

Warum ALM und PLM zusammen eingesetzt werden sollten

White Paper

In diesem White Paper werden die Unterschiede und Ähnlichkeiten zwischen Application Lifecycle Management (ALM) und Product Lifecycle Management (PLM) analysiert. Es wird untersucht, warum keine der beiden Lösungen die andere ersetzen kann und warum sie zusammen verwendet werden sollten.

www.siemens.com/plm

Inhalt

Kurzdarstellung	3
PLM und ALM – Definitionen der Systeme	4
Was ist PLM?	4
Was ist ALM?	4
Vergleich der Lebenszyklusphasen in ALM und PLM	4
ALM-Phasen	4
PLM-Phasen	5
Ähnlichkeiten und Unterschiede zwischen PLM und ALM	6
Vergleich zwischen Bauteilen und Dateien	6
Browserbasierte Zusammenarbeit	6
Lean Manufacturing und agile Softwareentwicklung	6
Vergleich zwischen PLM-Verfolgbarkeit und ALM-Verfolgbarkeit	6
Vergleich zwischen Versionskontrolle und Änderungs-Management	8
PLM und ALM: Ähnlichkeiten und Unterschiede	8
Wichtige Punkte	9
Software im Produktfertigungsprozess	10
Wichtige Punkte	10
Die Notwendigkeit, PLM und ALM zu verbinden	11
Zusammenfassung	12

Kurzdarstellung

Software überholt schnell den früheren Vorrang von Hardware im Produktentwicklungsprozess, vor allem für technologisch komplexe Produkte und Branchen wie Automobilindustrie, Luft- und Raumfahrt und Verteidigung sowie die Herstellung medizinischer Instrumente.

Technologiehersteller verwenden heute im Allgemeinen Lösungen für das Product Lifecycle Management (PLM), um Markteinführungszeiten zu verkürzen, die Prozesseffizienz zu erhöhen, die Zusammenarbeit zu verbessern und Compliance-Anforderungen zu erfüllen.

Herkömmliche PLM-Systeme verwalten jedoch Software im Allgemeinen als ein „Bauteil“ im Kontext des Produktfertigungsprozesses und haben Mühe, die komplexen Entwicklungsprozesse für Software zu verwalten. Software hat ihren eigenen, spezifischen Lebenszyklus, mit anderen zu verwaltenden Informationen, anderen Zusammenarbeitsprozessen und -methoden, anderen Spezifikationen und Elementen. Dieser Lebenszyklus wurde bisher durch herkömmliche PLM-Lösungen zu wenig berücksichtigt.

Um erfolgreich hochwertige softwaregesteuerte Produkte auf den Markt zu bringen, müssen Hersteller über ihr konventionelles eigenständiges PLM- oder ALM-System hinausgehen. Sie müssen aktiv nach einer integrierten Lösung suchen, die einen vollständigen Satz von Produkthanforderungen für Hardware und Software ermöglicht und die Zusammenarbeit mehrerer Disziplinen unterstützt, die notwendig ist, um ein End-to-End-Management der Software- und Hardwarekomponenten sicherzustellen.

PLM und ALM – Definitionen der Systeme

Was ist PLM?

PLM ist der Prozess, mit dem der gesamte Lebenszyklus eines Produkts von der Idee über die Konstruktion und die Herstellung bis hin zum Service und zur Stilllegung verwaltet wird. Das PLM integriert Personen, Daten, Prozesse und Unternehmenssysteme und bietet einen Ort, an dem Unternehmen und ihre Umgebung Produktinformationen gemeinsam nutzen können.

Mit PLM-Systemen können Unternehmen die wachsende Komplexität und die technischen Herausforderungen der Entwicklung neuer Produkte für globale, wettbewerbsorientierte Märkte bewältigen, indem sie jede Phase des Produktentwicklungsprozesses verkürzen und vereinfachen. Mit PLM-Lösungen können Unternehmen Produkte schneller auf den Markt bringen, immer strikere Compliance-Anforderungen und Branchenstandards erfüllen sowie mehr Zusammenarbeit und Kommunikation im gesamten Produktentwicklungsprozess erreichen.

Die Marktkategorie für PLM entstand im Jahr 1985 bei der American Motor Corporation (AMC) als eine Möglichkeit, den Produktentwicklungsprozess zu beschleunigen und am Automobilmarkt wettbewerbsfähiger zu werden.









PLM wird in der Automobilbranche weiterhin intensiv genutzt, aber auch viele andere Branchen wie Luft- und Raumfahrt, Gesundheitswesen, medizinische Instrumente, Prozessfertigung und Energie haben die Methode inzwischen eingeführt.

Was ist ALM?

