Podcast summary

Product Line Engineering: Underwriting ALM and PLM convergence

Bola Rotibi

August 2010

This document accompanies a podcast discussion led by MWD Advisors on "Product Line Engineering: Underwriting ALM and PLM convergence". It summarises the main guidelines and key messages highlighted in the discussion. To access the podcast audio please follow this link or see the instructions on page four of this document.

MWD Advisors is a specialist IT advisory firm which focuses exclusively on issues concerning **IT-business alignment**. We use our significant industry experience, acknowledged expertise, and a flexible approach to advise businesses on IT architecture, integration, management, organisation and culture.

www.mwdadvisors.com

This paper has been sponsored by IBM

Introduction and overview

Over the past decade, advances in software, hardware and networking technologies have enabled more and more technology to be woven into and around physical goods, adding value to those goods even without that technology necessarily being "embedded" in any hard-wired sense.

Many see the greater convergence between different physical devices and the rise of "systems of systems" precipitated by the increasing use of embedded software to drive functionality and enable greater management control and connectivity. Software provides the possibility to deliver feature variability that can be updated or removed without potentially changing the underlying base hardware, as well as the ability to obtain greater management control, especially remotely.

While many products will continue to be connected together by electronic circuit boards, wires and appropriate mechanical parts, it will be software embedded within the collective systems that will allow higher forms of intelligent integration and application to be achieved.

The embedded software and complex systems market is now both broad and varied. It is a market that ranges from software components and applications embedded in hand-held devices and consumer appliances, to large complex computational software systems controlling big mechanical constructs such as tanks and planes. This is a market that is rich in growth opportunities and is exciting both in terms of commercial reach and of what can be achieved.

Ultimately, the convergence of PLM and ALM is occurring because of the increasing role that software has in product innovation. So the need to collaborate throughout the development lifecycle – and understanding the cascading impacts that changes and enhancements have on the other domains is key. Developing multiple flavours of a product – or product lines – adds additional complexity. The fact that much of this innovation is coming out of the software or ALM lifecycle makes it even more important to integrate ALM and PLM processes.

As a consequence of the wider use and investment in software, the supply chain through which smarter products are designed, integrated, tested, delivered and serviced is a lot more complex and involved than the more traditional product delivery and lifecycle management process. However, there is a disconnect in the convergence between software and product delivery lifecycles that is challenging those developing and delivering smarter products. As a solution to this disconnect, Product Line Engineering (PLE) presents a mechanism and an approach that leverages manufacturing concepts of production line delivery and supports the common themes impacting the software delivery process to offer a framework for bringing together and managing the convergence between software delivery lifecycles (and the concerns of Application Lifecycle Management, or ALM) and the Product Lifecycle Management (PLM) process.

In this podcast we discussed the role and value of PLE as a cross discipline/domain approach to managing the additional complexity that arises from converging the disciplines of ALM and PLM and the challenge of feature variations in order to bring about a unified governance and management framework for the delivery of "smarter products".

The discussion focused on four key points:

1. The role that PLM and ALM play in the delivery of software systems and products (complex or otherwise)
2. The need for converging both the ALM and PLM disciplines and the challenges faced
3. The role that a PLE platform and framework plays in supporting the convergence of two well-defined and heavily-practised disciplines with long-established tooling products, processes and methodologies in place.
4. Important considerations when choosing a PLE strategy and tooling platform and key directions for involving and engaging business sponsors

Key conclusions and recommendations

The discussion delivered a number of conclusions and recommendations based on the experiences and knowledge of the participants in working with clients in the market.

