SIEMENS

Future Combat Air System
Stopover : Product Definition

AN INTERVIEW BETWEEN EXPERTS

Click here to listen to the podcast now !

Allan Behrens
Hello. My name is Allan Berens. I'm from Taxal. We are industry analysts looking at the linkage between technology and manufacturing. I'm now joined by Jon Field, who is here to discuss product definition with me. Jonn, what's your particular interest in this domain?

Jon Field
Well, my background comes from both manufacturing and product lifecycle management, but over the last 15 years or so, I've been increasingly focused on the systems engineering side.

A.B. : Jon, we're going to be talking about product definition. What exactly is product definition?

J.F. : Well, product definition is the scope and the performance requirements driven by the voice of the customer. That's the Cimdata definition. So that's the really simple kind of nugget of it, right?

A.B. : In a bit more depth. What does it matter to those who may be listening?

J.F. : Well, I mean, we're talking today about product definition for sixth generation future combat air systems. So if we look at how fast jet aircraft have developed from the Lightning of the 1950s through the generations, we've also seen the evolution of engineering computing running in parallel and enabling those fast jets. We've seen significant changes since the Typhoon and even the F 35 Lightning II in the way that we're supporting the companies involved in these technologies.

A.B. : Right, so the words “product definition” - just expand on that. What sort of area of the design through manufacturer phase are we talking about? When we talk about the product definition.

J.F. : We're talking about something that occurs very early in the concept phases of these complex systems programmes. In the past, we've seen a lot of challenges of communication in systems engineering through language and through silos in information. So the product definition has evolved and become far more connected to the lifecycles of concurrent engineering within those communities, the Communities of Practise that make up the partner and supply chain. So while once engineering was very much based around 2D drawings, for example, that evolved to parametric mechanical computer aided design, we now see product definition evolving from textual requirements, the musts, the goods and the shoulds, to a more functional definition involving simple two word kind of verb noun descriptions with interrelationships and parameters. So this helps people to communicate much better. For example, where European teams of experts get together to collaborate around diagrams, we're reducing the dependency on text and language, and that's a good thing. Avoiding the differences in interpretation, avoiding the misunderstandings.

A.B. : And this is all part of what we term as model based, isn't it? It's using diagrammatic and algorithmic methods to try and clarify what people understand within the various phases of the design and manufacturer of the aircraft.

J.F. : Yes, absolutely. I mean, model based is really the way that we verify in a recursive way, going down gradually into more and more depth of decomposition, starting from the very abstract down to the very specific, the concurrent engineering aspects of it means shifting those engineering activities left to understand the constraints. So particularly the 50/30. So providing a programme that delivers in 50% of the time and at 30% of the cost. So that manufacturing as much a part of that product definition. So, yes, model based is a concept, but we're connecting it very much to all of the downstream activities all the way through into manufacturing, in service logistics and so on, throughout the digital thread.

A.B. : And why does this matter? Why does it matter to those who might be listening?

J.F. : Well, key technologies need to be demonstrated very early in the programme. We need to understand technology readiness level in all of the areas. So product definition is about connectedness along the whole ecosystem of the programme, avoiding those silos, avoiding the late changes. And those late changes can result from lack of communication from systems, from systems people to design people, manufacturing people and so on.

A.B. : Right, and cost of inaction is what are some of the benefits and how might people see benefits from it?

J.F. : Well, cost of inaction is really the fact that we're needing to work so much faster than we ever have before. These systems of systems are much more complex. The expectations also of graduate engineers coming into the industry have been set at much higher levels of interaction and connectedness and the ability to move down into specifics in a very fluid way. So compared to the previous programmes, where we focused on configuration management and evidence that everything has been done safely and correctly, we're now in the concept design phase and we're providing much lighter weight review and release processes that look a lot more like simple project task management in a software world than the engineering change processes that are needed at the high levels of maturity.

A.B. : Right, and this affects who in the audience should be particularly keen to listen and take action on this?

J.F. : Well, I think everyone involved in the whole engineering process. I mean, we're not talking about some specific narrow domain here of systems engineering, model based systems engineering, we are just talking about the whole world of engineering. This is what engineering is in the 21st century.

