



Geordnete Verhältnisse bevorzugt

Autor: Jan Stürtz, Contact Software GmbH, Bremen

Ob Haushaltsgerät, Fahrzeug oder Fertigungsanlage: Viele Produkte stecken heute voller Elektronik und entsprechender Steuerungssoftware. Deren Programmierung erfolgt jedoch immer noch losgelöst vom eigentlichen Produktentstehungsprozess, was ein ganzheitliches Product Lifecycle Management (PLM) deutlich erschwert. Beispiele hierfür sind die Qualitätssicherung, Produktdokumentation und Wartung bis hin zu Fragen der Haftung und Gewährleistung. Eine Lösung ist die Integration von Softwarekomponenten in die PLM-Strategie durch die Einbindung des Software Configuration Managements in das allgemeine Produktdatenmanagement (PDM).

Nicht nur Hightech-Systeme wie ABS oder EPS im Fahrzeug werden über Software gesteuert, auch ein Fensterheber mit Kindersicherung oder eine U-Bahn-Tür funktionieren heute nicht mehr ohne die Binärprogramme. Dahinter verbirgt sich ein komplexes Geflecht von mechanischen, elektrischen und elektronischen Bauteilen sowie verschiedenen Softwaremodulen, die beispielsweise die Befehle zum Fensteröffnen über das inter-

ne Bussystem an den Steuermotor weiterleiten oder durch die Überwachung von Lichtschranke und anderer Sensoren verhindern, dass sich jemand an einer schließenden U-Bahn-Tür verletzt.

Viele Unternehmen nutzen bereits eine Produktdatenmanagement-Lösung (PDM) als zentrale Plattform bei der Entwicklung ihrer »Hardware«, sprich: Produkte. Moderne PDM/PLM-Systeme bilden den gesamten Lebenszyklus ab, un-

terstützen aber mit der Verwaltung mechatronischer Produktstrukturen (ECAD/MCAD), Dokumenten-, Rechte-, und Änderungs- und Projektmanagement vor allem die interdisziplinäre Zusammenarbeit am virtuellen Produktmodell.

Software hingegen wird oft noch ausschließlich auf dem Arbeitsplatzrechner oder auf Projektlaufwerken entwickelt. Manche Unternehmen setzen aber auch Versionskontrollsysteme wie Sub-

version oder CVS (frei verfügbar) beziehungsweise Rational ClearCase, Serena PVCS oder Perforce (kommerzielle Systeme) ein. Software Configuration Management (SCM) wird schwerpunktmäßig zur Verwaltung und Kontrolle von Softwareprojekten und zur Synchronisation der Arbeit von mehreren Entwicklern in einem Softwareprojekt eingesetzt, konzentriert sich also in erster Linie auf den reinen Software-Entwicklungsprozess. SCM kann aber auch den gesamten Lebenszyklus eines Softwareproduktes unterstützen, was insbesondere im Bereich der Software-Wartung genutzt wird. SCM-Tools verwalten Quellcode-Dateien und Module sowie alle Änderungen

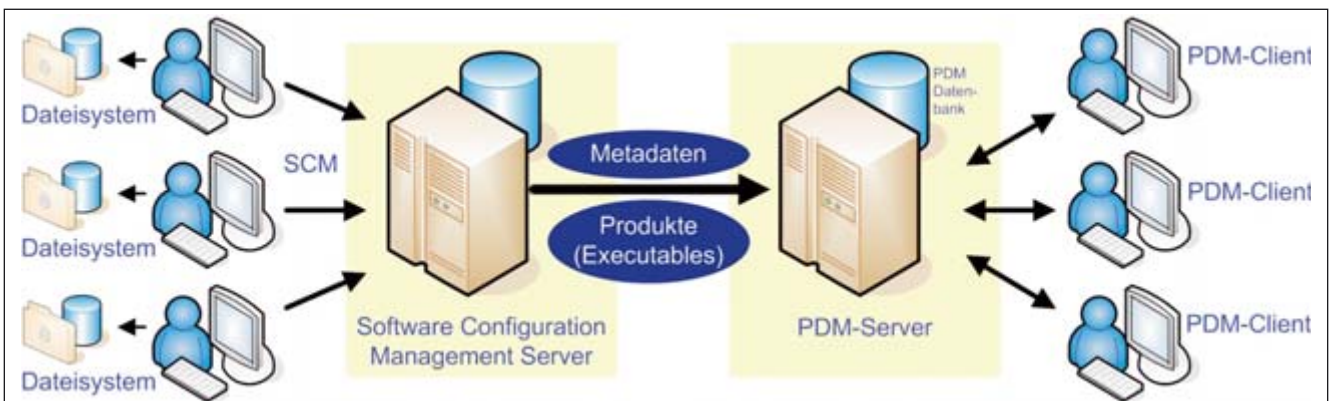
nun mehr Platz ein, sodass jetzt ein anderer Chip mit mehr Kapazität verbaut werden muss. Der ECAD-Entwickler kann das Layout jedoch nur entsprechend anpassen, wenn er rechtzeitig über diese Änderung informiert wurde.