- **An effective PLE strategy requires the right tooling platform.** Don't forget the importance of software as the critical factor in complex systems and products. Just as important will be the support of a platform architecture that recognises that different dynamics, relationships and constraints are at play across the product lifecycle that is impacted by both ALM and PLM processes. What is necessary is a platform framework that focuses on the software and systems delivery lifecycle process as a unified supply chain. This is not about a one size tooling approach. An effective PLE platform will offer a workbench approach that will employ open interfaces to integrate different solutions and tools together to manage the requirements in design and the quality across the entire product lifecycle. Such a platform will optimise and maximise the capabilities of both application and product lifecycle management tools. This is a case for using the right tools together and equipping your developers in the right way – be they mechanical engineers, systems engineers or software engineers – and enabling the right level of collaboration and communication to occur. Ultimately, an effective PLE strategy and platform brings manageability to the complexity of delivering different software components or systems for different hardware configurations and keeping the different delivery processes (ALM/PLM) aligned and in sync. The PLE approach recognises, allows for and supports the different process dynamics involved. It offers an engineering discipline and solution for what is ultimately an engineering problem.
- **PLE is an important model for addressing the challenges of an inevitable trend.** Systems and complex product providers need to recognise the important role a Product Line Engineering platform and approach plays in supporting the necessary convergence of ALM and PLM processes. It is a necessity brought about by the wider role that software plays and the complexity it adds to the overall product delivery process. Organisations can no longer allow their PLM and ALM lifecycles to run independently of each other. We are now beginning to see a rise in the risks, errors and critical issues associated with products developed and delivered through processes that continue to promote their separation. The stakes are too high and organisations can no longer ignore the problem. ALM and PLM need to converge, supported by a tooling platform that takes advantage of the best of breed capabilities from the respective lifecycle and the tools that support them. PLE is a recommended approach to designing, developing and delivering a product line – taking into account the convergence of PLM and ALM. It is the ability to provide the "glue" mechanism for achieving this, and in providing a feature variation management abstraction layer consistently across both lifecycles, that makes PLE the most appropriate and important framework for underpinning this convergence. PLE ultimately provides the mechanism for the parallel development and delivery of reusable systems across the domain of disciplines involved in delivering a complex portfolio of interconnected products and components.
- **PLE is a transformational process that requires organisational buy-in and clearly defined goals.** An effective PLE platform and strategy will eventually impact everybody up and down the lifecycle, all the way from the individual developers up to the highest level executives in the organisation, and will see a change in the way they go about doing their work. Don't underestimate the focus and the discipline that is needed to move into a highly effective PLE strategy. To make it work effectively, organisations will need to undertake a serious examination of what is going to be required to achieve success. They will need to define success metrics carefully and develop a strategy for incremental, non-disruptive transformation.

About the podcast

The discussion was led by Bola Rotibi, Principal Analyst at MWD Advisors.

The two guest speakers joining the podcast discussion were:

- Kim Roberts, a world-wide Industry Solution Executive with IBM Rational, with extensive expertise in building strategies to help clients tackle the challenges that are derived from establishing and modernising quality software and systems development processes while maintaining compliance with various regulations and standards.
- Dr Charles Krueger, the founder and CEO of BigLever Software, a leading provider of systems and software product line (SPL) and Product Line Engineering frameworks, tools and services, with proven expertise in leading commercial software product line development teams, and helping companies establish some of the industry's most highly acclaimed SPL practices.

We thank both our guest speakers for their valuable insights and contributions to the discussion.

Listen to the podcast discussion

To listen to the podcast discussion in full please follow this link or copy the link below into your web browser:

https://www.ibm.com/services/forms/signup.do?source=swg-rtl_tl_genisv&S_PKG=pd_BigLever-PLE-ALM-PLM

Podcast summary

In this section we drill down into some of the detail under each of the four discussion points.

Discussion point #1: The role that PLM and ALM play in the delivery of software, systems and products (complex or otherwise)

Both ALM and PLM are well-established and well-practised disciplines for the delivery of software applications and products respectively. The discussion outlined the key role and purpose of both disciplines in delivering software enabled systems and products.

- **PLM: governance and management platform for the product lifecycle.** Product Lifecycle Management (PLM) is commonly defined as the process for managing the entire lifecycle of a product from its conception, through design and manufacture, to service and disposal. Within the wider market, PLM solutions and platform products essentially refer to the class of tools used for the management of the lifecycle of mechanical product lines and the management of the corresponding physical parts (e.g. electrical/electronic), the computer-aided mechanical design and the bill of materials (BOM) of all the physical components. Ultimately, PLM platform products and strategies go beyond the lifecycle of the mechanical build and design of the product to integrate people, data, processes and business systems and provide a product information backbone for companies and their extended enterprise.

