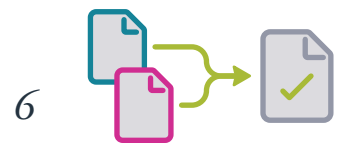


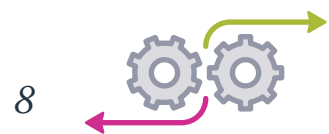
CONFIGURATION MANAGEMENT DIE WICHTIGSTEN USE CASES

Software Configuration Management macht Änderungen bei der Entwicklung von komplexen Systemen kontrollier- und nachvollziehbar. Dafür wurden im Lauf der Zeit folgende Methoden entwickelt:

Use Case 1: Versionen



Use Case 2: Builds



Use Case 3: Changes



Use Case 4: Releases



Ein gutes Configuration Management im Unternehmen bringt folgende Vorteile mit sich:

- ▶ **Änderungen sind kontrollier- und nachvollziehbar**
Jede Änderung wird isoliert durchgeführt und nach einem Freigabeprozess in das Produktivmodell übernommen.
- ▶ **Produktkonfiguration immer im Griff:**
Ein freigegebener Stand des Modells wird zu einem Bestandteil der Produktkonfiguration.
- ▶ **Verbesserte Transparenz:**
Unterschiedliche Produktkonfigurationen lassen sich strukturiert verwalten und warten, die aktuelle Entwicklung ist mit den Anforderungen vergleichbar.
- ▶ **Qualitätssicherung:**
Nur die Modelländerungen, die die festgelegten Qualitätsprüfungen (Review, Validierung, etc.) passiert haben, werden in die Produktkonfiguration übernommen. Jeder potenzielle Fehler lässt sich leicht identifizieren, untersuchen, zurückverfolgen und bei Bedarf korrigieren.
- ▶ **Produktivitätssteigerung:**
Änderungen sind parallel und isoliert durchführbar. Jede Modelländerung kann gezielt und flexibel in die gewünschte Produktkonfiguration übernommen werden. Der Einsatz der in der Softwareentwicklung etablierten Tools ermöglicht einfachere Teamkommunikation.

Configuration Management auch für Modelle wichtig

Configuration Management entwickelte sich in der 50er/60er Jahren in der Luftfahrtindustrie, um die Entwicklung komplexer Systeme kontrollier- und wiederholbar zu machen. Später wurden die Prinzipien von der Softwareindustrie übernommen und stetig weiterentwickelt. Inzwischen zählen mehrere Configuration Management Methoden zum state-of-the-art der Softwareentwicklung: Änderungs-, Versions-, Build- und Release-Management sowie Continuous Integration.

Bei der Entwicklung komplexer Cyber-Physikalischer Systeme ist heute allerdings der gesamte Entwicklungszyklus einzubeziehen. Configuration Management muss daher alle entwicklungsrelevanten Artefakte erfassen: Von den Anforderungen über die Modellierung bis hin zur Dokumentation bzw. zum Source Code. Das fordern etwa auch weit verbreitete Normen der funktionalen Sicherheit wie die ISO26262. Modelle spielen hier eine zentrale Rolle, da sie die Nachverfolgbarkeit zwischen den einzelnen Artefakten des Entwicklungsprozesses ermöglichen. Modellierungswerkzeuge weisen jedoch beim Configuration Management gravierende Defizite auf, die LieberLieber mit LemonTree und darauf aufbauenden Lösungen behebt. In den folgenden Use Cases zeigen wir daher für jeden Teilbereich auf, wie unser Lösungsangebot dazu aussieht.

