

For another organisation architecture activities include SysML modelling of functions and interfaces, while system design documentation describes how each function works and is implemented.

The question of how much detail to include at different abstraction levels was raised as a current concern for some participants. In general the boundary between design definition and architecture seems to not be clearly defined, even in organisations with a long history of both architecture and design activities.

Trace to requirements was made via functional architecture in one organisation, while in another the requirements trace directly upwards from one abstraction level to the next, and architecture and design is used in a kind of zigzag model as a means to extract and refine the requirements for the next level down, or for implementation at component level.

Summary

After a long, enthusiastic and very interesting discussion both groups stood by their initial assessments that that there is no black and white distinction between the two processes that is suitable for all situations.

The participants identified some factors that be taken into consideration to help an organisation find the level that is right for them.

- System complexity
- Internal development of the system / subsystem versus external development
- Organisational structure
- Abstraction levels
- Architecture is more likely to be the same or similar between devices with many common use cases, while the design is likely to be more different.

And the most concrete suggestion was that we should ask ourselves the four questions **Why, What, Where** and **How**, and that:

"Where should the system do the **what**?" could be considered system architecture.
And **"How** should the system do the **what**?" could be thought of as design.