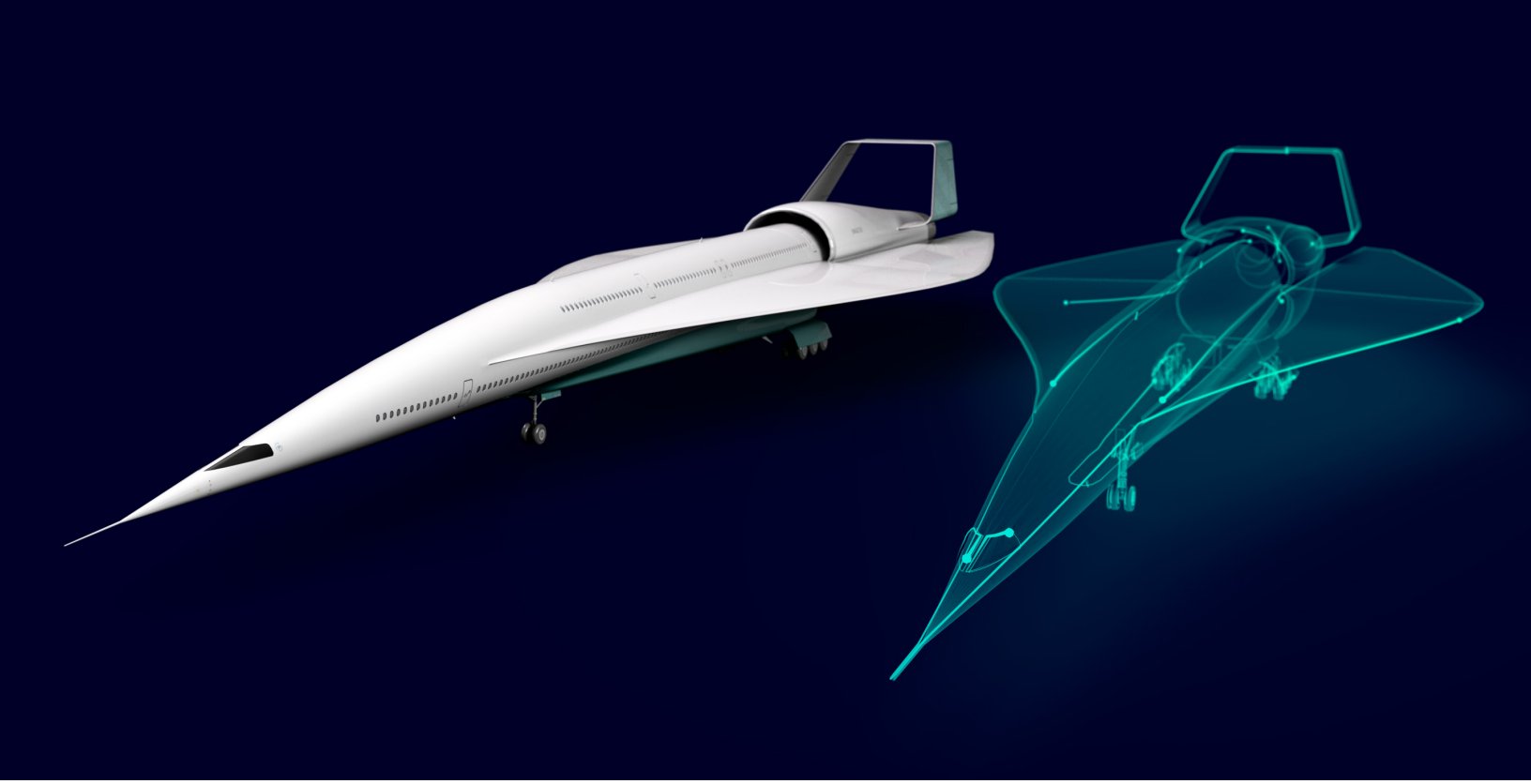


DIGITAL INDUSTRIES SOFTWARE

Automating verification, validation, and compliance for aerospace software systems

Accelerating functional safety compliance with DO178C, DO254 and DO278

Executive summary
There's no questioning that safety and performance are paramount in the aerospace and defense industry. Verification and certification processes are critical throughout the full aircraft system lifecycle. Companies today are required to quickly automate verification, validation and compliance while also delivering products on time and within budget. To make this possible, organizations need to keep all elements for the proof of compliance in context of the digital twin. This is true whether these elements are based on comparison or similarities with the previous programs, virtual test-based or physical test-based.

