A BRIEF HISTORY OF CONFIGURATION MANAGEMENT SOFTWARE


This month’s CMsights looks back at the history of Configuration Management software to understand the impact of the movement toward PLM being delivered as an open federated platform of interoperating applications. The evolution away from huge monolithic enterprise PLM solutions – promising everything for everyone – offers a strategy to prevent the all-important “digital thread” from becoming a digital trap for those already struggling with increasing product requirements and system configuration complexity. This has substantial implications to aerospace and defense programs and their project managers.

IN THE BEGINNING

Special-purpose software tools supporting the needs of product engineers and configuration data managers have been around for decades. In the ‘70s long before CAD, PDM, and PLM software was widely available, home-grown tools were developed by engineers to manage increasingly complex product configurations they designed, manufactured, procured or serviced. Customized FORTRAN code created by all-knowing data manager experts ran these databases on mainframe computers.

Starting in the ‘80s many of these legacy codes would be ported to client-server architectures as mainframe computing faded away. As in the early days of BOM management, Lotus 1-2-3, dBase, Microsoft Access and Microsoft Excel utilities would next be used to manage product configurations and all their variants. These sacred tools were meticulously cared for by CM experts, and even though occasionally cursed by others who had to use them replacing this software was met with resistance.

CM MODULES IN ENGINEERING-CENTRIC PDM

An impressive succession of Product Data Management (PDM) software appeared commercially during the late ‘80s and into the ‘90s. Solution providers rapidly added capabilities beyond those needed for basic CAD file management including secure vaulting, document management, BOM management, visualization, CAM integration, user access administration, workflow automation, and version control. However, because these PDM solutions were developed primarily for the product engineering department, they struggled with delivering full-lifecycle change control, much less true configuration management.

An inconvenient truth engineering-centric PDM providers could not circumvent was that most users of CM tools and CM data were not in product engineering. Instead, they are found in downstream functions like procurement, logistics, test, quality, supply chain, compliance, tech pubs, and aftermarket service. These users have very different needs – as well as levels of training and tolerance – for CM software, data, and workflows than those of users found in engineering. If the only routine users of configuration data ended up being found in engineering, it was doubtful that there was real configuration control being performed across the entire organization.
CUSTOM CM TOOLS

The growing use of engineering PDM software with a proliferation of digital product data had the effect of inciting the need for more, not less, capable CM tools. The next generation of CM software would need to fully embrace the five guiding principles of CM.

These foundational capabilities are:
- Configuration Planning
- Configuration Identification
- Configuration Change Control
- Configuration Status Accounting
- Configuration Traceability and Audit

To obtain these must-have CM capabilities, industries like aerospace and defense invested millions of dollars on proprietary software. It was not uncommon that these one-off applications were utilized by a single program spread across dozens of contractors working on long-life products, whether a complete weapons system or support test equipment in the field.

Ultimately, the resource drain of maintaining legacy CM solutions became prohibitive. If these codes didn’t die a natural death they were killed off when the original software developers retired, taking the knowledge to maintain fragile code with them.

EMERGENCE OF COTS CM SOFTWARE

The cost of extending PDM software or developing custom code to support a single CM-intensive project was rarely financially sustainable. Besides development costs, the expense of training staff to use and maintain these solutions siphoned off valuable resources better utilized across an A&D program.

During the late ‘80s and ‘90s commercial-off-the-shelf (COTS) CM software emerged from industry-focused solution providers. This included CMstat which in 1989, offered the first client-server CM software in the market. The adoption of COTS software was accelerated by the migration from mainframes to client-server architectures and eventually web-based applications.

Over the history of configuration management this next generation of CM software – finally under the control of user departments – found quick acceptance due to the sharp focus on addressing

THE PROMISE OF BIG PLM

During the late ‘90s and into the ‘00s the capabilities of PDM software expanded beyond simply meeting product engineering needs. Many grew into or were rebranded as PLM-enabling solutions capable of supporting the entire enterprise and its extended supply chain. New software modules were either developed in-house or acquired, then integrated to provide digital support of the full product lifecycle.

These new PLM applications included project management, portfolio management, requirements planning, cost management, model-based systems engineering, digital manufacturing, simulation, process optimization, and of course configuration management. All together they promised to deliver a single version of the truth that would collapse organizational silos and transform the enterprise. With such a grandiose sales pitch it was thought that if PDM had not sounded the death knell for specialized CM software then enterprise PLM surely would finish the job. This would not be the case.
THE REALITY OF ENTERPRISE PLM

It is no secret that PLM solutions were often sold based on their promise to provide full-lifecycle change control and systems-level CM across all functions of the enterprise for the OEM and their supply and service chain partners. The appeal of this sales stick was financial; the cost and liability to the corporation from product failures or disasters due to a lack of effective change control was already a chief concern of the executive suite. The sales carrot was the imaginary ROI projected once full-lifecycle, system-level configuration control was in effect for the OEM and supply chain.

Less widely known is that for many PLM deployments, millions of budget dollars and months of calendar time were exhausted before reaching the point in the deployment road map where CM could be implemented. It was not uncommon that before the CM stage gate was reached in the schedule, customer requirements, budget allocations, management priorities, or executive sponsors would change. Or if not these disruptions within the customer’s organization, then the PLM solution provider, their software products or system integrators had been changed, acquired, merged, replaced, or obsoleted. Worse yet for users who just had a job to do was when solutions were “reimagined” halfway through a deployment with the promise (or threat) of “transforming” their workflow processes.

Many project managers were silently thankful for all this as it avoided anyone being blamed for enterprise PLM deployment failures that were over budget, over schedule, over weight, and woefully underwhelming. Regrettably, users once again had to settle for basic change control instead of comprehensive configuration management.

WHAT COMES NEXT

Admittedly, during the over-hyped era of big-bang PLM there would be little oxygen left in conference rooms to consider best-in-class or industry-focused software applications of any type, including CM. However, there were still numerous occasions where configuration managers were asked to evaluate and compare the different CM strategies and CM capabilities between engineering PDM, enterprise PLM, and industry-focused COTS CM applications.

More often than not, the latter CM solution strategy would be ranked as the most capable, deployable, usable, maintainable, and affordable solution for the project at hand. Yet, the program office was often forced to swallow over-weight PLM solutions into their project plan due to the other notorious “P” in PLM, “politics”. And admittedly, there were few other strategy options at the time for them to get what they really wanted in CM all while supporting corporate mandates from above. That is until now.

In Part 2 of this CMsights series we will look at the future of CM software by examining the emerging strategy of “Platform PLM” where functional services like CM are delivered via an open, federated architecture comprised of rapidly-deployable industry-configured applications. These best-in-class applications interoperate to create a more robust, sustainable, and resilient architectural vision of PLM that can last into the future without constant tearing up or ripping out of whole PLM solution stacks. We’ll end by summarizing what this means to the A&D program office with recommendations for planning forward so that the digital thread opportunity does not become a digital trap nightmare.

Until then, what do you remember from your own career experience about the history of configuration management software? Share it by sending us a note to information@cmstat.com or visiting www.cmstat.com to leave a comment.