„Das Application Lifecycle Management (ALM) ist kein Produkt, sondern ein Prozess“, erklärt der Branchenanalyst Ovum in seinem Bericht zum Software Lifecycle Management. Ovum definiert ALM dann als den Prozess, mit dem Unternehmen in Informationstechnologie (IT) und Softwareentwicklung ihre Software während des gesamten Lebenszyklus erstellen, bereitstellen und betreiben.¹

Vergleich der Lebenszyklusphasen in ALM und PLM

ALM-Phasen





- | | | |
|--|---|--|
| <p>1  Projekt- und Portfoliomanagement für Anwendungen
Vor Beginn eines Softwareprojekts wird eine Investitionsanalyse durchgeführt und ein Business-Case entwickelt.</p> | <p>2  Projektbeginn und Sammlung von Anforderungen
Informationen zum Markt werden gesammelt, potenzielle Anwender/Kunden der Anwendung werden befragt, und Daten werden gesammelt, um dokumentierte Anforderungen zu erhalten.</p> | <p>3  Anforderungsmanagement
Wenn Anforderungen sich weiterentwickeln oder verändern, muss auch das Anforderungsdokument weiterentwickelt werden, um die Auswirkungen auf Entwicklungszeitpläne, Bereitstellungstermin, Ressourcen usw. zu analysieren.</p> |
| <p>4  Entwurf und Anwendungsfall-Analyse
Die zugrunde liegende Architektur des Softwarecodes wird definiert, und verschiedene Anwendungsfälle werden entwickelt, um die möglichen Interaktionen von Anwendern mit dem fertigen System zu modellieren.</p> | <p>5  Codierung
Anwendungscode wird geschrieben oder im Fall eines Updates erweitert oder verändert.</p> | <p>6  Testing und Qualitätssicherung
Die Software wird einem systematischen Debugging sowie Performance-, Last- und Stresstests unterzogen, und der Code wird nach Bedarf verändert.</p> |
| <p>7  Release und Bereitstellung des Builds
Das abschließende Release wird erstellt und finalisiert, und die Anwendung wird in der Produktion bereitgestellt.</p> | <p>8  Anwendungsperformance
Die fortlaufende Wartung der Anwendung umfasst Erweiterungen und Fehlerbehebung während des gesamten Anwendungslebenszyklus bis zur Stilllegung.</p> | |



Wichtige Punkte

Es ist zwar möglich, Software gemäß dem Wasserfallmodell zu entwickeln; die verbreitete Akzeptanz der Agile-Entwicklungsmethodik bedeutet jedoch, dass Software heute stärker iterativ entwickelt wird: in kurzen, schnellen „Sprints“, wobei Anforderungen häufig geändert werden und viele fortlaufende Veränderungen vorgenommen werden.

PLM-Phasen

- | | | |
|--|--|---|
| <p>1  Idee
Informationen werden am Markt gesammelt, Kundenanforderungen werden ermittelt, eine Vorstellung des Produkts wird entwickelt, und technische Spezifikationen werden anhand dieser Informationen erstellt.</p> | <p>2  Konstruktion
Die Konstruktion des Produkts wird zunächst entworfen, dann verfeinert, getestet und mit Werkzeugen wie CAD- und CAE-Analysen überprüft. In diesen Schritt wird eine Reihe von Engineering-Disziplinen einbezogen, darunter Mechanik, Elektrik, Elektronik, Software (eingebettet) und Simulation, sowie bereichsspezifisches Fachwissen wie beispielsweise Automobil-Engineering.</p> | <p>3  Realisierung
In dieser Phase ist die Produktkonstruktion abgeschlossen, und die Herstellungsmethode wurde festgelegt. Nun werden Werkzeugentwurf, Analysen, Simulation und ergonomische Analysen durchgeführt.</p> |
| <p>4  Service und Stilllegung
Diese letzte Phase des Produktlebenszyklus ist die Servicephase. Hierzu können Reparaturen und Wartung, Abfallmanagement und die Stilllegung (Entsorgung, Zerstörung) des Produkts gehören.</p> | | |



Wichtige Punkte

PLM-Prozesse folgen im Allgemeinen dem Wasserfallmodell mit iterativen Arbeitsmustern, die inkrementelle Änderungen innerhalb des Prozesses berücksichtigen.

Ähnlichkeiten und Unterschiede zwischen PLM und ALM

Um einen effektiven Business-Case für die Integration von PLM und ALM zu entwerfen, ist es wichtig, die Ähnlichkeiten und Unterschiede zwischen den beiden Systemen zu verstehen. Mithilfe dieses Vergleichs können wir die jeweilige grundlegende Orientierung dieser Systeme besser verstehen, ermitteln, wie die Systeme voneinander profitieren können, und die wichtigsten Punkte für eine wirklich effektive Integration der beiden Systeme identifizieren.