The aerospace industry is continually challenged with proof of compliance, especially as it relates to software. This white paper outlines the need to drive and foster synergies across disparate development teams and empower all stakeholders with the capabilities and information they need to excel in the highly complex aerospace ecosystem.

SIEMENS

Introduction

Software problems emerging in both commercial and military aircraft have plagued the airworthiness process, delaying projects, extending completion dates and dramatically raising costs. As reported by the Government Accounting Office (GAO) in 2021, setbacks to the F35 fighter program relating to the final software (Block IV) capability problems alone have increased costs more than 40% and pushed deliveries to 2029. The project delivery has been delayed more than eight years and is $165 billion over original budget. Almost all of the newest technology is dependent upon software and its relationship with the control systems that are crucial to airworthiness.

In the commercial aerospace sector, Boeing is still feeling the effects of the 737 Max software solution to solve the altitude correction needed for the larger engines. In addition to the three major flaws in the software, the problem was also exacerbated by a failure to completely test the interaction of the software with flight control systems. It was reported in 2020 that this software-based problem was likely to exceed $18 billion in costs to Boeing. Although the problem did not lie completely within the discipline of software development, it exposed disconnects in communication between engineering teams in their attempt to solve a cross functional technical problem. In addition, a compressed quality control process overlooked important aspects of verification and validation of the software operating within the technical ecosystem. As a result, Boeing faces a much higher level of scrutiny and FAA oversight in approving engineering data and performing quality inspections, which limits their ability to use designees, and subsequently their entire system of airworthiness approvals.

Software in aircraft represents the most impactful source of innovation and has also become the largest area of challenges and complexity. Startups in the autonomous, VTOL and urban mobility spaces have exploded in the past decade. These organizations, as well as traditional aerospace firms, depend heavily on agile methodologies to speed up time to market within a competitive and highly regulatory environment. Introducing agile methodologies into the compliance process introduces many challenges to the forward motion of a project.

Unlocking the value of these opportunities will require new kinds of thinking and collaboration – including a considerably higher degree of cooperation both among OEMs and suppliers, and between the industry and regulators. Furthermore, labor shortages among skilled software developers will force manufacturers to staff their teams with lesser-quality engineering resources from around the world, another factor contributing to the 737 Max problems. This will in turn require streamlined processes, transparency and real-time information exchange on a global basis to avoid the kind of communication breakdowns that have been leading to a continuous number of defects and a record level of project impacts. As time goes on, functional safety issues that result, in part, from disparate teams working across continents will only become more challenging, and possibly more frequent.

But it's not necessary to look to the future to realize that development organizations need to reinvent themselves and upgrade the digital tool portfolio they use to stay on track. Nearly every day it seems there are stories about yet another software problem. The amount of those problems is not as surprising when we consider the number of software-based functions in modern avionics and functional systems. A large percentage of the innovative features that control an aircraft are software-driven. This includes critical functions, such as flight control, Full Authority Digital Engine Controls (FADEC) and navigation, among others. The complex software development environment has been stretching development teams to the limit. Norms and processes have evolved over the past several years, such as the Radio Technical Commission for Aeronautics (RTCA) functional safety standard DO178C for airborne software systems, as well as DO254 safety standard for electronic hardware and DO278 for ground based systems. These standards, cited by regulations, are intended to ensure safety and avoid the defects that lead to catastrophic events caused by flaws or defects in airborne software.