Die fehlende Integration auf IT-Ebene verursacht nicht nur mehr Zeitaufwand bei der Abstimmung zwischen Software- und Elektronik-Entwicklung, sondern ist auch mit Risiken behaftet: Das anhand der Spezifikation erzeugte Programm (Binary) wird an den Elektro-Entwickler per Mail geschickt und dann auf den Chip gebrannt. Die Quellen werden lokal oder auf einem Projektlaufwerk – unter Umständen unterstützt durch ein

men und mitindiziert. Jetzt hat man zwar dokumentiert, welches Binary wann verbaut worden ist, aber es bleiben weiterhin viele Fragen offen: Wie ist das Binary entstanden? Welche Dateien in welcher Version sind »verbaut« worden? Welche Änderungswünsche sind in die neue Softwareversion eingeflossen?

Engineering Change Management

Doch genau diese Antworten sind für die Qualitätssicherung und eine vollständige Produktdokumentation unerlässlich: Sei es in Bezug auf gesetzliche Nachweispflichten, zur Durchführung von



Lose Integration: Bei dieser Variante entfällt die Notwendigkeit für ein gemeinsames Datenmodell zwischen Product Data Management System und dem Software Configuration Management.

daran im Produktlebenszyklus – inklusive der Steuerung des gesamten Build-Prozesses zur Erzeugung der Software. SCM ist also mehr als nur die reine Versionskontrolle.

Zielen Hard- und Software-Entwicklungsprozesse auf ein gemeinsames Endprodukt ab, erschweren voneinander entkoppelte IT-Tools ihre Synchronisation.

Mangelnde Transparenz

Auch Softwarekomponenten müssen als integraler Produktbestandteil in der Produktstruktur im PDM-System wiederzufinden sein, wie ein einfaches Beispiel zeigt: Auf den bisher verbauten Chip passen zwei Kilobyte Code. Durch eine Änderung nimmt der Code

Versionskontrollsystem – verwaltet. Änderungswünsche des ECAD-Entwicklers gehen anschließend via Telefon/E-Mail an den Entwickler zurück, der wiederum ein neues angepasstes Binary erzeugt, es dem ECAD-Entwickler zurück schickt und so weiter. Wenn überhaupt, werden Änderungswünsche nur im SCM-System dokumentiert. Versionen des Binaries, die vielleicht schon ausgeliefert wurden, existieren maximal bei den beteiligten Entwicklern lokal im Dateisystem. Es ist nicht nachvollziehbar, in welcher Produktversion welche Software verbaut wurde.

Als »Workaround« werden erzeugte Binaries oder auch ein Zip-File mit allen Quellen als Dokument in die Produktstruktur aufgenommen

Wartungs- und Instandsetzungsarbeiten oder in Haftungs- und Schadensfällen.

Alle Änderungen an einem Produkt müssen einem Änderungsprozess zugeordnet werden, auch die Software-Änderungen im SCM-System. Eine Kontrolle des Änderungsprozesses unterstützen oft nur zusätzliche Tools wie beispielsweise PVCS Tracker, Visual Intercept, Rational Clear Case, Bugzilla oder ähnliche selbst geschriebene Lösungen.

Oft wird das Anforderungsmanagement für Hard- und Softwareprodukte außerhalb des PDM-Systems, teilweise sogar in unterschiedlichen Systemen für die jeweiligen Disziplinen, gepflegt. Dies ist aber ebenso wie das Trouble

Management und das Verwalten der Change Requests Bestandteil des Produktlebenszyklus (PLC).