Vergleich zwischen Bauteilen und Dateien

Der erste wesentliche Unterschied liegt darin, dass ein PLM-System auf das Management von „Bauteilen“ ausgerichtet ist, mit denen ein physisches Produkt hergestellt wird und denen Anforderungen, Konstruktionsattribute und Änderungen zugeordnet sind. Ein ALM-System hingegen ist auf das Management von Softwaredateien/-elementen ausgerichtet, wie Anforderungsdokumenten, Softwarecode oder Testfällen sowie auftretenden Änderungen an diesen Dateien.

Browserbasierte Zusammenarbeit

Die Softwareerstellung und die ALM-Werkzeuge haben sich bisher im Allgemeinen in der Software-Entwicklungsabteilung befunden. In den letzten Jahren sind ALM-Daten jedoch auch für immer mehr andere Beteiligte wertvoll geworden. ALM deckt heute auch andere Disziplinen neben der Softwareentwicklung ab. Eine höhere Anzahl Beteiligter bedeutet, dass leichter Echtzeit-Zugriff auf Daten immer wichtiger wird, im Allgemeinen über eine Browser-Benutzeroberfläche. Immer mehr ALM-Anbieter stellen heute browserbasierten Zugriff auf ihre Systeme bereit, und moderne Web 2.0-Produkte wie Polarion® ALM sind von Haus aus webbasiert und bieten Zugriff auf die Cloud sowie über mobile Geräte.

Die webbasierte Zusammenarbeit zwischen Teammitgliedern ist jedoch eine wesentliche Hürde für Anbieter herkömmlicher PLM-Systeme, denen es nur gelungen ist, Teile ihrer Lösungen über das Internet zugänglich zu machen. Mit dem browserbasierten Ansatz von Polarion ALM können Anwender eine detaillierte 3D-Darstellung und Stückliste (Bill of Materials, BOM) erstellen, einschließlich der eingebetteten Software mit Verfolgbarkeit beispielsweise bis zum Getriebe eines Fahrzeugs, das in China angezeigt wird und von einem

Server in Stuttgart stammt. Ohne die browserbasierte Integration sowie die Interoperabilität von ALM und PLM, die von Polarion ALM bereitgestellt werden, ist eine derartige Funktionalität nur sehr schwer zu erreichen.

Lean Manufacturing und agile Softwareentwicklung

In den 1990er Jahren führte das Toyota Production System das Konzept des Lean Manufacturing als Managementphilosophie zur Verbesserung des Kundennutzens ein. Toyota konnte dank dieser Philosophie von einem kleinen Unternehmen zum größten Autohersteller der Welt aufsteigen.

Kurz gesagt gehört zu Lean jede Produktionspraxis, mit der Wert für den Kunden geschaffen wird, also etwas, wofür der Kunde zu zahlen bereit ist. Alles andere wandert in den Papierkorb.

Man kann argumentieren, dass die Agile-Methodik der Softwareentwicklung zumindest in philosophischer Hinsicht vom Lean Manufacturing abgeleitet wurde: Mit agilen Methoden stehen die Anwender im Zentrum der Entwicklung und definieren, was implementierungswürdig ist.

Es ist für alle Hersteller wichtig, ALM- und PLM-Systeme zu integrieren, um Produkt- und Software-Entwicklungsmethoden zu integrieren. Es ist eine natürliche Entwicklung, dass das Lean Manufacturing und die Agile-Entwicklung sich zu einem einzigartigen Ansatz für die Anwendungs- und Produktentwicklung vereinen.

Vergleich zwischen PLM-Verfolgbarkeit und ALM-Verfolgbarkeit

Ein weiterer Unterschied zwischen PLM und ALM liegt darin, wie die Verfolgbarkeit definiert wird. In einem PLM-System wird die Verfolgbarkeit definiert als die Zerlegung eines vollständigen Systems in seine Bauteile. Ein Auto besteht aus einem Fahrzeugrahmen, einer Achse, vier Reifen und so weiter. In einem ALM-System wird die Verfolgbarkeit definiert als die Links zwischen Dateien/Elementen, die zu unterschiedlichen Phasen gehören. Eine Änderung an einer Anforderung kann sich auf eine Codezeile auswirken oder die Entwicklung eines neuen Testfalls erfordern, mit dem die neue Anforderung überprüft werden kann.

Ein PLM-System setzt Informationen mit PLM-Elementen in Beziehung und verknüpft sie mit ihnen, etwa mit Anforderungen, Konstruktionsobjekten, Materialien, Toleranzen, Änderungen usw. Ein ALM-System hingegen setzt Informationen mit Softwarecode in Beziehung und verknüpft sie mit ihm, etwa mit Anforderungen, Änderungsanfragen, Testfällen und Commit-Befehlen.