Aerospace manufacturers are investing an increasing amount of time and money in the development and improvement of their processes and process models to map to these new norms. To support those initiatives, leading organizations are looking for better tools to overcome the limitations of their legacy systems to achieve airworthiness compliance more efficiently. Increasingly, they are discovering that disconnected development systems challenge the ability to adapt to their specific environments while providing built-in expertise to retool quickly for the current challenges and get ready for the road ahead.

When considering the parameters and types of tools to employ that will provide an efficient and valuable development experience, it's important to be aware of the fundamentals that should exist for such a management environment.

Architecture

The architecture should be structured as a singular environment to accommodate the management of all software development functions within the environment, using a universal user interface that any of the cross-functional teams can navigate easily within and with an easy framework of collaboration.

Digitalization

The first step toward digitalization is to convert homogenous content in the form of documents (electronic or paper) into their individual elements (requirements, specific document formatting, unstructured content, metadata, graphics, etc.) so that they can be tracked and managed. Once digitized, this content can then be easily organized, related, controlled, distributed, searched and accessible in several different ways.

Documents

The first concept can be thought of as a container that organizes both unstructured content along with structured project data in a format that looks like a familiar word processor. Creating, defining and editing documents such as requirements, safety goals, risks and test cases should be as easy as using Microsoft® Word.

A familiar and easy-to-use online word processor will enable writing, editing and formatting content easily. A major capability, conceptually, is that some document content should have the ability to be marked as artifacts and classified as essential project components. In this way, one user role can take advantage of workflow and project management, while document authors can continue to work with content from a document perspective. This approach provides the best of both worlds: office document usability and a data-driven process that enables project management for the organization.

Business analysts and requirements engineers who typically work with documents should not have to change their paradigms or sacrifice functionality and ease of use, while technical people can exploit the workflow and management capabilities of the data-driven user interface (UI) option. Executives and others responsible for compliance issues get the reports they need, and the whole organization benefits from improved efficiency, transparency and communication.

Team members should have the ability to easily import and leverage existing assets using a rule-based import method that parses artifacts according to their intended function in the project. Ideally, it should be configured to recognize artifacts like requirements, test cases, defects and other such items contained in Microsoft Word or Excel® and easily import them, providing a quick path to digitalization.

Conversely, exported project data in the form of Microsoft Word or Excel documents for reviews, approvals, and edits should have the ability to be imported back seamlessly and become part of the project content that can be included in the project workflow and fully tracked.

Artifacts

An artifact can be anything to be tracked in the project. This basic concept can be broken down and defined as artifact types for requirements, activities, change requests and test cases. Custom work item types – for work products, safety goals or anything imaginable – could also be defined as required.

• Artifact data fields, custom fields

Each artifact should have available many default data fields used to describe and categorize the item, assign it to a user, incorporate it into project planning and tracking, set its status and so forth. Custom fields should have the ability to be defined for any artifact type, enabling tracking of and querying on, any kind of information. The Design Assurance Level and Objective of a requirement can be managed and tracked using a pre-defined work item custom field to show compliance in an audit. For each work item, the look and feel is completely customizable.

• Artifact lifecycle and workflow

Each artifact type should have its own lifecycle or workflow definition. A workflow is a set of states, status transitions, transition conditions and dependencies that an artifact passes through in its lifecycle. Each of its elements' status, transitions, conditions and dependencies can be customized, enabling customization of workflow to support any process.

• Links between artifacts; link attributes and roles

Linking artifacts is the key to taking advantage of traceability and impact analysis. Artifacts should be linkable inside one project, between different projects and even between different repositories. This capability enables relating artifacts to different products and/or product variants and to obtain traceability and impact information that is not limited by project scope.

The links between artifacts need to be defined and categorized by link roles. Link roles are distinguished by their names (relates to, implements, verifies, etc.). The roles can have different semantics if needed and can be customized to meet specific needs. Using links, it is easy to manage a complete requirement flow from the concept phase down to hardware and software requirements and related activities, work products, risk items and/or test cases.

