ovum

ALM–PLM integration: Why it matters for multi-domain engineering

Polarion ALM and its integration with Siemens Teamcenter

Management summary

Introduction

Software has become the defining differentiator in engineering manufacturing in multiple ways. For the manufacturer its engineers often have a choice to construct a requirement using mechanical, electronic, or software technology. Increasingly software wins as the choice that offers speed of creation, degree of flexibility/ease of change, and lowest cost. For consumers software adds value by providing choice and flexibility to configure custom preferences. A good example is how consumers faced with choosing a car include in their criteria the infotainment system and advanced driver assistance.

The trend with software-driven systems is for increasing software content; for example, high-end vehicles now contain 100 million lines of source code. The flip side to the benefits of software is that it contains bugs. The world of enterprise IT has been familiar with software quality issues since its earliest days and application lifecycle management (ALM) solutions grew as part of the drive to manage software development lifecycle (SDLC) complexity and quality. Now we see that as the sheer volume of software in engineered products increases it is necessary to manage the challenge of software quality and complexity in engineering.

Ovum has seen the highest rate of ALM system adoption in mechatronics industries that also embrace PLM. The highest potential for growth of ALM is in the PLM world. Although open-source tools dominate at the core developer level and tend to change frequently with development fashions and rapid advances in technology, the management tools need to be stable and long lasting. This applies to both enterprise IT and engineering IT, and ALM in engineering is still a relatively new concept.

In addition, the biggest disruptor in software development has been the emergence of agile and DevOps practices, which is also impacting software in engineering. A new generation of ALM solutions has grown to manage the new processes as well as continue to support legacy processes (waterfall and hybrids). Whether in enterprise IT or in engineering, ALM systems must support a range of patterns of work, from V-model-based processes to agile iterations and continuous engineering delivery.

Another significant technology disrupter to emerge is the Internet of Things (IoT). When combined with cloud services this gives the feedback loop in continuous engineering delivery a new significance, because products can be updated and serviced in situ, over the air. There are also implications for mining big data analytics in maintenance repair and operations and remote servicing, leading to improved customer service response and more.

Software engineering is taking center stage in building advanced manufactured products and ALM systems are essential to manage the ensuing software complexity and software quality challenges. With ALM systems increasingly found in PLM environments there is the need to connect these systems together to optimize the production and reduce delivery time.

Siemens is a leading PLM provider with its Teamcenter solution and has acquired Polarion Software. The two companies worked closely together to integrate Teamcenter with Polarion ALM, offering a significant step forward in addressing the need for ALM–PLM integration. The current integration supports use cases that address integrated requirements management, change coordination and management, and closed-loop embedded systems software traceability. With the acquisition Siemens PLM will continue to enhance the ALM-PLM integration.

The topic of ALM–PLM integration is a key one in software-driven engineering today, addressing the management of software quality and complexity, often with safety critical aspects. There is a need for lifecycle traceability and this is driving the adoption of ALM across industries. To understand the advantages of integrating Polarion ALM with Siemens Teamcenter, the market needs to learn about the benefits that can be realized in ALM–PLM integration.

The need for ALM–PLM integration

Software adds benefits and complexity

The old days of mechanical engineering dominating manufacturing are being transformed by new technologies: electronic controls, computers, and software have taken over. In particular, software content in advanced engineered products is growing massively because it offers greater flexibility in design, ease of change, increasing product innovation, and greater re-use with product variants.

The current technology waves will accelerate this change. Cloud services and IoT are having far-reaching consequences for a range of industries, including mechatronics, automotive, and aerospace and defense. These waves are accelerating opportunities for servicing and upgrading products over the air. A wealth of monitoring data is also being captured and fed back over the air, leading to acceleration in product design and development as engineers reduce the cycle time in trialing new features and continuously improve products and production processes. Complexity arises with the fast pace of change as new technologies are introduced and new product development and introduction becomes more challenging.

Co-design processes between electrical-electronic and mechanical engineering are well established in Siemens Teamcenter. These design processes cannot exist separately – at certain points they must interact. For example, the effect of constraints (e.g., match dimensional constraints and mount points, achieve density and complexity requirements in smaller spaces) in one process will have an impact on the other. During simulation and analysis it is desirable to co-simulate all the engineering elements. More design and development work has been moved into virtual environments to speed the cycle time and improve quality.

