IDC ANALYZE THE FUTURE

IDC ANALYST CONNECTION

*Sponsored by: IBM*

**Companies need to embrace digitally transforming their engineering processes to remain competitive in the evolving global market. To successfully develop products that integrate hardware and software, organizations need an integrated solution that supports both hardware and software development methodologies.**

# How to Fuel the Digital Engine Driving Product Development

*February 2020*

**Questions posed by:** IBM

**Answers by:** Melinda-Carol Ballou, Research Director, Agile ALM, Quality, and Portfolio Strategies

## Q. Products are becoming software-intensive, interconnected systems — "systems of systems." How should companies respond?

**A.** High-quality, adaptive software increasingly drives competitive positioning and business execution for products as well as coordination with supporting software-intensive systems and systems of systems. Yet when we look at the ways in which organizations must evolve to incorporate software into their products, we often see that those focused on mechanical, electrical, and/or systems engineering tend not to be as mature in their software life-cycle processes. In these same organizations, we typically see IT departments are much more mature and experienced with application development and software engineering. Therefore, these companies must rapidly evolve software-intensive product development life-cycle management strategies to address quality, change management, requirements, and coordination for end-to-end system development. Moving from siloed product development to the demands and complexity of systems of systems is vital to enabling complex product development in these highly competitive, collaborative, and dynamically changing environments while addressing compliance and security demands.

We also believe there will be a growing dependence on artificial intelligence (AI) and machine learning (ML) to enable organizations to deliver software for complex and diverse environments. It's hard enough to architect software for Internet of Things (IoT)–enabled mobile platforms, but the complexity is compounded by the need to continuously test, manage, and deploy changes across those environments and systems to enable responsive requirements as well as manage risk, value, and compliance. Assessment of gaps, evaluation and adoption of appropriate automated tools, and agile process change are imperatives. We observe responsive organizations underpinning adaptive product development with strategies for creating systems-of-systems life-cycle engineering that combine intelligent leverage of ML and AI with support across their development ecosystem.

In engineering environments, the development life cycle increasingly needs to span geodispersed teams and the partner/supplier ecosystem, embracing emerging technologies and smarter methods to optimize the entire engineering process. This approach is critical for organizations that depend on software for competitive product differentiation across dynamically changing markets.

## Q. What are the most promising investments that you see companies making to address increasingly complex product development?

**A.** In considering product development, start with what it is that you're seeking to accomplish with the product and the software that's enabling it. Requirements are like the DNA for products. But flexibility and adaptability in how you evolve requirements to be responsive to changing business needs and pressures are vital. If you don't have the requirements right, and you aren't interacting in adaptive ways to enable change from a business perspective, then the product will not reflect what the business needs, which is especially important in rapidly changing, highly competitive markets. Companies need agile, iterative approaches for gathering requirements and user stories and epics and for scaling those processes to adaptively respond to changing competitive demands across enterprises, suppliers, and customers.

Requirements are becoming increasingly complex as we move toward software-intensive product development. In that context, we see ML, AI, and intelligent analytics playing key roles in enabling organizations to benefit from the knowledge of more experienced engineers and also to provide visibility into metrics. These technologies can help identify anomalies and corrections up front for issues that are problematic so that engineers can be more efficient in leveraging work that's been done. AI can augment requirements checking to enable more sophisticated and experienced engineers to do higher-value creative work. Making metrics visible proactively can give context to engineers creating software and products so that the data is pragmatic and actionable to improve execution.

Another aspect is the role that model-based systems engineering can play. We find that this approach helps model requirements and systems design to enable better understanding of required underlying functionality and structured interfaces. When you add simulation to those models, you can help ensure the validity of the design by decomposing the complexity of the product and modeling its subsystems.

Product complexity is also driving the need to coordinate and co-innovate with clients and partners as companies expand their dependence on external suppliers to help develop increasingly sophisticated products. They must consider how they control the product development life cycle, and coordinate both internal and external information, as well as intellectual property and safeguards for proprietary innovations to protect what they're doing.

Therefore, companies require an engineering life-cycle management solution that encompasses partners, suppliers, and customers to integrate that feedback into development processes rather than having it be fractured and noncollaborative. A shared engineering system of record (SOR) will enable organizations and external partners and suppliers to have a single source of proof for engineering information and the ability to exchange timely data for reporting, making it easier to integrate development ecosystems with the engineering life cycle.

## Q. As products evolve into software-intensive systems that "cannot afford to fail," how should companies adapt development processes that can support the oversight and checks that are demanded in a commercially viable manner at market speed?

**A.** There must indeed be a shift in development mentality toward "You can't afford to fail" as we look at the life-preserving and product-enabling roles that software now plays. You won't get a second chance in life-critical environments, for instance. So, as you think about "can't afford to fail," what methodologies do you need to embrace to help make software failproof? Government agencies are advocating compliance models, but these fall short if they're an afterthought in the development life cycle for companies building products. Standards published by industry and governmental agencies in the form of compliance models must be baked in up front. Also, organizations should establish an integrated, complete engineering life-cycle management solution to help ensure that they're following an auditable process so that as compliance standards are incorporated, there's the ability to know what was done to support the regulatory structures in place.