Traceability

• Tracking/audit trail

Every artifact change in a project needs to be tracked and reported using an underlying configuration management system. A complete audit trail should always be available (who did what, when, and why). At the core, it should simply not be possible to change anything without leaving a trace, even if an artifact is deleted. All configuration changes should have the ability to be rolled back if needed.

• Traceability and impact analysis

All traceability and impact analyses must be based on the links between artifacts. A variety of different views and reports representing the results of traceability and impact analysis are necessary to provide context to the intended audience. Consider these essential components that enable traceability:

- The current state of every document is always available online.
- User permissions ensure that access is as limited or as open as needed.

- A history of each document is readily available. Each time a document is saved, a new entry or revision is created in the history. It's easy to review any revision, and comparisons of any two revisions serve to understand what changes took place between the older and the newer revisions.
- The document's URL can be shared with other users so they can collaborate on the content.
- A robust mechanism for export and reimport allows for collaboration between external stakeholders who aren't registered users and brings that input into the data flow.

Requirements management

Defining and managing requirements within a singular solution provides significant advantages over legacy approaches:

- Best of both worlds support: Those who are accustomed to a document-centric approach can continue to work with documents, and those who need to work with individual data items can view in lists, in a familiar spreadsheet format.
- Integration of requirements into the overall process: Requirements captured are an integral component of the overall development process from start to finish, so there is no struggle to manage isolated office documents that are decoupled from the processes of implementation and testing.
- Requirements based on standards applicable across projects can be re-used within different projects.
- More efficient and timely collaboration: All stakeholders always have access to the same version of requirements. Edits are reflected in real time – there are no delays waiting for emailed copies. The process is integrated and automated into project workflows so that no steps are missed or skipped due to miscommunication, everyone can see the current status, and everyone is notified automatically as changes take place and requirements move forward in the process. Those responsible for approval and sign-off can do so electronically online.
- Easier and more robust traceability: With other legacy approaches, rigorous and thorough traceability has been difficult. To accommodate functional safety compliance, traceability needs to connect from the highest level artifact to the most granular, in particular a single line of source code for DO178C and DO254. This should be easy to implement and totally transparent to the user.

Test and quality management

Some test management considerations to deliver these capabilities:

- Specify and manage tests using familiar user interfaces and/or with integrated tools
- Easily create traceability down to defects, up to system level requirements
- Manual or automatic test runs; optionally import results from external testing tools
- Automatic test execution history with detailed statistics
- Customizable test runs from ready-made templates

Change and configuration management

For integrated change management:

- Collect, manage and track change requests in one unified solution
- Use impact and traceability analysis to decide which project artifacts must be checked, changed or added
- Link change requests with their related requirements
- Apply suspect management to assist in propagating a change
- Use audit trails (history) for work items and documents to show who changed what and when
- Collaborate via threaded comments, voting and automated notification of implemented changes

In order to track all of these elements to provide the necessary traceability, the underlying data structure should support highly granular and finite traceability. Typically, this should be a system that never deletes any artifact, but rather removes from view those artifacts no longer active. Forensics allow all of this information to be accessed and reported on to cover all activity associated with the artifact.

In such an architecture, changes are automatically versioned on each save. Each revision is always available via history information of an artifact. Differences between artifact revisions, work items and documents can be displayed graphically, enabling visual comparisons of any work items managed in the system, including requirements, change requests, test cases, source code, activities and others.

Audit trail

It should not be possible to change anything in the system without having the change tracked. The complete audit trail always needs to be available (who, when, what, why, etc.). Even if something has been deleted, it needs to be tracked to show complete and total traceability.

Baselines

Baselines are a key functionality used to mark the current state of a project, including all project artifacts, so that team members can check the differences between different baselines or between a baseline and the current state of the project. By selecting a baseline for comparison, it is possible to specify one or more revisions to compare against a single baseline.