The rise of software content in products only increases the complexity. For example, a major Chinese conglomerate has 2,000 software releases per year, entailing 50,000 builds per day, 100 million test cases run per day, 480,000 code-reviews per year, and 170,000 systems integrations per year. However, most engineering companies do not build and test every single component from a single location. Instead a complex supply chain of partners builds and tests various components of a product. This can result in thousands of release cycles for all the different components. The impact of a change in cycle frequency of one component will have a ripple effect to other components and the overall product update cycle.

Many different product models from different manufacturers share the same component with some variant, so that the change and variant management become extremely complex. In addition, variants due to regional requirements further complicate an already challenging situation.

The risks of not using an ALM system to manage software development are manifold:

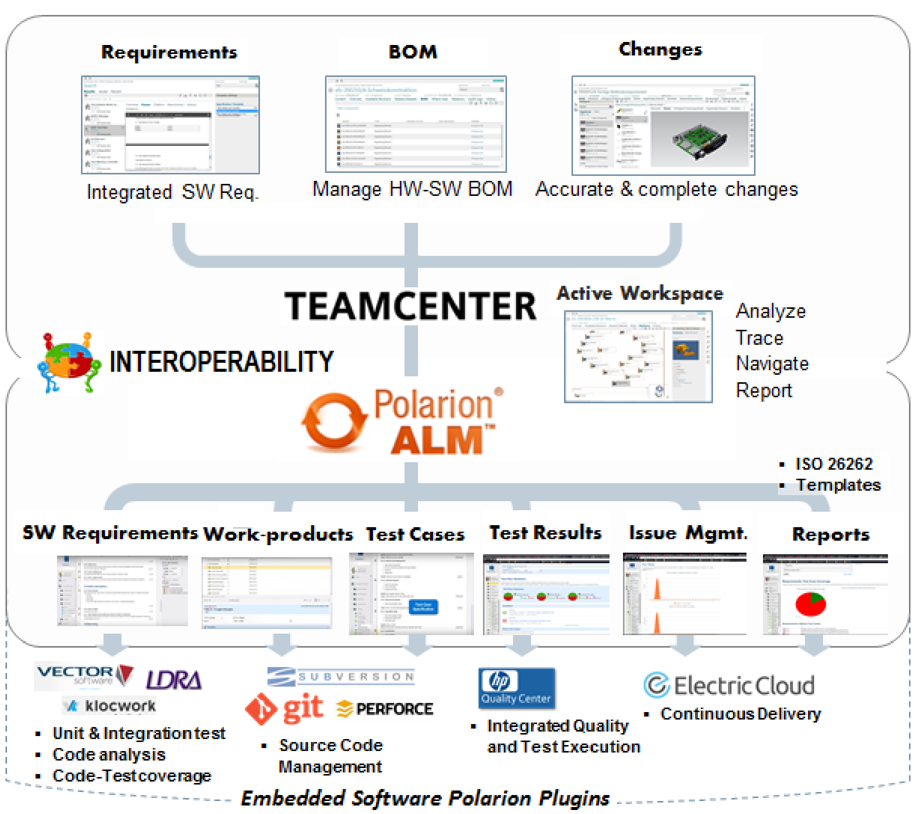
- Managers have no insight into project progress.
- Software quality problems such as trends related to modules and their bug counts are discovered late.
- Lack of traceability results in delays in gathering information, slowing development.
- Lack of collaboration and sharing of knowledge can result in duplicated work and knowledge walking out the door when developers move jobs.
- Without auditable systems there is no proper governance of safety critical components.

This list goes on and on. Once the size of the software development reaches a certain level, such as multiple teams and suppliers, the use of ALM solutions becomes essential.

Teamcenter and Polarion: ALM–PLM integration

Multi-domain engineering with the addition of software requires a closed loop between hardware and software design and development, with continuous engineering delivery. Across all the domain processes there needs to be a global orchestration of traceability, verification, and validation; security enforcement (authentication, authorization, and export limitation); and monitoring of quality and cost, leading to predictable quality and reducing wastage, such as duplicate efforts and infrastructure.