- **ALM: governance and management process for a software application lifecycle.** ALM can be more broadly defined, depending on the focus of the project. Within the context of software business applications, Application Lifecycle Management (ALM) is a process and governance framework for managing the delivery and lifecycle of software products and business applications deployed within corporate IT systems and infrastructure. From the viewpoint of the systems and embedded software market, a core sector driving the development for "smarter products", ALM solutions and strategies look to manage the entire development lifecycle of all the application components of the product. In this context, the application component relates to the software and higher systems models, and also includes the overall system and how the software interacts across all components of the system including the mechanical and electrical components. Like PLM, the ALM framework stretches to factor in considerations for people, processes and tools.

Importantly, both disciplines encompass a lifecycle strategy that goes beyond the technological implementation to encompass and address people (behaviour, culture), broader policies and constraints, data information and other connected systems to ensure robust governance and traceability. The end-to-end governance and management framework of both disciplines is an imperative for the product delivery process for a number of significant reasons:

- Compliance with safety and quality critical constraints is paramount because of the dire consequences that can occur when they are breached.

- There is a more tangible impact to the business bottom line and brand equity when a product fails or requires a recall. An end-to-end governance and management framework can provide the level of traceability and process control needed to enable business and regulatory compliance.

- The forensic investigation to prevent future reoccurrences of known issues or to prove or disprove culpability in the case of legal pursuits requires a framework where there is full traceability of all aspects of the product workflow.

- Time-to-market and adaptability-to-market dynamics, regulation changes and technological advances improve with the control and discipline of an integrated governance and management framework and connected repository architecture that manages the relationships and dependencies of all product assets.

Discussion point #2: The need for converging both the ALM and PLM disciplines and the challenges faced

Both guest participants in the discussion highlighted the main reasons why the convergence of PLM and ALM is critical to really driving effective smart product development and delivery and the challenges that this presents.

The growing market for more software driving "smarter product" enablement

The increasing role of software has been an on-going trend over the last two decades that is pushing the boundaries and capabilities of traditional mechatronic products and opening up the way for new sophistications.

The broadening and deepening scope of software-related functionality is coupled with a trend towards "smarter products". With this, the role of software is growing increasingly in importance and value.

The software factor:

- **Product engagement.** In the past, the focus was on hardware and electrical and electronic systems; but now the focus for innovation is very much on the role of software. More software functionality is being incorporated into all manner of devices that previously would have had very little software embedded. Products and systems are becoming more dependent on software for the delivery of innovative features, and also for the delivery of compelling user experiences for customers. This allows all sorts of new interactions and experiences to be delivered (e.g. enabling medical personnel to employ their skills more effectively and ergonomically with sophisticated intelligent precision devices).

- **Software dependency and product line expansion through feature variation and innovation.** Many systems and products today have a critical dependency on the functionality from embedded software: cell phones with smart applications and firmware; automotive and aerospace sophistication; medical equipment advancement and finesse. Software also helps providers create and deliver feature variations across product lines without changing underlying product hardware (so reducing cost), increase product lifetimes, reduce support costs (enabling feature fixes to be applied to products without lengthy and costly returns processes), and deliver products and systems that can be monitored and managed remotely.

- **Hardware longevity and rising R&D costs.** The roles that software can play have been enlarged due to the maturity and availability of low-cost, high-powered yet open and embeddable computing platforms. For many organisations in the complex systems and mechatronic equipment market, investment in software development has reached parity with those made for hardware development. Much of the underlying drive towards software has resulted from hardware adaptability being too expensive in terms of both research and delivery costs. Additionally, the current generation of programmable controllers and flash memory means that hardware has achieved an extended lifecycle. Newer protocols can be sent to the hardware and incorporated without the need for expensive hardware board changes. This, from an engineering perspective, has meant that hardware installations can now be treated as general tools rather than as a large number of specialist components.

The differing dynamics of the virtual (software) and the physical (mechatronic) requires a process delivery and management model that recognises and addresses their unique constraints and shared goals for effective deployment and maintenance

Smarter product requirements need to be considered less in terms of the delivery processes relating to individual components and instead, more in the context of the delivery of a set of integrated software components and systems that together form part of a fully-functioning product – be that a car, handheld device, industrial machine, consumer appliance or the avionic control system of a plane.