Bei einem ganzheitlichen PLC, bei dem Soft- und Hardware zu einem Produkt gehören, stellt sich die Frage, in welchem System ein Change Request erfasst wird. Und was ist mit Change Requests, die sowohl Software als auch Hardware betreffen? Meistens liegt der entsprechende Change Request schon im PDM-System vor. Eine Integration des SCM-Systems mit dem PDM ist also mehr als wünschenswert; Änderungen im SCM ohne Angabe eines Change Request dürfen dann de facto nicht mehr vorkommen und werden vom SCM abgelehnt. Umgekehrt kann man nun bei jeder Änderung einer Datei im Versionskontroll-

denen Systeme quasi »von Hand« bewältigen müssen. Es bleibt also nur das PDM-System, um den PLC durchgängig zu erfassen und zu dokumentieren.

Produktdaten-Management und Software Configuration Management haben viel gemeinsam: Beispiele sind das Change- und Release-Management, die Verwaltung von Metadaten, das Configuration Management, Attribute auf verwalteten Daten und die Abbildung von Produktstrukturen.

Integrationsmöglichkeiten der Systeme

Dennoch werden Anwender in den jeweiligen Disziplinen immer vorrangig entweder das SCM- oder das PDM-System ein-

SCM-System ausreichende und stabile Schnittstellen bietet und ein gemeinsames Datenmodell für PDM und SCM etabliert wird, was für PDM-Hersteller wie Anwender bestehender Tools eine große Herausforderung darstellen würde.

Die zweite Möglichkeit ist eine lose Integration, die von den verfügbaren Erweiterungen für die jeweiligen SCM-Tools auf der Anwenderseite (z.B. Subversion Plugin für Eclipse) profitiert, genauso wie von der Unterstützung verschiedener am Markt verfügbarer SCM-Systeme durch ein PDM-System mit offenen Schnittstellen. Hier entfällt die Notwendigkeit für ein gemeinsames Datenmodell zwischen PDM und SCM.

Lose Integration bedeutet aber nicht, dass die Integration



Übergabe: Um die Daten aus Software-Projekten besser und einfacher zu verwalten, werden die Informationen an das firmeneigene PDM-System weiter gereicht.

system sehen, warum und in welchem Zusammenhang die Datei geändert wurde. Entsprechende automatisch erzeugte Links von der SCM Log-Meldung in das PDM-System können hier den Entwickler ebenfalls unterstützen.

Auch nicht zu vernachlässigen sind die Lizenz- und Wartungskosten sowie die hohe Komplexität im gesamten Prozess, die durch den Einsatz vieler verschiedener Tools entsteht. Es ist aufwändig und fehlerträchtig, wenn Anwender die Integration der verschie-

setzen, um – berechtigterweise – die ihnen vertrauten Arbeitsweise nicht aufgeben zu müssen. Da aber Hard- und Software zum Produkt gehören, bleibt die Frage nach einer Integration dieser beiden Welten im Sinne des Product Lifecycle Management. Hier gibt es zwei Möglichkeiten: Eine vollständige Integration des SCM-System in das PDM-System, was zu einer gemeinsamen Ablage der Daten und einem einheitlichen Tool beim Anwender führen würde. Dies setzt voraus, dass das zu integrierende

nicht automatisch abläuft. Jede Änderung am SCM-System kann und muss sofort und automatisch an das PDM-System übertragen werden.

Parallelen zur CAD-Integration

Eine lose SCM-Integration in das PDM-System funktioniert im Prinzip wie eine CAD-Integration: Die Daten werden im CAD-System erstellt und dann ins PDM-System eingecheckt beziehungsweise ge-

PDM	SCM
<ul style="list-style-type: none"> ▪ Verwaltung von Metadaten ▪ Verwaltet Objekte ▪ Verwaltet Änderungen an den Objekten ▪ Verwaltet Beziehungen ▪ Eine Beziehung kann Attribute haben ▪ _____ 	<ul style="list-style-type: none"> ▪ Verwaltung von Metadaten ▪ Verwaltet Dateien und Verzeichnisse ▪ Verwaltet Versionen für alle Änderungen einzeln ▪ Nur „kopieren aus“ bzw. Vorgängerversion ▪ _____ ▪ Branch- und Merge-Funktionalitäten zur gleichzeitigen Bearbeitung (Feature Branches etc.)