Figure 1: Siemens PLM integrated system design and engineering



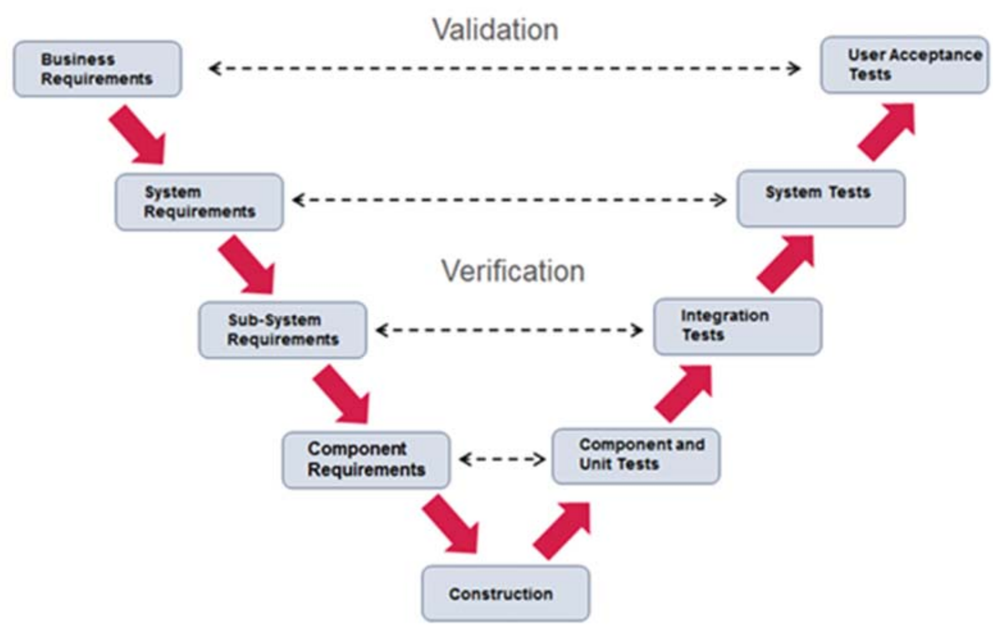
Source: Siemens

Figure 1 shows the roles the Siemens PLM Software's Polarion and Teamcenter solutions play. Teamcenter is where all the electrical and mechanical processes and design, development, and production engineering data are collected and the various engineering processes orchestrated. Polarion is where all the software development project data is collected, where third-party executions and test results are collected, software builds are managed, and other software lifecycle management orchestrations are conducted.

For example a requirement arrives from the PLM solution into Polarion ALM and a closed-loop change management is performed with the PLM solution and updated as the requirement is fulfilled in the ALM process. Once a binary is ready to be built, the Polarion release and build management solution can be implemented, possibly with the addition of third-party build accelerators, to complete the continuous engineering delivery.

Polarion requirements can read a wide range of document types, including models written in UML, SysML, Matlab, and Simulink documents. The ALM–PLM integration benefit is that every document that can be displayed in Teamcenter can be exposed in Polarion. Only information is transferred between the ALM and PLM solutions, so that data always stays at source. Teamcenter is driven through requirements, and in particular change in requirements, with nearly all new requirements based on a change of existing requirements. Requirements are central to how Teamcenter manages PLM and therefore the integration with Polarion ALM requirements management is a critical part of the overall ALM–PLM integration.

Figure 2: The engineering V-model



Source: Ovum

Test management is the next stage in ALM–PLM integration

Requirements and testing follow closely in ALM. Testing has two main purposes: it verifies that requirements are being met in the developed code (doing the right thing) and it validates that the code is working correctly (doing the thing right). In the engineering V-model shown in Figure 2 requirements represent the left arm and testing the right arm. Please note that this model can be applied in both waterfall and iterative processes.

Integrated ALM systems are designed to manage the connection between these arms and will integrate with the various testing tools, both automated and manual, that perform the actual testing. To facilitate governance such as compliance and auditing the ALM and PLM solutions need to interface and be able to interrogate the systems for the status of testing.