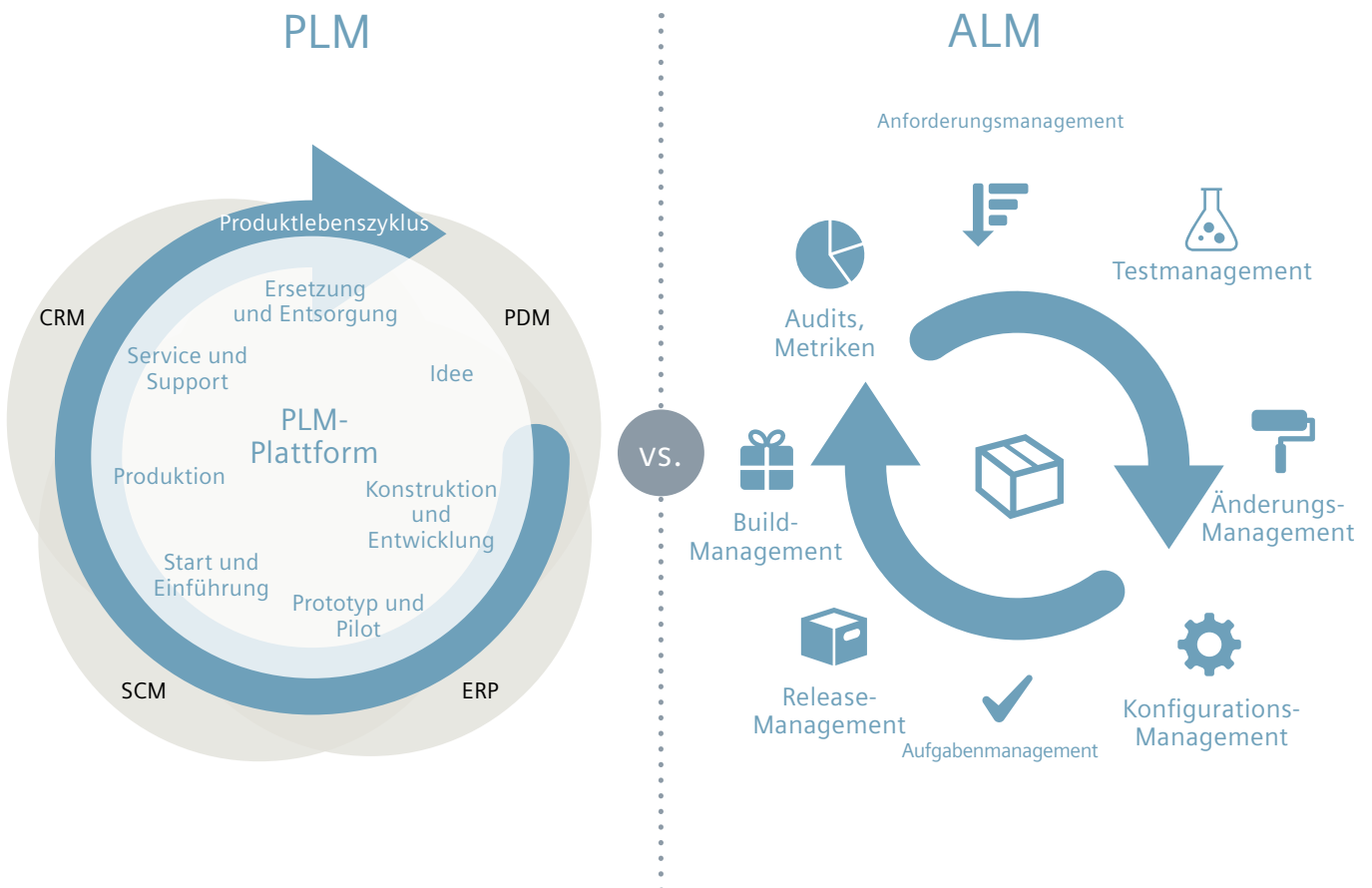
Die Existenz von Software als „Bauteil“ eines vollständigen Systems ist eine wesentliche Motivation dafür, PLM und ALM zu vereinen. Software kommt im herkömmlichen PLM als ein einzelnes Bauteil vor, aber mehr Management ist nicht vorgesehen. Das Softwaremanagement herkömmlicher PLM-Systeme gelangt nicht bis an den Punkt, an dem dieses „Bauteil“ seinen eigenen Lebenszyklus aufweist, zu dem auch zahlreiche Dateien/Elemente und Änderungen an diesen Dateien/Elementen gehören würden.