Build and release management

Each artifact of the project should have the capability to be linked to releases. Information can be retrieved using Polarion search and reporting features. Benefits include:

- Accelerate compile and error recovery processes with mail notifications on build or test failures
- Quickly track down issues with complete audit trails of builds and releases
- Measure re-use of requirements, test cases and other artifacts across projects

Collaboration

Team collaboration

Stakeholders need to collaborate and communicate easily on various levels. Seamless collaboration accelerates project trajectory and contributes to fostering an Agile methodology and processes. For discussions and collaboration at higher levels, personal and individual interaction is encouraged through digital means both internally and externally to the development environment.

More granular collaboration and communication takes place in comments on individual work items. Discussions on multiple threads can occur among project team members. Comment visibility should be optionally controlled and limited; for example, some comments may be visible only to managers.

Interchange/collaboration between OEMs and suppliers

- **Native integrations with RIF/ReqIF, MATLAB®, and Simulink®**
  An open platform that provides different options for OEMs and suppliers to interchange data, including native Requirements Interchange Format (ReqIF) round-trip and data exchange via the MATLAB/Simulink integration is a critical capability for communicating with other tools and integrating MBSE into the development process. These integrations help optimize reviews, impact assessment and traceability.
- **Sharing and reviewing documents/work products**
  The ability for data modification, including approval of requirements via standard documents, is a key factor in collaboration with external stakeholders. Document export and import capability of documents containing managed artifacts would be exported to a Word or Excel document, which can then be shared with and reviewed by people who don't have access to system. After changes (the type of which can be optionally restricted during export), the Word or Excel document can be re-imported, where the changes it contains are incorporated into the data repository, and the document history is updated.
- **Web-based collaboration**
  Inviting external partners to comment on work items or documents using a Web client interface requires only a web browser and an internet connection. In this way, OEMs and suppliers can work and collaborate hand in hand without media barriers or loss of information and data.

Risk management

Product development teams manage risk at many levels, including:

- Ensuring that the latest requirements and specifications are available to all and are communicated clearly in a timely way
- Helping to spot resource bottlenecks before they reach critical mass
- Revealing the impact (and cost) of change before resources are committed
- Ensuring adherence to process and compliance with standards
- Providing visibility on what was changed, when, by whom and why

Taken all together, these capabilities provide a solid basis for risk analysis and fulfillment of DO178C in every project and across all projects.