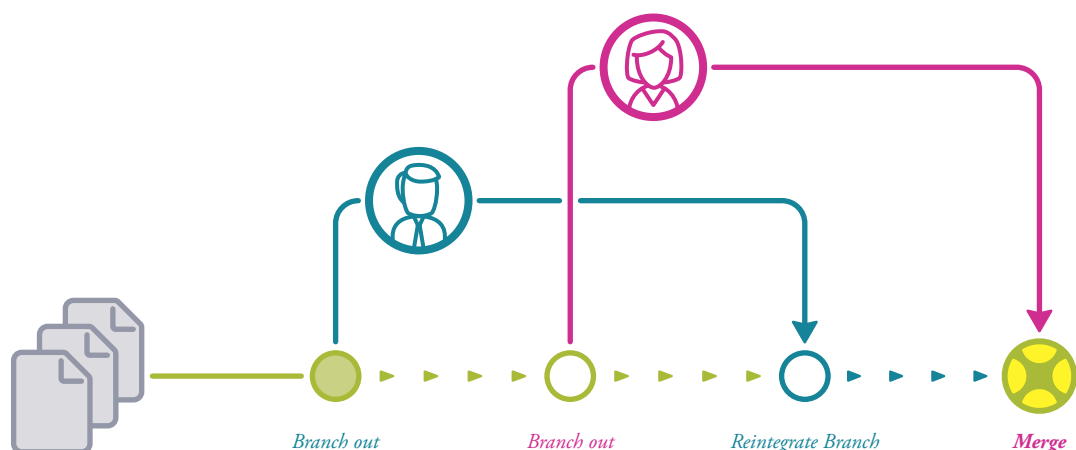
VERSIONSMANAGEMENT VON MODELLEN MIT STATE-OF-THE-ART VERSIONIERUNGSSYSTEMEN

Bei der Verwendung eines modellbasierten Entwicklungsansatzes sollen Teams an parallelen Versionen eines Modells effizient arbeiten können. LieberLieber LemonTree für Enterprise Architect bietet dafür einen detaillierten Modellvergleich und garantiert eine präzise Versionierung von Modellen, definiert durch die Modellierungssprachen wie UML, SysML, BPMN etc. (Offene Standards der OMG).

Die wichtigste Funktion von LemonTree ist der Vergleich und die Zusammenführung von verschiedenen Modellversionen. Damit werden Änderungen der jeweiligen Modellversionen für Anwender leicht nachvollziehbar. LemonTree bietet Ihnen die drei wichtigsten Bestandteile eines zeitgemäßen Modellversionierungssystems:

- ▶ **Jede Änderung ist isoliert durchführbar**
- ▶ **Der Stand vor/nach einer Änderung wird archiviert. Damit lässt sich jede Änderung (Changeset) explizit referenzieren und dokumentieren**
- ▶ **Review und Freigabe jeder Änderung wird durch den Changeset repräsentiert**
- ▶ **Die Validität der Modelle wird immer gewährleistet**

LemonTree wird nahtlos in bestehende Versionierungssysteme integriert. Unsere Kunden haben LemonTree bereits erfolgreich mit SVN, Git oder PTC im Einsatz und profitieren so von den Features, die diese mitbringen. Die Modellversionen werden zentral und automatisch verwaltet. Das ermöglicht die parallele Bearbeitung der Modelle im Team, wie es bei der Entwicklung von Source Code seit Jahren üblich ist.



*Beispiel für die
Anwendung in
der Praxis*

HIMA Paul Hildebrandt GmbH

Die HIMA-Gruppe ist Anbieter von smarten Sicherheitslösungen für die Industrie. Etwa 60 Entwickler nutzen Enterprise Architect für die Erstellung von Modellen und generieren daraus auch Softwarecode für Anwendungen im Embedded- und PC-Bereich. Da bei den Entwicklungen stets die Vorgaben der funktionalen Sicherheit eingehalten werden müssen, ist die genaue Nachverfolgbarkeit bei der Versionierung der Modelle ein wesentliches Kriterium. Dipl.-Ing. (FH) Stefan Müller, Teamleiter Development Software bei HIMA:

„Generell fordern Normen wie IEC 61508 die Existenz eines Configuration Managements. Das bezieht sich auf alle Elemente, also auch auf die UML-Modelle. LemonTree von LieberLieber ist für uns der Schlüssel dazu, ermitteln zu können, was in welcher Revision geändert wurde.“

Eine der großen Herausforderungen bei HIMA war die Integration von Subversion (SVN), einer Software zur zentralen Versionsverwaltung von Dateien und Verzeichnissen. Da die Entwickler täglich fünf bis zehn neue Versionen erstellen, darf der Integrationsvorgang nicht lange dauern. Dazu Oliver Mumenthey, Software-Entwickler bei HIMA:

„Das Team von LieberLieber konnte die Performance von LemonTree auch für unsere bis zu 6 GB großen Modelle immens steigern. Durch die enge Zusammenarbeit erzielen wir im Projekt ausgezeichnete Ergebnisse, das stärkt unser Vertrauen in LieberLieber außerordentlich.“

