DE Desktop Engineering®

Check It Out

September 18, 2012

Managing Product Complexity with Systems Engineering

Managing Product Complexity with Systems Engineering

Dear *Desktop Engineering* Reader:

I'm awed by ballet. It's an extraordinarily complex blend of components -- rigidly specified movements, athleticism, storytelling, and music. Three sets of disciplines -- dancers, musicians, and set crew -- as well as other pros need to collaborate, combine, and execute their individual tasks in total harmony to achieve bravo performance. But add a new dance number the night of the show and you get a middle school dance recital instead of beauty. It's a lot like product development, isn't it?

"Yeah, yeah. Dudes in tights and product development," you snort. Go ahead. Jeer. But the embedded systems and electro-mechanical products you create are extremely complex fusions of hardware and software. They're so engineering- and software-intensive that they are beyond traditional mechatronics definitions. And your schedule is tighter than a leotard, and everyone's task requirements are minutely choreographed. And you've got to make what you engineer work flawlessly with what someone else does.

And yet, amazingly, the software that is your market differentiator often is coded in an isolated workflow then united with the hardware late in the development cycle. This approach spawns development miscues -- delays, rework, scrap, and recalls. It's the cousin cycle to the olden days when you tossed a design over the wall to the analysts to see if you're on the right track.

Today's Check It Out is a trifecta -- two reads and a video all from one link -- addressing the challenge that systems engineering poses to organizations. We're not talking about PLM (product lifecycle management). Rather, it's about tools to control, manage, and integrate your systems software development lifecycle within your PLM environment. The toolset that frames the discussion is PTC's Integrity software system lifecycle management solution. The thesis merits your consideration. Here's why.

The essence of the proposition is that engineering organizations must modernize their approach to the development of software-intensive, software-differentiated products to remain competitive and innovative. The suggested solution is to re-engineer your product development to promote a multi-disciplinary and collaborative systems engineering process. Here, the focus is on managing requirements -- say, user requests, changes, and configurations -- and on tightly integrating workflows across mechanical design, electrical design, and software engineering so that interdisciplinary collaboration is early, often, and standard operating procedure.

Requirements, workflows, and engineering disciplines are interdependent. So controlling, managing, and orchestrating their activities to foster collaboration and realize requirements ultimately gets products to market faster and less expensively. And those products are validated as built to the specified requirements. As a bonus, you also achieve real-time metrics abilities, auditability, a smoother path to design reuse, and process, communications, and organizational consistency.

Frankly, today's Check It Out talks about a lot more than that. There's much to say about managing the complexity of the multipart systems engineering processes that go into your complex products. PTC has done a good job of making its case. Hit the link over there and see what I mean. It won't take you long to read and watch it all, but you'll be thinking about it for a long time to come.

Thanks, Pal. -- Lockwood
Anthony J. Lockwood
Editor at Large, *Desktop Engineering*

For more information on this topic, please visit www.deskeng.com