Das Fehlen eines Lifecycle Management für das „Software-Bauteil“ führt zunehmend zu Problemen für Produkthersteller. Qualitätsprobleme bei Software sind die eigentliche Ursache vieler teurer Misserfolge bei Produkten und der Grund eines hohen Prozentsatzes der Produktrückrufe. Herkömmlichen Ansätzen für PLM und sogar für die Integration von PLM und ALM fehlt jedoch häufig die Möglichkeit, alle Beziehungen zwischen Hardware und Software zu identifizieren, um die Ursachen von Produktfehlern und Softwareproblemen zu ermitteln.

Betrachten Sie als Beispiel eine Situation in der Automobilbranche, in der in Release 3.4.5 einer Softwarekomponente beim Wert 3,5 für Parameter X ein falsches Signal erzeugt wird. Diese Softwarekomponente wird in mehreren Fahrzeugen verwendet. Wie findet ein Hersteller also heraus, bei welchen Fahrzeugen dieses Problem auftritt und behoben werden muss?

In einem aktuellen Blog-Beitrag beschreibt Tom Grant, Analyst von Forrester Research für die Anwendungsentwicklung und -bereitstellung, die Kluft, die zurzeit zwischen vielen PLM- und ALM-Systemen besteht, und argumentiert, warum PLM- und ALM-Anbieter diese Lücke schließen sollten:

„Unternehmensprozesse wie die Formulierung von Anforderungen, die sich sowohl auf Hardware- als auch auf Softwarekomponenten beziehen, sind ein Grund, aus dem ALM und PLM miteinander integriert werden müssen. Während Produktteams bereits hierzu in der Lage sind (indem sie beispielsweise die Anforderungen als „Systeme von Systemen“ formulieren), sind ihre Werkzeuge hierfür nicht immer geeignet. PLM-Werkzeuge, die theoretisch sowohl Hardware als auch Software einbeziehen sollen, sind im Allgemeinen für den Umgang mit dem digitalen Teil des Produkts nur unzureichend geeignet. Einige Elemente von ALM, wie die Quellcodeverwaltung, gibt es in PLM sogar gar nicht.“²



Vergleich zwischen Versionskontrolle und Änderungs-Management

Herkömmliche PLM-Systeme umfassen eine Art Versionskontrolle – sie berücksichtigen, dass ein Bauteil sich im Laufe der Zeit verändern kann – aber sie zeichnen im Allgemeinen nicht den Grund für eine solche Veränderungen auf und können ihn nicht abrufen. Da viele Systeme auch nicht wie ALM auf Verfolgbarkeit abzielen, können sie nicht identifizieren, welche Auswirkungen eine Änderung an einer Software auf alle anderen verbundenen Elementen im System hat.

PLM und ALM: Ähnlichkeiten und Unterschiede

Ähnlichkeiten	Unterschiede
Beide Systeme beziehen sich auf Prozesse und Kerndisziplinen.	<p>ALM ist auf „Softwaredateien“ hin orientiert und schreibt einen Prozess für die Erstellung von Softwareanwendungen vor. Die Anwendungen bestehen aus mehreren Elementtypen und komplexen Beziehungen zwischen <i>Software</i>- Elementtypen, die Auswirkungsstrukturen erzeugen.</p> <p>PLM ist auf „Bauteile“ hin orientiert, zwischen denen eine Struktur von „ist Teil von“-Beziehungen besteht. Die Elementtypen und komplexen Beziehungen zwischen diesen Arten von Hardwareelementen erzeugen Auswirkungsstrukturen.</p>
Beide Systeme umfassen Workflow, Variantenmanagement, Testmanagement, Anforderungen und Spezifikationsmanagement.	<p>Bei ALM geht es um das Abstrakte. Bei PLM geht es um das Konkrete. In ALM können Software-Engineers abstrakte Funktionen planen, untersuchen, definieren, implementieren, testen und pflegen.</p> <p>Bei PLM geht es um die Bereitstellung einer vollständigen Stückliste für die Produktionskette zusammen mit einem Verständnis der Produktkonfiguration. Die Funktion der Komponenten in PLM ist das Produkt selbst.</p>
Beide Systeme ermöglichen die gegenseitige Verknüpfung von Komponenten.	<p>ALM umfasst zahlreiche unterschiedliche Arten von Beziehungen und Verknüpfungen, die Abhängigkeitshierarchien zwischen Softwareelementen bilden.</p> <p>PLM umfasst zahlreiche unterschiedliche Beziehungen und Verknüpfungsarten zwischen Hardwareelementen, die Abhängigkeiten und Enthaltenshierarchien bilden.</p>
Beide Systeme ermöglichen das Verknüpfen von Informationen mit Komponenten.	<p>In PLM sind diese Informationen im Allgemeinen quantitativ. Sie können Anforderungen, Ziele, Konstruktionsobjekte, Materialien, Toleranzen usw. umfassen.</p> <p>In ALM sind die mit Elementen verknüpften Informationen beschreibend: Texte, Modelle, User Stories, Testszenarien usw.</p>
In beiden Umgebungen werden Modelle häufig eingesetzt.	<p>In PLM zeigen Modelle die Zerlegung in Bauteile und definieren Elemente der Produktentwicklung. PLM-Modelle sind häufig in unterschiedliche Produkt-Subsysteme segmentiert: elektrische Auslegung, Brems-Subsystem, Getriebe, Innenausstattung usw.</p> <p>In ALM zeigt ein Modell die funktionale Zerlegung mithilfe von Diagrammen wie Entity-Relationship-Darstellungen oder objektorientierten Darstellungen.</p>