*Besondere
Highlights*

Rund um die Versionierung von Modellen bringt der Einsatz von LemonTree entscheidende Vorteile:

- ▶ **Revisionsicherheit nach ISO Normen**
- ▶ **Nachvollziehbare Änderungen**
- ▶ **Paralleles, unabhängiges Teamwork**
- ▶ **Versionsmanagement der Modelle durch Tools wie SVN oder Git**
- ▶ **Model Branching**
- ▶ **Änderungshistorie verfügbar**

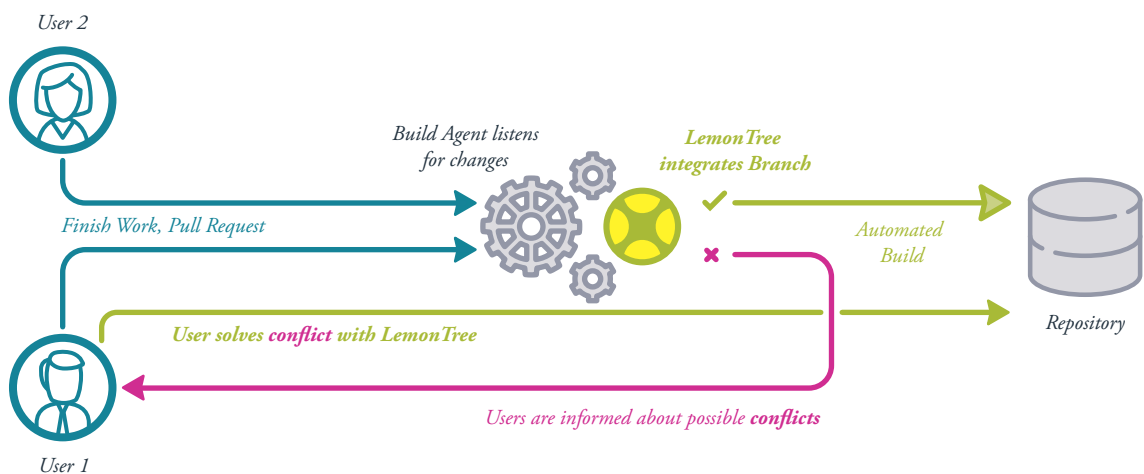
*Nutzen für
Anwender*

Systemingenieure und Modellierer gewinnen durch den Einsatz von LemonTree ein hohes Maß an Sicherheit bei der Entwicklung mechatronischer, komplexer Systeme. Durch die Auslagerung des Versionsmanagements an Werkzeuge wie Git oder SVN wird auf bewährte Konzepte zurückgegriffen.

Diese Werkzeuge zeigen auf, welcher User zu einem bestimmten Zeitpunkt Modellinhalte geändert hat. Darüber hinaus ermöglichen sie einem Team die parallele Durchführung von Änderungen bei Modellen, ohne dabei die Arbeit der anderen Teammitglieder zu beeinflussen. Zur Sicherstellung eines geordneten Change- und Reporting-Prozesses auch für Modelle können User bestimmte Revisionen referenzieren.

BUILD MANAGEMENT - CONTINUOUS INTEGRATION VON MODELLEN

Continuous Integration ist die fortgeschrittenste Methodik des Configuration Managements in der Softwareentwicklung. Dabei wird jede freigegebene Änderung des Source Codes kontinuierlich in einem Repository zusammengeführt und anschließend auf einem Build Server automatisch überprüft, gebildet, getestet und paketierrt. So wird sichergestellt, dass die entwickelte Software stets getestet und lauffähig ist. LieberLieber LemonTree erlaubt erstmals, die Methode der Continuous Integration nun auch für die Modellierungswelt einzusetzen. Ähnlich wie bei der traditionellen Entwicklung von Software können Teammitglieder verteilt und zeitversetzt an verschiedenen Bereichen eines Modells arbeiten und ihre Änderungen einbringen. LemonTree Automation führt dann die einzelnen Versionen am Build-Server (z.B. Jenkins) ohne jegliche Benutzerinteraktion zu einer integrierten Modellversion zusammen. Nur in komplexen Konfliktfällen, wenn also widersprüchliche Änderungen desselben Modellelements auftreten, werden die betroffenen Benutzer benachrichtigt und müssen diese Konflikte mit LemonTree manuell auflösen.