Agile software development

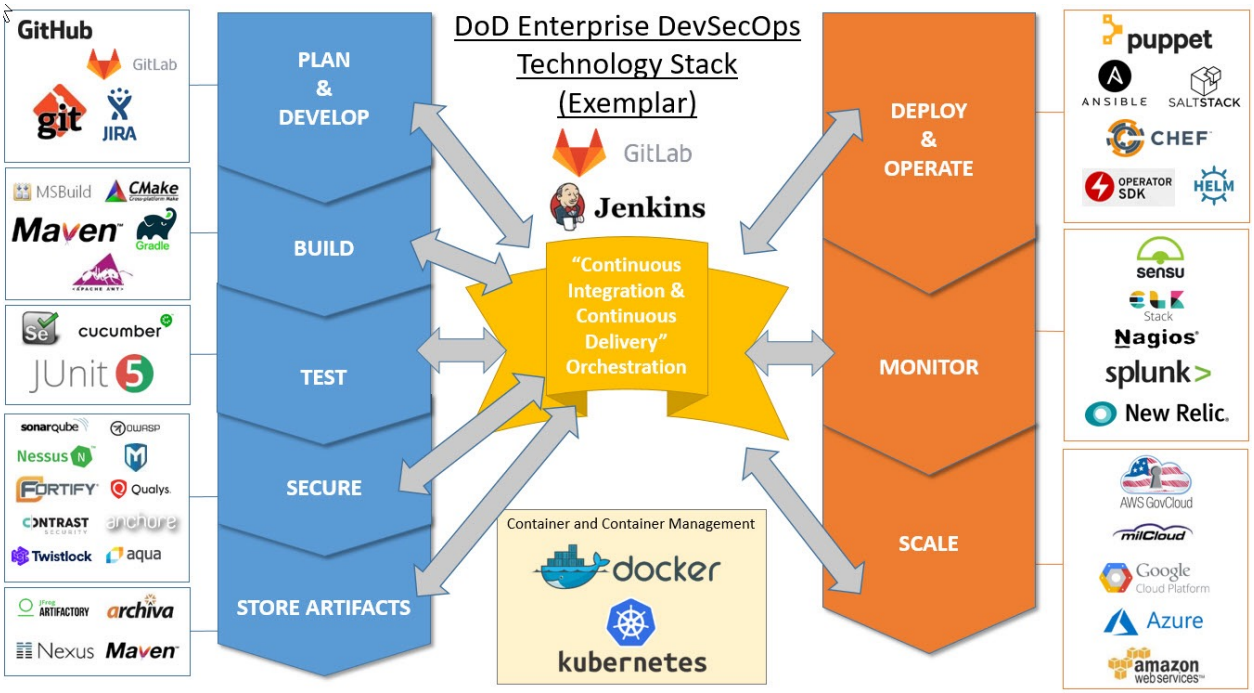
Adopting an Agile methodology or approach at an enterprise scale has challenges, especially when adopting among a cross-functional team environment. For enterprises that have distributed and disconnected agile teams, getting every team on the same page is a challenge. This is why it's important to adopt a homogeneous strategy with Agile using the Scaled Agile Framework (SAFe) to bring consistency across all the teams. Adopting a SAFe approach brings the following benefits:

- Faster time to market
- Improvements in quality
- Increase in productivity
- Better employee engagement

Agile adoption is accelerated when working with SAFe Platform Partners and based upon an approved SAFe template within the software development and management tool.

Software security

Protecting software code from intrusions and exploits has become a growing and critical challenge for software development organizations, and just as importantly to their end users. The concept of including security considerations within the development process, especially when Agile processes are employed, has attained high visibility, especially within the Department of Defense and for its contractors. So much so that the acronym DevSecOps has been attached to it, placing security squarely within the Continuous Integration and Continuous Development process of software development. The ability to deliver this capability significantly adds to the speed of development and further ensures the resilience of the software package.