Ganz grundlegend formuliert sind PLM- und ALM-Systeme unterschiedlich, aber sie ergänzen einander sehr gut. Sie wurden zu unterschiedlichen Zeitpunkten entworfen, um sehr unterschiedliche Arten von Informationen zu verwalten. Sie sind für unterschiedliche Prozesse und unterschiedliche Arten von Anwendern bestimmt. Es ist nun an der Zeit, PLM und ALM zu vereinen. Software spielt eine so große Rolle in heutigen technologisch fortschrittlichen Produkten, und die Risiken, die entstehen, wenn diese Software nicht verwaltet wird, sind so groß, dass es unverantwortlich wäre, diese Systeme weiterhin in separaten Silos zu belassen.

Wichtige Punkte

- Software spielt heute eine kritische Rolle in Produkten und ist häufig die Ursache für ihren Ausfall. Im Jahr 1999 führte ein Softwarefehler im NASA Mars Climate Orbiter dazu, dass das 125 Millionen US-Dollar teure Raumschiff – das Herzstück des Mars-Erkundungsprogramms – zu niedrig und zu schnell in die Atmosphäre des Mars eintrat. Danach hat nie wieder jemand etwas von dem Raumschiff gehört.
- PLM- und ALM-Systeme wurden für die Verwaltung sehr unterschiedlicher Dinge entwickelt und bieten ihren Anwendern sehr unterschiedliche Funktionen. Wenn ein Hersteller ein Produkt mit einem hohen Softwareanteil erstellt, werden sowohl PLM als auch ALM benötigt, und sie *müssen* miteinander zusammenarbeiten.
- In PLM- und ALM-Systemen ist die Verfolgbarkeit unterschiedlich definiert. In PLM wird unter Verfolgbarkeit die Zerlegung eines Produkts in unterschiedliche Bauteile und Komponenten während des gesamten Lebenszyklus verstanden. In ALM bezeichnet Verfolgbarkeit die Verknüpfungen zwischen Elementen in den unterschiedlichen Phasen des Softwareentwicklungsprozesses. Auch hier werden sowohl PLM als auch ALM benötigt, und sie *müssen* miteinander integriert sein.

Software im Produktfertigungsprozess

Heute müssen Dutzende Mikroprozessoren 100 Millionen Codezeilen ausführen, um ein hochwertiges Auto aus der Garageneinfahrt zu bewegen. Und diese Software wird in Zukunft nur noch komplexer werden.³

Das Software-Engineering wird zunehmend zur stärksten Kraft bei der Herstellung von Produkten für Verbraucher und für Unternehmen. Siemens beschäftigt heute in seinem High-Tech-Business mehr Software-Engineers als Microsoft, Oracle oder SAP.

Die wachsende und kritische Wichtigkeit von Software in Produkten bedeutet neue, zusätzliche Komplexität für den Produktentwicklungsprozess. Hersteller sind nicht mehr nur für die Hardware verantwortlich. Sie müssen heute auch zusätzliche Prozesse und Verfahren für die Entwicklung komplexer, eingebetteter Softwaresysteme entwerfen. Da die Qualität und die Performance dieser Software unter einigen Umständen (etwa bei einem medizinischen Instrument oder bei der Steuerung eines Flugzeugs oder Fahrzeugs) über Leben und Tod des Anwenders entscheiden können, wird der Softwareprozess von Behörden strikt kontrolliert, und die Vorschrifteneinhaltung wird überwacht. Wenn diese Standards nicht eingehalten werden, können schwere Strafen oder die Stilllegung des Fertigungsbetriebs die Folge sein.⁴

„Die Wichtigkeit der systemorientierten Produktentwicklung und des Systems Engineering steigt weiter, da Hersteller zunehmend Software integrieren, um Produktfunktionen bereitzustellen. Der Übergang von einer geringen Koordination der mechanischen und elektronischen Konstruktion sowie der Softwareentwicklung zu einem strikter umgesetzten systemorientierten Ansatz erfordert zeitaufwendige Änderungen an der Entwicklung neuer Produkte, an Prozessen, Organisation und Unternehmenskultur. Hersteller, die diesen Übergang nicht früh genug durchführen, werden Schwierigkeiten haben, im Wettbewerb zu bestehen.“