Beispiel für die
Anwendung in
der Praxis

Ein führender deutscher Autozulieferer

Gerade in der Autoindustrie sowie bei deren Zulieferern rücken Verbesserungen rund um die Software- und Systementwicklung immer mehr ins Zentrum der strategischen Überlegungen. Einen mutigen und zukunftsweisenden Schritt setzte in diesem Zusammenhang nun eine Abteilung eines führenden deutschen Autozulieferers. Hier löst man nämlich die bisher übliche, dokumentenzentrierte Entwicklung von Software durch eine

vollständig modellbasierte ab.

„Unser Ziel ist es, das Modell zum Kern unserer Spezifikation zu machen. Dabei wollen wir mit den Modellen aber auch so komfortabel arbeiten wie mit Code in der agilen Softwareentwicklung. Darüber hinaus haben wir in einem Projekt mit der Technischen Universität München eine eigene Methode entwickelt, die in den Modellen anzuwenden ist“, umreißt der Projektleiter die anspruchsvollen Vorgaben.

Beispiel für die Anwendung in der Praxis

Funktionierendes Configuration Management für die Modelle hat sich in diesem Projekt als entscheidender Aspekt auf dem Weg zum Erfolg herauskristallisiert.

Das Ziel: die automatisierte und kontinuierliche Integration von Modellen, auch „Continuous Integration“ genannt. Es wurde beschlossen, die etablierten Methoden aus dem Bereich der Softwareentwicklung auf die Modellierungswelt zu übertragen. Änderungen des Modells sollen laufend in eine Baseline integriert werden, aber anders als bisher mit genauer Kontrolle nach dem vorgegebenen Prozess.

Jede Änderung wird über ein Ticket-System beauftragt und durchläuft dann mehrere Schritte:

1. Spezifikation
2. Durchführung in einem Versionierungssystem
3. Review
4. Freigabe
5. Automatische Integration in das produktive Modell durch einen Build-Server

Eine besonders anspruchsvolle Aufgabe in der Zusammenarbeit mit dem Autozulieferer bestand darin, die geänderten Modellversionen mit jedem Check-In eines Architekten automatisch in das zentrale Modell überzuführen (Build Management - Continuous Integration von Modellen). Im Unternehmen kommt nämlich ein Build Server zum Einsatz, der regelmäßig die mittels Ticket beauftragten, durchgeführten und freigegebenen Änderungen automatisiert in das produktive Modell überführt. Bisher war dazu aber vorab eine manuelle Zusammenführung der Änderungen in LemonTree notwendig, die nun auch in den automatischen Prozess integriert wurde.

Besondere Highlights

Rund um das Thema „Build Management“ von Modellen bringt der Einsatz von LemonTree entscheidende Vorteile:

- ▶ **Automatische Zusammenführung von Änderungen**
- ▶ **3-Wege Vergleich/Merge**
- ▶ **Konflikt-Auflösung durch „Power-User“**
- ▶ **Änderungshistorie verfügbar**