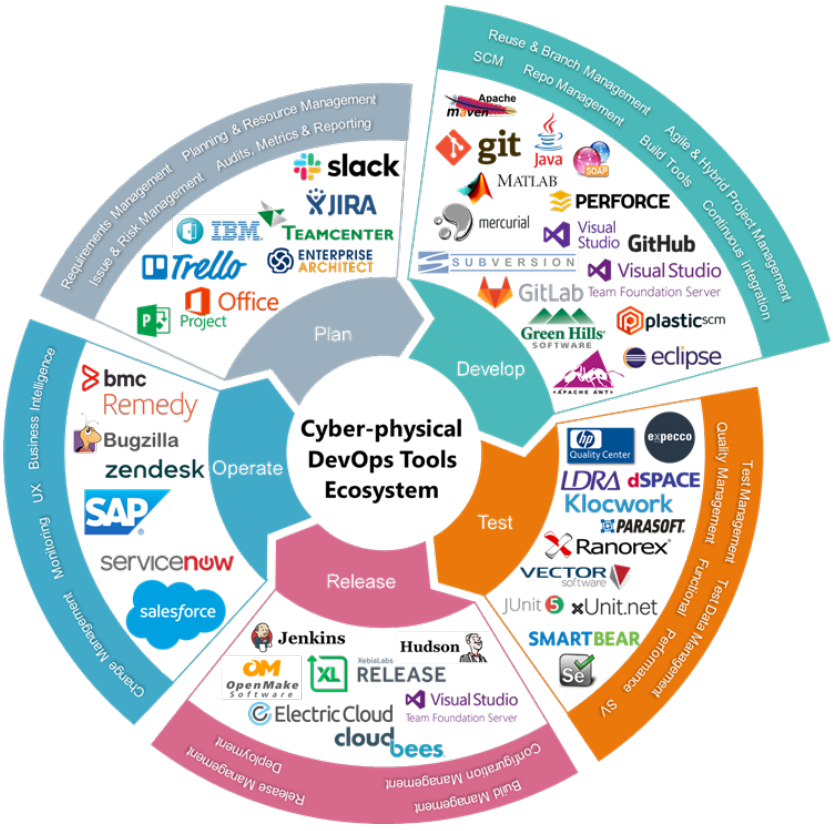
DoD DevSecOps Topology

Currently, this process is somewhat involved, requiring use of external tools to provide an analysis of the executable code against databases containing known threats and vulnerabilities. In conjunction with this security topic, there is growing interest and discussion about the creation of a Software Bill of Materials (SBOM) that not only specifies security of the software, but also every component that is used to construct the software.

Both of these ideas are important and useful to end users as well as organizations that develop software for any kind of use and should be included within the course of the development process, ideally within the same tool that is used for management of requirements, testing, release and resources. In the end, it is all about mitigating risk as it relates to exploitation of software assets.

Interoperability

Software development teams use a variety of different and specialized tools in the construction of a software program. Coordination of the results from those various tools needs to be tightly integrated into the development process. Access to a wide range of connectors and integrations is important when looking to streamline the process for efficiency and accuracy. Ideally, all the results from these external tools should have the ability to be managed within the development management framework.



DevOps Landscape

Integration with physical systems

As mentioned previously, today's complex products rely heavily on software. As such, the software process must be coordinated with the larger manufacturing process and included in the product lifecycle management (PLM) management process. Software will ultimately be embedded in some sort of mechanical or electrical device to provide the desired functionality. The right version of software must be matched with the right version of the physical component. To achieve this, these manufacturing processes must also contain information about the software and how it is to be implemented within the physical system or subsystem. Therefore, there needs to be a direct connection between both PLM and software application lifecycle management (ALM) systems for easy validation and verification of both the software and its relationship to the physical system it will reside within.

Electronic design automation (EDA)

More and more products are now employing embedded software right into integrated circuitry. Viability of the software needs to be validated and verified within the confines of the related integrated circuit topology. Software must be evaluated and tested within this infrastructure, which is executed a bit differently. The management environment must also have the capability to include this within the software development process through links or integrations.

Model based systems engineering (MBSE)

Product engineering today takes advantage of several systems that allow design and analysis using graphical based models. As with other engineering disciplines, software development should embrace the various modeling tools to include the picture of software within the model.

Summary

Top 10 benefits of a web-based solution for management of aerospace software development

- Capture and manage requirements and changes using one solution and one repository
- Leverage existing assets and tools, and re-use requirements to increase overall efficiency
- Create granular traceability to track across all key artifacts down to source code
- Address software security to identify and eliminate threats and intrusions
- Provide real-time visibility on product status
- Establish requirement libraries to leverage standards and variants of requirements
- Automate and standardize workflows to establish governance
- Leverage risk management capabilities
- Provide a link between ALM and PLM systems
- Provide and establish interaction with modeling tools for inclusion in the MBSE process

Siemens Digital Industries Software

Americas: 1 800 498 5351

EMEA: 00 800 70002222

Asia-Pacific: 001 800 03061910

For additional numbers, click here.

Siemens Digital Industries Software helps organizations of all sizes digitally transform using software, hardware and services from the Siemens Xcelerator business platform. Siemens' software and the comprehensive digital twin enable companies to optimize their design, engineering and manufacturing processes to turn today's ideas into the sustainable products of the future. From chips to entire systems, from product to process, across all industries, Siemens Digital Industries Software – Accelerating transformation.

siemens.com/software

84601-D6 5/23 A