Engineering manufacturing would benefit from more standards in the ALM and PLM tools community so that tools could transfer information and interoperate among themselves out of the box and with minimal custom programming. Some standards do enjoy popularity: in the automotive industry Requirements Interchange Format (ReqIF) is mandatory. To date there is no universal “plug-and-play” equivalent adopted for quality assurance and testing, although the Object Management Group launched a first version of a new Test Information Interchange Format (TestIF) in May 2015 and there is a clear need for such a standard. The software tools industry needs a joint standards approach to improve the management of test cases: test case generation, test case data, execution, rollback, managing different types of test, and how it affects continuous delivery – not just within ALM but also with ALM–PLM integration

To deal with this integration challenge, Polarion QA and xUnit Integration offer test information integration out of the box, supporting third-generation test automation software that deploys xUnit or jUnit testing frameworks, and via APIs and out-of-the-box integration with the leading software testing tool solutions.

Variant management

Product lines today contain more variants than ever before. Manufacturers can release new variants on a monthly basis with the very latest feature enhancements. Software makes it possible to multiply the number of choices available on the market to match the precise needs of customers. From a logistics view point this creates a management challenge and variant management is growing in importance in engineering ALM.

Manufacturers have grappled with the design complexities of multiple variants by designing for variants from the start, creating modular, component architectures. Keeping track of these variants is best done using a dedicated tool such as Polarion Variants. Variants allow manufacturers to satisfy the long-tail needs of a diverse set of customers while achieving the customization within budget. ALM and variant management allow manufacturers to achieve such ambitions.

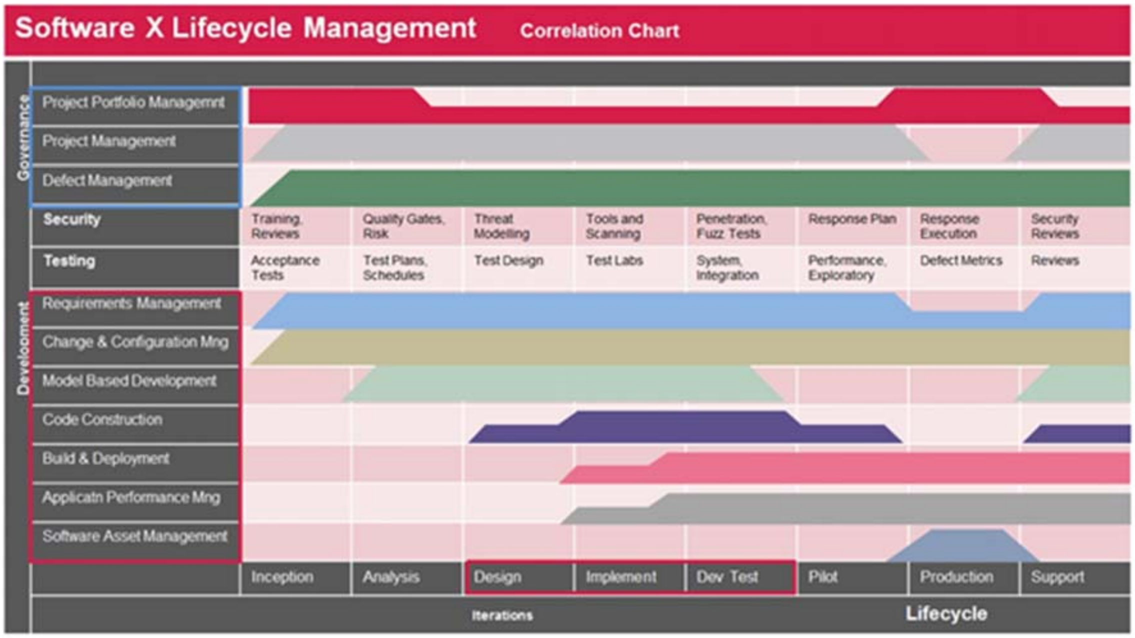
As manufacturers gain more experience in using software in their products and the amount of software in products consequently increases, the need for variant management as a specialist discipline in requirements engineering will grow. This discipline also ties in well with product line engineering; as a combination they will lead to creation of software factories in the near future.