Gartner – Marc Halpern, Janet Suleski: *Predicts 2013: Product Design and Life Cycle Management*. 30. November 2012

Wichtige Punkte

- Das Kampfflugzeug F-35 aus dem Joint Strike Fighter-Programm der US Air Force enthält ca. 5,7 Millionen Codezeilen.
- Ein durchschnittliches medizinisches Instrument enthält heute eine Million Codezeilen, und diese Anzahl verdoppelt sich alle paar Jahre.
- Software ist ein zunehmend dominanter Innovationsfaktor bei industriellen Produkten.

Die Notwendigkeit, PLM und ALM zu verbinden

Weiter oben in diesem White Paper haben wir über den zunehmenden Umfang von Software in hergestellten Produkten gesprochen – insbesondere in technologisch komplexen Produkten wie Autos, medizinischen Instrumenten und Flugzeugen.

Diese zunehmende – und zunehmend dominante – Präsenz von Software fügt dem Produkt-Engineering- und -Fertigungsprozess eine neue Dimension von Komplexität hinzu. Diese höhere Komplexität muss verwaltet werden.

Oberflächlich betrachtet scheinen die Methoden für die Lebenszyklus-Verwaltung für Produkte (PLM) und für Software (ALM) einander ziemlich ähnlich zu sein. Sowohl PLM- als auch ALM-Systeme basieren auf einem integrierten Prozess und einer Reihe von Kerndisziplinen. Damit ist die Ähnlichkeit aber auch schon zu Ende.

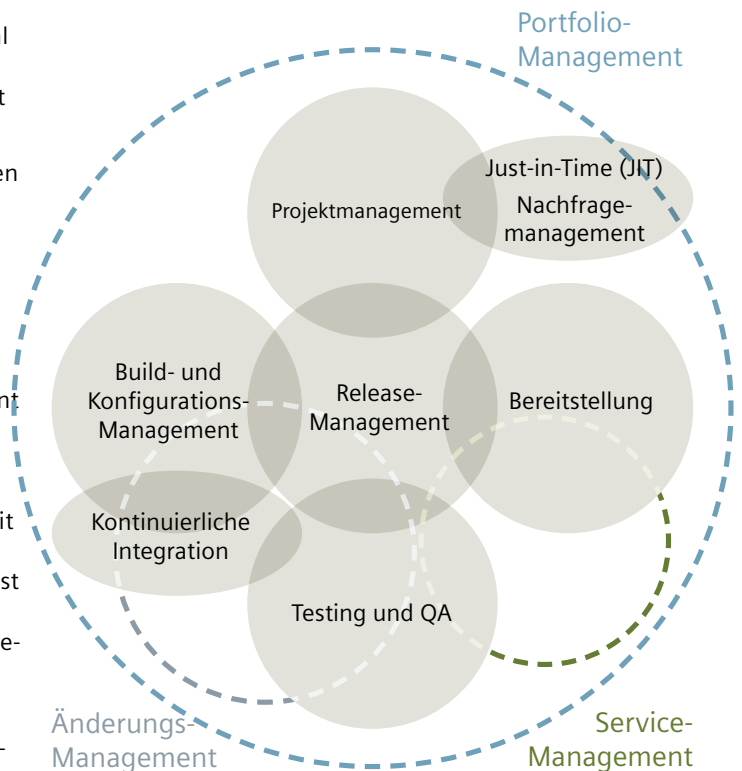
Einige CIOs und IT-Manager mit wesentlichen Investitionen in PLM-Systeme haben versucht, ein PLM-System für die Verwaltung von Software zu verwenden. Mit einem PLM-System können produktbezogene Workflows, Spezifikationen, Konstruktionen und Versionen verwaltet werden – also warum dann nicht auch Software?

Von einem PLM-System zu verlangen, dass es für die Verwaltung der Komplexität und für das Dateimanagement des Softwareentwicklungsprozesses geeignet ist – einschließlich iterativer Entwicklungszyklen, veränderlicher Anforderungen, Verfolgbarkeit von Elementen, Beziehungen zwischen Elementen usw. – geht jedoch weit über das hinaus, was viele PLM-Systeme leisten können. Für die Verwaltung von Softwareentwicklungsprozessen ist ALM besser geeignet, eine softwareorientierte Disziplin, die spezifisch auf die oben beschriebenen Aufgaben ausgerichtet ist.