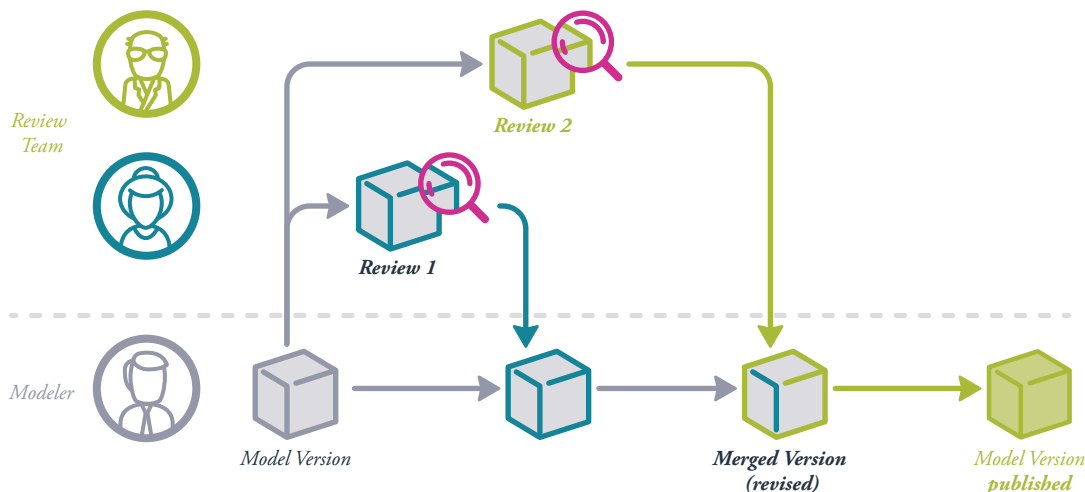
Nutzen für Anwender

Continuous Integration für Modelle mit LemonTree sorgt dafür, dass zwar die Modellierer unabhängig voneinander gleich an mehreren Änderungen arbeiten können, aber trotzdem immer ein konsistenter Modellstand vorhanden ist. Die Merge-Funktionalität von LemonTree unterstützt sie dabei, verschiedene Versionen von Modellen rasch, sicher und mit eingebauter Unterstützung konfliktfrei zusammenführen.

CHANGE MANAGEMENT UND REVIEWS VON MODELLEN

Normen der funktionalen Sicherheit wie die ISO26262 verpflichten zum Configuration und Change Management für alle entwicklungsrelevanten Artefakte. Modelle sind gerade bei der Entwicklung sicherheitsrelevanter, mechatronischer Systeme mittlerweile ein fester Bestandteil des Entwicklungsprozesses und müssen daher einem passendem Change Management unterliegen. Damit erfordern Änderungen an den Modellen, die die zu entwickelnden Artefakte beschreiben, besondere Aufmerksamkeit („first-class entities“).

Mit LieberLieber LemonTree werden diese Änderungen nachvollziehbar aufbereitet und sind zu jedem Zeitpunkt abfragbar. Dabei sind nicht nur der aktuelle Stand des Modells wichtig, sondern auch die Änderungen am Modell bzw. zwischen einzelnen Versionen des Modells. Die Information darüber, welcher Benutzer wann etwas am Modell geändert hat, hilft nicht nur bei der Zusammenarbeit im Team, sondern ist gerade im Functional Safety Bereich unabdingbar. LemonTree ermöglicht es auch, diese Änderungen zu begutachten oder zu „reviewen“. Die ausgeklügelte Darstellung der Änderungen und die Filtermöglichkeiten in LemonTree erleichtern diese Arbeit enorm. Darüber hinaus bietet LemonTree durchdachte Features, um eine Review-Konfiguration inklusive der zu begutachtenden Modellversionen und den aktuellen Stand des Reviews abspeichern zu können. So wird es möglich, im Reviewprozess jederzeit eine Pause einzulegen und danach ohne Informationsverlust fortzufahren. Nicht zuletzt lassen sich Review-Kommentare in eigene Branches nachvollziehbar einfügen, um einen optimierten Review-Prozess zu unterstützen.



Beispiel für die Anwendung in der Praxis

Ein führender deutscher Autozulieferer

Gerade in der Autoindustrie ist die nachweisbare Einhaltung von Normen der funktionalen Sicherheit wie die ISO 26262 unumgänglich. Damit

sind die Hersteller und Zulieferer zum Configuration- und Change Management auch in Modellen verpflichtet.

Beispiel für die Anwendung in der Praxis

„Für uns sind verschiedene Safety-Normen wie Automotive SPICE und die Einhaltung des „Automotive Safety Integrity Level ASIL“ maßgeblich bei der Softwareentwicklung. Da in der Abteilung Software-Architektur die Modellierungs-Plattform Enterprise Architect schon länger erfolgreich eingesetzt wird, konnten wir uns von der Nützlichkeit der modellbasierten Softwareentwicklung überzeugen. Aber erst durch den Kontakt mit Lieber-Lieber und die Verwendung von LemonTree wird unsere tägliche Arbeit mit den Modellen nun wesentlich erleichtert und beschleunigt“, so der Mitarbeiter eines führenden deutschen Autozulieferers, der in der Abteilung für Funktions-Software an der Erstellung sicherheitskritische Software mitwirkt.

Um die Einhaltung aller gesetzlichen Vorgaben

auch bei Änderungen in der Software belegen zu können, sind für die Arbeitsgruppe Reviews besonders wichtig. Dabei werden die Arbeitsergebnisse der Softwareentwicklung manuell geprüft, um die geforderte Qualität sicherzustellen.