The multiple lifecycles in ALM

The familiar ALM concept has been seen mainly as SDLC focused, but there are other lifecycles within ALM. For example, testing practitioners have been using a dedicated lifecycle for managing test case generation, test execution and results analysis, and more. In addition, security practitioners have honed a software security development lifecycle based on best practices in baking-in security throughout the SDLC.

Ovum has introduced the Software X Lifecycle concept to highlight the multiple lifecycles that run concurrently in software development. Figure 2 shows how project governance, development, testing, and security apply during software development. The x-axis does not represent a process, but rather phases of the lifecycle. The figure should therefore not be read as a strict timeline, but as a correlation between disciplines and lifecycle phases. It is applicable to any chosen development methodology or process.

Figure 3: Governance, development, testing, and security lifecycles



Source: Ovum

The future for ALM–PLM integration is to coordinate the concurrent PLM processes with these concurrent ALM processes.

Case study: CNH Industrial

The engineering background at CNHI

CNH Industrial (CNHI) is a global manufacturer of tractors, combines, excavators, wheel loaders, trucks, buses, firefighting, and civil protection vehicles. It has 64 manufacturing plants, 49 research and development centers, and a workforce of more than 69,000 people in 190 countries. CNHI's market is typically building highly specialized vehicles in low volumes, rather than mass vehicle production. Ovum spoke with Edoardo Sivera, who runs a cross-departmental team responsible for methodologies, standards, and supporting other teams in adopting new technologies. The introduction of Polarion to CNHI is a major undertaking with the aim of improving processes.

A large amount of custom embedded software is developed for any given vehicle. This is produced by internal teams and by external partners and suppliers, with as many as 15 to 20 contributing to one vehicle. For example, a typical truck will have 20 to 30 SKUs with software engineering content: body computer, dashboard, navigation system, engine control unit, braking control unit, and so on. Each SKU can contain from 20 to 100 applications. One of the key tasks for CNHI is integrating the work items from the various suppliers and internal teams. The software programming languages used are typically C for body electronics and engine control and C++ for infotainment applications, with increasingly Java used for these applications.

Sivera's team is responsible for the software in the SKUs, running a central point within CNHI for software development standards. The embedded software ends up in dashboard vehicle controls, instrument clusters, and displays. For example, tractors today have large displays providing a mass of information. A typical application transferred to flash memory varies from 500KB to 24MB in size.

Although the decision to adopt ALM was made by the engineers, the acceptance of the solution met with some resistance due to the clash of cultures that ALM introduces within the PLM world. To manage the traceability and the impact analysis of changes it is necessary for the engineers to create links between requirements, Matlab/Simulink/Stateflow models, and test cases. Polarion creates the links automatically and manages the traceability, but the engineers can be reluctant to take the time to do so. This is the cultural challenge: although some engineers are working enthusiastically with ALM, others are resisting the new tool. This is a continuing change management challenge that will take some years to fully unfold.

The interaction between the many external suppliers and the 20+ internal design and development centers is also complex. Some external suppliers are already Polarion users, but with those that are not there is little leverage that CNHI can apply due to the low production volumes and the reluctance of these suppliers to bear the cost of switching tools. Again, the resolution of these challenges will come with better support for plug-and-play tool interoperability standards.

CNHI has been using Polarion as its global ALM solution since 2010. Where teams are using Polarion the process of work flow has been made easier; such centers are linked together to share information using a common database. The processes used in development span waterfall, agile, and hybrid. Where waterfall is used it follows the traditional engineering V-model and the main processes are agile and V-model with an incremental lifecycle. Projects typically have three or four iterations before delivering the final product, driven by the combined software, electrical, and mechanical lifecycles. In addition, model-based design and development is the norm for all projects started at CNHI since 2012, exploiting simulation and auto code generation for speeding development and improving quality. Given the many contributing suppliers and sources of software, CNHI has developed a modular architecture for both the hardware and software.

CNHI uses Siemens Teamcenter for its PLM solution, and started working with the ALM–PLM integrated edition in July 2014, so it is still early days for the project. Only a subset of the 20–22 centers has been trying out the integration. Siemens Teamcenter has been present in CNHI since approximately 2000 and is a stable and consolidated tool within the organization, broadly used for product description, creating bills of materials, and so on. However, Teamcenter is a typical mechanical engineers' tool, used to manage the components and the physical parts of a vehicle. It became necessary to integrate Teamcenter with Polarion and create a bridge between the software development and the other parts of engineering a vehicle.