Nachdem wir uns die zahlreichen Unterschiede zwischen den Ansätzen und Werkzeugen von ALM und PLM angesehen haben, können wir festhalten:

1. Es ist heute dringender als je zuvor, Software- und Produktlebenszyklen auf integrierte Weise zu verwalten.
2. Herkömmliche PLM-Werkzeuge sind für die Verwaltung der Softwareentwicklung nicht gut geeignet.
3. ALM-Werkzeuge sind für die Verwaltung der Produktentwicklung nicht gut geeignet.

Daher bleibt die Frage: Wie kann ein Hersteller den wachsenden Softwareumfang am besten verwalten, der heute als Komponente in einem Bauteil in einem Produkt enthalten ist? Im Idealfall wird dieses Problem durch Integration und Interoperabilität zwischen einem PLM-System und einem ALM-System gelöst.



Quelle: The Forrester Wave™: Application Life-Cycle Management, Forrester Research, Inc.

Zusammenfassung

Software läuft der ehemals dominanten Hardware bei der Produktentwicklung zunehmend den Rang ab. Dies gilt insbesondere für technologisch fortschrittliche Produkte (wie Automobile, Flugzeuge, medizinische Instrumente und Smartphones).

System- und Produkt-Engineers müssen aktiv nach Werkzeugen suchen, die über ihr herkömmliches PLM-System hinausgehen und eine Zusammenarbeit zwischen mehreren Disziplinen ermöglichen – vor allem mit ihren Kollegen im Software-Engineering. Diese Werkzeuge sollten ein End-to-End-Management von Software- und Hardwarekomponenten sicherstellen.

Da die Software- und die Produktentwicklung sehr unterschiedliche Disziplinen sind, gibt es keine Möglichkeit, beim Systems Engineering nur PLM oder nur ALM zu verwenden. Hersteller müssen beide verwenden. Daher

müssen PLM und ALM integriert werden, damit alle Disziplinen und alle Entwurfs- und Konstruktionsprozesse Produkt- und Softwareanforderungen gemeinsam verwenden und miteinander verknüpfen können. So können sie enger zusammenarbeiten, indem sie die bereichsübergreifenden Beziehungen aufbauen, die sie benötigen, um die Auswirkungen von Änderungen vollständig einzuschätzen und um auf diese Informationen mit den Werkzeugen zuzugreifen, die ihnen vertraut sind. Erfolgreiche Hersteller verlangen die Art von Integration und Interoperabilität, die in Teamcenter und Polarion ALM vorhanden sind.

Referenzen

1. Ovum, Software Lifecycle Management 2011.
2. Forrester Research, Inc., Blog-Beitrag, August 2012 – „ALM and PLM: Make it Work, People“, Tom Grant.
3. This Car Runs on Code – IEEE Spectrum, Februar 2009.
4. Getting Better Software into Manufactured Products – McKinsey Quarterly, März 2006.

Siemens PLM Software

Hauptsitz

Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Amerika

Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europa

Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asien-Pazifik

Suites 4301-4302, 43/F
AIA Kowloon Tower, Landmark East
100 How Ming Street
Kwun Tong, Kowloon
Hong Kong
+852 2230 3308

Über Siemens PLM Software

Siemens PLM Software, eine Business Unit der Siemens Digital Factory Division, ist ein führender, weltweit tätiger Anbieter von Software, Systemen und Dienstleistungen für das Product Lifecycle Management (PLM) und das Fertigungsmanagement (MOM) mit über neun Millionen lizenzierten Anwendern und mehr als 77.000 Kunden in aller Welt. Siemens PLM Software mit Sitz in Plano, Texas, entwickelt in enger Zusammenarbeit mit seinen Kunden branchenspezifische Softwarelösungen, die Unternehmen in allen Bereichen durch Umsetzung bedeutender Innovationen einen nachhaltigen Wettbewerbsvorteil verschaffen. Weitere Informationen über die Produkte und Leistungen von Siemens PLM Software unter www.siemens.com/plm.

Weitere Informationen über Polarion ALM finden Sie unter www.polarion.com.

www.siemens.com/plm

© 2016 Siemens Product Lifecycle Management Software Inc. Siemens und das Siemens-Logo sind eingetragene Marken der Siemens AG. ALM, D-Cubed, Femap, Fibersim, Geolus, GO PLM, I-deas, JT, NX, Parasolid, Polarion, Solid Edge, Syncrofit, Teamcenter und Tecnomatix sind Marken oder eingetragene Marken der Siemens Product Lifecycle Management Software Inc. oder ihrer Niederlassungen in den USA und in anderen Ländern. Alle anderen Logos, Marken, eingetragenen Marken oder Dienstleistungsmarken sind Eigentum der jeweiligen Inhaber.

55633-Y6 8/16 o2e