Gemeinsamkeiten: Da viele Informationen sowohl für im PDM, als auch im SCM hinterlegt sind, liegt die gemeinsame Verwaltung der Daten nahe.

speichert. Beim Speichern wird die Baugruppe samt ihrer Metadaten im PDM-System aktualisiert und ist für alle berechtigten Anwender sichtbar.

Ähnlich verhält es sich beim Editieren einer Quellcode-Datei in einem beliebigen Editor des Anwenders und anschließendem Abspeichern im SCM-System. Nach dem erfolgreichen Einchecken werden sofort die Metadaten im PDM-System zu der eingetragenen Quellcode-Datei und dem jeweils referenzierten Produkt aktualisiert. Bei einer solch losen Kopplung kann der Anwender nach wie vor mit seinen gewohnten Tools in seiner gewohnten Umgebung arbeiten, trotzdem gewinnt der gesamte PLC eines Produktes deutlich hinzu. Innerhalb des PDM-Systems sind alle aus dem SCM-System übertragenen Daten, wie auch im SCM-System selbst, schreibgeschützt und können nur gelesen werden.

Bleibt die Frage, wie bei einer losen Kopplung sichergestellt wird, welche Änderung sich auf welchen Change Request bezieht. Hierzu kann man etwa die Eingabe einer Change Request-Nummer in der SCM-Änderungsmittlung vorsehen, welche dann den Change Request und darüber beispielsweise das Produkt et cetera automatisch referenziert.

Alles in allem ist eine lose SCM-Integration in das PDM-System eine dankbare Abgleichaufgabe, da sich die Daten auf SCM-Seite, einmal angelegt, nicht mehr verändern und die Datenhoheit beim SCM-System bleibt. Auch eine nachträgliche Umladung eines bestehenden SCM-Repositorys ist möglich. Aus dem PDM-System gibt es dann direkte Übergänge von einem System in das Andere, um sich beispielsweise die Unterschiede zwischen zwei Dateiversionen anzeigen zu lassen.

Erfahrungen in der Praxis

Die Benefits sind eine angemessene SCM-Funktionalität für den Software-Entwickler, Integration in bestehende Prozesse und die Verfügbarkeit der SCM-Metainformationen in einer Datenbank für Recherche, Reporting oder GUI (Grafic User Interface). Auch können Anwender ohne installierte SCM-Software Änderungen am SCM Repository über das PDM-System nachvollziehen.

Bei Contact setzen wir für den Produktentwicklungsprozess (PEP) schon seit Jahren eine Integration zwischen SCM-System (erst CVS, nun auch Subversion) und unserer PDM-Software CIM Database ein. Beide SCM-Systeme sind inzwischen vollständig »lose« in das PDM-System integriert. Jeder

Änderung am Quellcode geht ein Change Request im PDM-System voraus. Im Anschluss kann man sich aus dem PDM-System heraus die einzelnen Codeänderungen, den benötigten Aufwand und vieles mehr anzeigen lassen. Auch lässt sich überwachen, ob Bugfixes für eine ältere Version auch in zukünftige Versionen überführt worden sind. Funktionalitäten, die in Subversion aufgrund des Datenmodells nicht so einfach möglich sind, konnten wir durch Übertragung der Metadaten in die PDM-Datenbank leicht nachrüsten: Man kann dadurch sehr schnell erkennen, welche Kopien von einer bestimmten Version einer Datei (welche Branches) gemacht worden sind, und weiß dementsprechend, wo ein Bugfix durchzuführen ist. Zuvor musste Subversion zeitaufwändig jeden Checkin analysieren und den kompletten Baum traversieren. Nun stehen solche Informationen in Sekundenfrist zur Verfügung. -sg-

Referenzen:

»Implementing and integrating product data management and software configuration management« / Ivica Crnkovic; Ulf Asklund; Annita Persson Dahlqvist / ISBN 1-58053-498-8 / Artech House computing library 2003

Contact Software, Bremen
Tel. 0421/20153-0, www.contact.de