Benefits of the ALM–PLM transformation at CNHI

The integration approach taken is to keep data at source and create live links across the tool chain: data residing in Teamcenter stays in Teamcenter and data residing in Polarion stays in Polarion. Triggers implemented at certain points in the various engineering processes initiate action across the ALM–PLM divide, in both directions. In this way it is possible to create automated workflows that take place concurrently in Teamcenter and Polarion and that cross the tool divide. For example, CNHI has spent years perfecting an environment for managing defect issues arising in product validation. When a defect is discovered it is logged and evaluated and if it has a software-related dimension it is entered into Teamcenter as a software problem. With the new integration this will automatically trigger a defect issue in Polarion.

A major aim of the ALM–PLM integration is the traceability of requirements, so that, for example, a software requirement is traceable to its implementation as embedded code in an electronic control unit. Whereas automated traceability within PLM and within ALM is possible to perform, the work to manage fully automated lifecycle traceability across the ALM and PLM divide is still ongoing, with some parts of the process requiring manual link creation. This creates a chore for engineers and meets with the heaviest resistance. However, as Teamcenter and Polarion continue to bridge the processes across the divide, traceability will improve and with it the adoption of ALM. One of the most critical areas from a quality assurance perspective is the traceability between requirements (from system to components) and testing. This is a high priority for the ALM–PLM integration project by the tool vendors.

CNHI has seen a big improvement in task or issue management. Prior to the ALM–PLM integration it was difficult to trace all activities in a project. A project typically has a long list of activities assigned to people and teams and it is now possible to know quickly the status of tasks – how many are completed, in progress, or waiting to start. For managers there is a clear view of the status of a project and this is the biggest improvement over the last year.

The most common form of writing requirements is in Microsoft Office products: Excel, PowerPoint, and Word. Fortunately Polarion has a round-trip mechanism so that a requirement in a Word document can be read and generated by Polarion. This means that CNHI suppliers can work with Office documents and Polarion can process these documents. The gap between the tools used between CNHI and its suppliers needs to be reduced for an improvement in speed to market. This is clearly a work in progress.

Conclusions

Ovum’s analysis

The engineering industries are facing a challenge in how they manage the software that is driving innovation and reducing costs, but at the same time introducing new challenges related to software quality, complexity, and security. ALM–PLM integration will become recognized as a necessity in ensuring these challenges are met and the benefits realized.

The Siemens acquisition of Polarion Software will enable an advanced integration to be offered between Polarion ALM and Teamcenter. This integration project has recently completed the first phase of a multiphase integration road map. There are few examples of such strong ALM–PLM integration on the market and the Polarion suite of products should be at the top of any Siemens Teamcenter-based manufacturer’s ALM short list. It offers unparalleled access to Siemens Teamcenter, resulting in deep ALM–PLM integration, and the comprehensive functionality available in Polarion ALM.

The ALM and PLM solutions exist within an ecosystem of design and development tools and for users to gain the optimum benefit of all these tools they ideally need to interoperate and exchange data. Therefore the industry as a whole will benefit from the creation and adoption of suitable standards. Standards organizations and industry bodies are driving the introduction of standards across the

SDLC and it will take some time for these to gain adoption, but there is a need for such standards if the benefits of software-rich engineered products are to be realized in safe and secure machines.

Appendix

Author

Michael Azoff, Principal Analyst, Software Solutions

michael.azoff@ovum.com

Ovum Consulting

We hope that this analysis will help you make informed and imaginative business decisions. If you have further requirements, Ovum's consulting team may be able to help you. For more information about Ovum's consulting capabilities, please contact us directly at consulting@ovum.com.

Copyright notice and disclaimer

CONTACT US

www.ovum.com

askananalyst@ovum.com

INTERNATIONAL OFFICES

Beijing

Dubai

Hong Kong

Hyderabad

Johannesburg

London

Melbourne

New York

San Francisco

Sao Paulo

Tokyo