The amount of software in systems and products requires a more organised approach to change management, maintenance and problem resolution that recognises the dynamics of the virtual as much as it does of the physical. ALM and PLM are two separate but equally critical components of the overall product development lifecycle. Both are driven by different sets of constraints: physics drives a lot of the assembly in the PLM world, whilst the complexity of the different states and possible interactions is a factor of the ALM process. Their differing dynamics can present a management, traceability and compliance headache in the event of product problems.

In "smarter product" production there is a requisite for the convergence of the PLM and ALM processes. This necessity is to improve the management and governance of a delivery process that is inclusive of the mechanical and electrical/electronic components and the embedded software across all variations of the product line.

The ALM process is simply not bound by the categorisation of a Bill of Materials

The ALM process is not simply another Bill of Material (BOM) component. The virtual and malleable nature of software makes it more adaptive and flexible in support of change; a hardware BOM component is fixed, whereas a software component can be configurable.

The application lifecycle phase consists primarily of strategic portfolio planning or project management requirements for the software design and development, along with verification and validation of the system or component before it is released. All of these things do not have mechanical parts per se, or materials that could be managed in something like a bill of materials for manufacturing, or even an engineering bill of materials that is used to put together the product.

More importantly, the software development and delivery element of the product lifecycle can be (and in many cases it is) a fraction of the overall product delivery timeline. Software evolves over time and this is very different to managing it as a BOM, especially when hardware constraints further down the line present the need for a "software" change. ALM tools and processes are needed since they are best suited to control the variations of the software and systems delivery process over the course of the product development and product delivery process timeline. The PLM approach, on the other hand, is best designed to manage the variations and the mechanical configurations of the parts as dictated by a BOM or engineering bill of materials.

Parity in software and hardware R&D spend raises the prospect for greater complexity and risk, driving a business demand for bringing an end to separations of concern and interaction

In the past, software played a lesser role in the value and functionality of systems; instead, the processes driving mechanical and electrical delivery held more importance.

As software usage was less and in general a lot simpler, many organisations found that they could manage the ALM and PLM processes separately and then manage the integration of the software and hardware at the end without too much upheaval. In some cases the small amount and simplicity of the software used meant that the software development process could be handled by the hardware teams to provide simple adaptability. However, advances in chip processors, enabling greater computational/intelligent features to be built within, has underpinned the rise in software usage. Further standardisation and computing power of available hardware has also allowed for greater reuse and the rise in use of standard software systems and components over proprietary developed ones.

Not only have these actions lowered the barriers to entry for many suppliers, but they have augmented the focus on providing differentiation through advanced software-based functionality and services. With this and other technology advances, software usage has increased to the extent that today significant R&D investment is spent on software innovation. The increasing parity in the spend on software innovation with that of hardware (in some cases, software R&D gets the lion's share) coupled with hardware advances is resulting in more complex systems and greater complexity for both lifecycles. Greater complexity from wider and more integrated use of software means that the individual development and delivery processes can no longer be kept separate or in isolation over the course of the complete product delivery lifecycle.

More importantly the industry is starting to see very high profile and publicly-visible failures and defects due to complex systems that did not have coordinated software and product delivery processes. The effect and impact on the business (damage to brand equity, the financial cost of addressing legal claims and resolving the issues and company reputation) has resulted in increasing demand for better cohesion between the two processes. Therefore it makes good business sense in having a conceptually integrated, complete and consistent lifecycle from the very beginning of the system inception all the way through parallel hardware and software design for all the different feature variations of the product.

As a result of all of the above, many smart product providers are now recognising and experiencing the following:

- Software development and inclusion is a prominent reality of product delivery; it is rare to encounter a product that doesn't have software embedded as some type of component within it adding new features or intelligence. Televisions, for example, have moved beyond being purely about the electrical and mechanical physical components to include embedded software that expands their capability and connectivity.
- The old approach of separate and non-communicative processes until the final integration is no longer working.
- Software, and the overall management of the product system as a whole, is allowing businesses to create more reliable innovative products that have greater meaning and differentiation within the market place.
- Convergence between ALM and the PLM process is necessary to manage the complexities of software and hardware integration, ensuring the overall quality of the delivered product and protecting market position and reputation.