A.B. : Right, and what about addressing the issues that you've noticed within your clients and you've worked on in the past? How does one go about changing both behaviour and capability?

J.F. : Well, it's well known that the costs of change start to increase as we increase in maturity. So we're focused on getting everyone talking to each other in the early stages to get it right sooner in the lifecycle. So we talked about avoiding silos, we're pulling people in at the concept of operations level so we can link to models, requirements and parameters that they're driving from the conops. Without saying that anyone is guilty of doing bad engineering, we do encounter companies that have identified less than ideal practises in their own projects, so that can be related to the system space wherever integrative activities need to take place. And we're finding emergent behaviours early in that shift left that we need to do. So, just to summarise, engineering is an activity of people, processes and tools. And there's a lot of tools, but the people and the processes are very much equally important.

A.B. : So the implementation, going back to your comment, the implementation is people, processing, tools. What are the sort of practical ramifications of implementation? How does a customer get going? How do they start?

J.F. : Well, there's a lot of shadow IT there's a lot of spreadsheets, databases, a lot of code being written, the programme's relying on all of this, but it's very difficult to make that better managed to reflect increasing maturity and the increased dependency. There's a lot of things hanging on all of those things and so we need to get that better managed. And we found that that's exactly the focus that our customers have been taking, to bring this all more under one roof and avoid the business risks, really, of having people going off and doing their own thing.

A.B. : And it's all the more important with such complex projects such as, obviously the aircraft systems we're talking about.

J.F. : Well, yes, absolutely. The benefits of running these programmes in a way that reflects their increasing software content are really significant. We've seen a proliferation of software tools, methodologies, they're all good in their own way. What we're helping the future combat Air community with now is integrating software configuration management with electronics, with the hardware, the mechanical domains. There's only one overall system, but as the functions are allocated to those domains, there tends to be a different suite of tool sets involved. At the same time, customers are asking us to remain open to the tools that they're using now. We're not forcing anyone to change toolsets where it makes very little sense.

A.B. : Right. Interesting. What about success? What does success look like within these types of programmes? Specifically in the area of product definition.

J.F. : Success is really the extent to which people are able to collaborate. None of this is done by one company anymore, so there's a huge focus on collaboration. We're being asked to facilitate this in a standards-based way. There's still a tendency for tool sets to only interoperate with themselves in terms of the slight implementation differences in the interchange standards that then result in nugatory work or even prevent information being shared at all. So that's on top of all the difficulties of being in classified environments, protecting IP, finding people that can work in those environments. So, we're sharing the work that we're doing with the standard bodies early with the communities, to help the community decide on strategies for collaboration and for data exchange that will have to prove the test of time for the next 40, maybe 60 years. Specific area here is the systems modelling language V2, which means we're supporting SysMLv2. So we're supporting existing versions now, but we're looking ahead to better collaboration and wider usage. That means companies joining that Community of Practise can do so now and then gradually take on the new standards as it becomes more practical and necessary to do so.

A.B. : Right, and if you were to summarise, what should people think about next? What's the action plan for companies really looking at product definition as a key focus area?

J.F. : Well, I think one of the things that we've referred to in other podcasts is the Maturity Review. So taking a really honest look at yourself and are you doing this in a joined-up way? We've talked about Silos, we talked about collaboration. This issue, it affects everyone in the systems engineering process, systems engineers, requirements, engineers, specific disciplines, safety, electronics firmware, actual writing of software as well as the mechanical domains, not to mention projects and programme managers. Managers are responsible for these groups of people who are having a significant impact on the programme. So bringing everyone together to really do that Maturity Review in a connected way is something that we've done a lot of work achieving that and sort of acting as the glue between the various disciplines that have tended to work in isolation, which I'm sure everyone has seen that a lot.

A.B. : Absolutely. And experience counts, doesn't it? Experience counts.

J.F. : Yes.

A.B. : Anyway, thank you very much for that, Jon. That was very interesting. Appreciate your time.

J.F. : Thank you.