SYSTEM REQUIREMENTS DEFINED BY CASCADES OF CREATIVITY

With more than 30 years of experience with some of the top names in the Netherlands' high-tech industry, Cees Michielsen reflects on his lessons learned and how he tries to relay this knowledge as the instructor of the "System requirements engineering improvement" training at High Tech Institute.

Collin Arocho

It was 1986 when Cees Michielsen got his start in the world of high tech. At the time, he joined the Philips EMT team, which would later become Assembleon and finally Kulicke & Soffa, to help build SMD placement robots. "Back then, our main customers were automotive companies like Ford, GM and Chrysler. We were completely self-contained and had all the essential disciplines and competencies in our business unit," Michielsen recalls.

When he entered the team, Michielsen's focus was on technical informatics, but early on, the trajectory of his career took a detour. "It was there at Philips that I started to develop into a systems thinker, and really got away from my own software discipline," expresses Michielsen. "In hindsight, I can say that was the best start for me in my career; the experience gave me an enormous head start and is why I'm still so passionate about it today."

Now, after more than three decades in the industry, Michielsen is spending his days as a requirements engineer at ASML, as well as an instructor at High Tech Institute where he shares his knowledge, and his many lessons learned, with the next generation of engineers in the "Systems requirements engineering improvement" training.

Credit: Cees Michielsen

Abstraction layers

In systems requirements engineering, especially at the system level, scoping the problem is the name of the game. It's about determining exactly what functions the system should have, the specific properties that are tied to those functions and accurately defining the problem being solved. "If we're, for instance, talking about projecting patterns on wafers, you can imagine that's the main function of the system, and several companies might be doing the same thing. But it's the properties of this function that distinguish one group from its competitors – the accu-

racy, yield, speed and reliability," highlights Michielsen.

For Michielsen, it's these characteristics that make all the difference in the world, and requirements engineering is the art of identifying the right functions and quantifying their properties to define the problem. "Once the problem is well-defined, finding the solution is much easier," Michielsen points out. "But you're not going to find the implementation of your solution straightaway, so you're probably going to go through a number of abstraction or decomposition layers."

Cascade

During his training session, Michielsen explains that, in a system, the highest layer of abstraction is the level with the most general requirements, ie the system needs to be fast or have a certain look. But as you go down deeper into the system, it gets much more detailed. Suddenly, the layers are referring to different subjects or using different languages to express the requirements, which can be a little tricky for engineers to keep the information flowing.

"That's the real objective of requirements engineering, finding different ways to ensure that the data continues to cascade from top to bottom and from stakeholder needs to implementation, all without losing any informa-

tion," suggests Michielsen. "I think if I were to summarize the challenge for requirements engineering, I would say that it lies mainly in the cascading of information throughout each abstraction or decomposition layer."

Quantification

According to Michielsen, one very important part of the method is to find the complete set of requirements for a system. "The question quickly becomes, 'when is the set complete?'", he poses. "The best approach we've seen so far can be expressed using an equation, which we share in the training. It allows us to fully define a system by its functions, properties and constraints, and can be applied from the highest levels to the components and parts at the lowest points." He continues, "By specifying and quantifying these criteria, the true requirements can be derived. This is one of the main steps of the training, learning how to put a value on each of the properties of the system."

"Once the goals are defined, we can identify solutions – design options – based on assumed capabilities of subsystems. This is where creativity leads the product development process, as many different options are considered for solving the problem," Michielsen depicts. "As long as we document the assumptions that are made during that creative design process, we can later translate these assumptions into requirements for the lower-level subsystems that we need in the solution."

Justification

To Michielsen, this is one of the most powerful elements of the whole method. The ability to see the complete line of logic from a quantified system definition to the design decisions and finally to the specific implementation of a solution. That is, if

engineers are able to maintain coherence between system requirements, system design and system decisions – a crucial factor.

"As long as the information feeds properly, we can derive requirements for the next layer and continue the cascade. That way we can ensure that whatever requirements we end up with at the lowest component level, through our method and our traceability, we can exactly come to the justification of each requirement and each decision made throughout each layer. That's the whole essence of the method."

After more than 30 years in the industry, what do you most want to share in your trainings?

"As a trainer, I want to help instill confidence in the process. Following the method is one way to achieve that, because the students get the feeling that the system can be complete, consistent and correct – in terms of specifications. That can really help it feel less daunting. Once you cross that hurdle, the students can almost immediately start determining the main functions of the system and decide what properties are related and which constraints apply at that level. By quantifying these aspects, they don't just state that the system should be reliable, they say explicitly just how reliable the system should be."

Lessons Learned

With his 30+ years in process architecting, Michielsen has developed several practical methods to keep the information flowing from layer to layer. His success in the field opened the door for him to work with top Dutch and European companies, like Prorail, Eurocontrol, Punch Powertrain and Vanderlande – and several others, to help establish and implement

processes for their own requirements engineering programs. "What I found was that there are enormous differences between each company, especially in implementation," recollects Michielsen. "When I went to work for DAF, we put in place a complete requirements engineering process in three years' time. We could successfully train hundreds of engineers and the method was paying off."

Noting the success of the DAF project, Siemens called to lure Michielsen to Germany to help establish the same approach for Daimler. "It was a huge step for me to be invited to implement the system, but it quickly became clear that the approach we developed at DAF wasn't going to be transferrable to Daimler," Michielsen calls to mind. "Daimler was just organized in a completely different manner, with responsibilities being spread among departments and people in a way that made successful execution really difficult. The inability to get something going there was disappointing," he says, continuing, "It certainly was a big learning experience for me, and it came with a lot of tough lessons learned."

Are these lessons learned what drives you in this domain?

"In part, yes. I have an enormous passion for this whole process. I want to help improve product capabilities and product manufacturing capabilities, especially in the area where I live and work. I want to make an impact on industry in that sense because we've learned so much and I want to spread this information," emphasizes Michielsen. "It's not all my doing, it's all the companies I've worked for and all my experiences. I'm extremely grateful for being able to do that, and I'd like to spread that knowledge to make sure that the entire ecosystem can benefit, and we grow from it." ⏻