Discussion point #3: The role that a PLE platform and framework plays in supporting the convergence of two well-defined and heavily-practised disciplines with long-established tooling products, processes and methodologies in place.

The discipline and process workflows for Product Lifecycle Management lend themselves well to the process and governance model of ALM. But whilst there is a natural alignment between ALM and PLM, their individual scope, focus and dynamics requires a different governance model to manage their converged goals. The discussion centred on a number of key factors driving the need for Product Line Engineering as a considered framework and process for managing and orchestrating the convergence between the software and physical product delivery process into one consistent and collaborative lifecycle.

Market drive for multiple product variations and product line portfolios

Few smart products are created as a single product model. In reality most products will have a number of model variations based on software or hardware feature variations or configurations. Many providers see that the key to their business success depends on the infusion of new ideas and new technologies to bring products to market in a timely manner, as well as management of complexity and capability across product lines. In order to achieve this goal, today's software systems-based product development organisations are turning towards the concept of delivering product lines or a portfolio of similar products with variations in the feature and functions, rather than individual one-off products. They therefore need a PLE framework to manage complexity so that they are able to inject new capabilities into different lines and create variances of their products to meet different market needs and cost points.

Multiple product variations raise the complexity of ALM and PLM convergence requiring the need for consistent manageability and automation

The complexity that arises from having to manage the systems and engineering processes for both the software and hardware processes for each of the different feature variations and configurations makes the convergence of the ALM and PLM processes intractable and complex.

A PLE platform provides a framework and approach for managing all the different feature variations and configurations of a product as a single product line. Its mechanism for abstracting out and managing on the basis of the feature variation rather than at the lower level of component piece or part variation where there is no common theme, provides the "glue" by which one can control complexity in a consistent and compatible way down both the software and hardware lifecycles, ensuring they are integrated and kept in sync throughout the whole delivery process. In short, PLE manages relationships, dependencies and configurations based on the feature variation rather than at the individual component level. This allows for greater reuse and a more stable base point for change modifications and future innovation. In the automotive industry, for example, a single feature could be an adaptive cruise control system consisting of cruise controls that are aware of ambient traffic conditions. In actuality this one system would be a combination of bumpers with sensors that were able to assess the surrounding traffic and then adjust the speed and distance accordingly. This demonstrates the mix of hardware (bumpers, sensors, radars etc) and software (logic for handling the sensor data and implementing the controlling algorithms) concerns. In addressing this as a single feature variation, the PLE framework is able to provide the consistency and integration points that can keep both the ALM and PLM processes aligned, synchronised and on track individually.

Discussion point #4: Important considerations when choosing a PLE strategy and tooling platform and key directions for involving and engaging business sponsors

In determining the key considerations for a strong PLE platform and strategy approach, the discussion raised a number of key requirements:

- **An extensible platform and tooling interoperability framework** that is capable of managing the end-to-end product, application and services lifecycles across the product portfolio. It is important that it is able to align the business and integrate the various mechanical, electrical and software delivery collaboration and communication activities. It should offer reporting metrics and measurements that provide visibility into the correlations between the dependencies. Also key to this requirement is an interoperability framework that supports investments already made in tools and the wide ranging tooling arsenal that organisations across the smart product provider landscape have access to. While many of their challenges will be similar, their tooling portfolio will most likely not be and they will want to retain tooling skills investment. There is no "one size fits all" approach.

- **Methodology and process support** that recognises that a change of approach in transforming the business to do a better job at integrating across lifecycles is as important as the ability to allow a mix of tools to interact and interoperate well. Process support across the lifecycles and strategies for tooling integrations are also as important as it is for the individual lifecycles. The methodologies need to be able to tackle the evolution and transformation across the lifecycle whilst enabling the preservation of skills acquired in invested tools.
- **Agreed business success metrics** are vital to ensure that you are able to measure the success of the platform and process in a way that relates to and is understood by the business. Communicating and collaborating with the business to determine these metrics will help build up a business case where they can recognise the value more easily and relate it to business goals and the business bottom line.