As both customers and people experiencing these products, we expect them to perform flawlessly, and we demand that companies automate more and more of the operation of the products we use. Automobiles are a good example of products developed with a "can't afford to fail" methodology — how they've evolved from purely mechanical systems to electromechanical and, now, software-driven systems. Consider the evolution of self-driving automobiles, which are complex systems of systems that cannot afford to fail as they transport families down congested freeways at 65 miles per hour. As these products become more software intensive, and for IoT devices that are consuming data, safety-critical development processes are evolving to incorporate the breadth that needs to be encompassed as part of these kinds of systems. Organizations must have full transparency into the development process and produce up-to-the-minute status reports to make better decisions throughout the life cycle.

Occasionally organizations will look at what they've done after product completion and say, "Wait a minute. Is this compliant?" Then they backtrack to document their processes and make the product compliant after they've already built it. If compliance isn't incorporated into the product development model underlying the product from the start, organizations may need to rewrite or, at a minimum, do major revisions of the finished product. The engineering teams creating products need to adopt a complete and holistic view through the development cycle for software-intensive products to ensure they meet compliance and regulatory requirements. This can be accomplished only by compliance integration throughout the product life cycle.

## Q. Digital engineering is not new, but there seems to be a growing interest in the digital transformation of engineering in general. What's real, what's important, and where is the most promising activity?

**A.** Many companies have seen the benefits of digitally transforming their business processes. It has been a key focus for IDC customers and for much of the research we continue to do around digital innovation. Automating manual or repetitive tasks to reduce mistakes and freeing up people power for other, more meaningful work is a key aspect of enabling the creative engagement of teams when restructuring the business for digital transformation. The ROI benefits to business processes in this context are well-documented and essential to competitive advantage.

Engineering companies are no exception. These organizations must embrace digitally transforming their engineering processes as well in order to be competitive. As designs become more digital, they can be optimized faster and cheaper. This optimization enables fast learning cycles and faster convergence to more successful solutions because of the fluidity and adaptability.

Multidisciplinary teams across the business, across development and operations, help bring expertise to the development process. Other keys to success are communication and collaboration, along with effective change management and the use of adaptive, continuous quality processes.

The digital transformation of engineering can be enabled by leveraging and embracing engineering solutions that also enable and benefit from digital thread and digital twin functionality.

Successfully developing increasingly complex products that integrate hardware and software is almost impossible to do without the benefits from digital transformation. Digital transformation not only helps streamline the development of hardware and software systems but also provides a foundation for effective automation of selective tasks across the engineering life cycle. The value of the proposed transformation is magnified by adopting an integrated, end-to-end life-cycle model so that valuable engineering resources are not absorbed in building and maintaining internal tools. Your company's engineers should focus on developing products you sell. The solution for engineering life-cycle management should provide an end-to-end development model that delivers seamless process handoffs and integration as a core capability.

The consequence of not having these capabilities is grossly underestimated. Therefore, IDC strongly recommends looking for coordinated, integrated end-to-end engineering management, which can help eliminate wasted people power that's typically spent on maintaining custom code, building interfaces, and migrating data when it's not necessary. Plus, the data tends to be dated, in silos, and of poor quality. In short, engineering teams need to stop using nonengineering tools such as Word and Excel to manage requirements and projects.

We find that many organizations still use tools that don't allow them to access dynamic tracking, sharing, updating, and reporting of engineering processes. Engineering teams need to raise their product activity, quality, and compliance while improving time to market. Effective life-cycle management solutions for engineering can enable new levels of automation, transparency, compliance, and auditability.

It's also helpful to look at what these automated tools have done in the past to enable execution for organizations such as yours. Where you have specific kinds of challenges that you're trying to address, make certain that the automated solution you're looking for is flexible and adaptable to the needs of the digital environments in which you're working. You must live with the solution once it's integrated into your development process, so select one with longevity and scalability. That way the system will enable your organization to move forward with consistent product capability.

# About the Analyst

***Melinda-Carol Ballou,*** *Research Director, Agile ALM, Quality, and Portfolio Strategies*

Melinda-Carol Ballou serves as Research Director for IDC's Application Life-Cycle Management (ALM) program. In this role, Ms. Ballou provides thought leadership as well as expert opinion and analysis through comprehensive research on end-to-end application life-cycle management — from requirements to quality, testing, change, continuous release, process, and project and portfolio management (PPM) with a focus on agile DevOps software life-cycle strategies.

**IDC Corporate USA**

5 Speen Street
Framingham, MA 01701, USA

T 508.872.8200

F 508.935.4015

Twitter @IDC

idc-insights-community.com

www.idc.com

IDC Custom Solutions