„Bei uns muss jede Veränderung im Modell – auch Change Request genannt – einem Review unterzogen werden. Dafür bietet LemonTree hervorragende Möglichkeiten, die unsere Arbeit sehr beschleunigen“, so der Mitarbeiter. Konnten bisher nur kleine Änderungen zwischen zwei Modellversionen gereviewt werden, so erweitert LemonTree die Möglichkeiten auf komplexere Modelle und unterstützt die Arbeit mit vielen nützlichen Features.

Besondere Highlights

Rund um das Thema „Reviews von Modellen“ bringt der Einsatz von LemonTree entscheidende Vorteile:

- ▶ **Revisionssicherheit nach ISO Normen**
- ▶ **Nachvollziehbare Änderungen**
- ▶ **Review-Kommentare zusammenführen**
- ▶ **Zusätzliche Checkmarks für die Filterung bereits begutachteter Änderungen**
- ▶ **Filter-Möglichkeiten in der Änderungsliste**
- ▶ **Export der Änderungen als CSV**
- ▶ **XML Export für eigene Change-Visualisierungen**
- ▶ **Änderungshistorie verfügbar**

Nutzen für Anwender

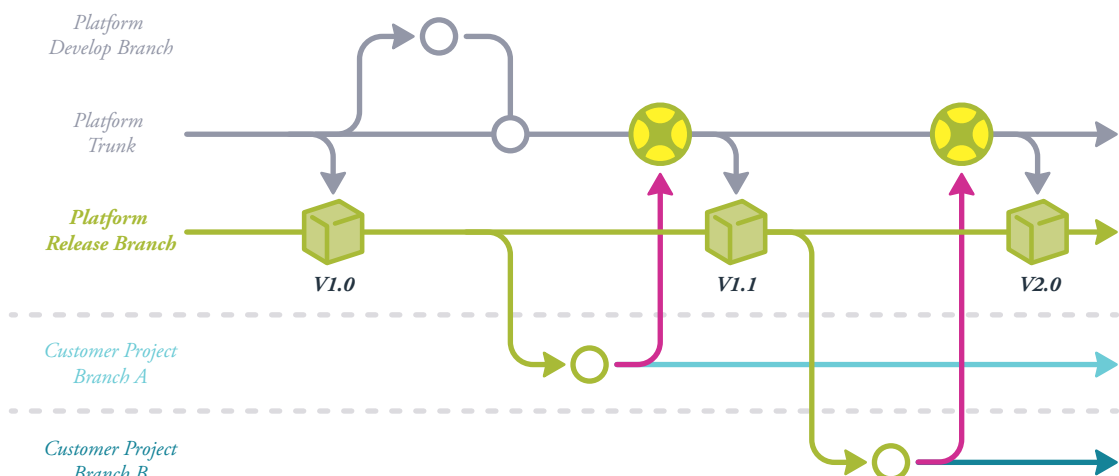
Modellierer und Reviewer bekommen mit LemonTree die Sicherheit, dass alle Änderungen an den UML- oder SysML-Modellen zuverlässig zusammengeführt werden und jede Änderung für sich präzise dokumentiert ist. Die lückenlose Information darüber, welcher User exakt was und wann geändert hat, ist in der Entwicklung sicherheitsrelevanter Artefakte oder auch in anderen Bereichen äußerst wichtig. Die Modelländerungen werden verständlich präsentiert und lassen sich jederzeit einem kontrollierten Review unterziehen. Ergänzende Reviewkommentare werden in einer eigenen Modellversion gepflegt, was dem Benutzer garantiert, das produktive Modell nicht zu zerstören. Sollten mehrere Reviewer solche Anmerkungen pflegen, so lassen sich auch diese einfach mit LemonTree zusammenführen. Verschiedene Export- Möglichkeiten (z.b. XML) dienen dazu, Änderungen über den Zeitverlauf zu dokumentieren.

RELEASE MANAGEMENT FÜR MODELLE IN DER PRODUKTLINIEN-ENTWICKLUNG

Kosten- und Innovationsdruck sind die zentralen Treiber für den Umstieg auf eine modellbasierte Produktlinienentwicklung. Um damit die erwünschte Verkürzung von Entwicklungszyklen auch wirklich zu erreichen, ist eine enge Zusammenarbeit zwischen der Plattformentwicklung und den verschiedenen Kundenprojekten notwendig. Bei komplexen Produkten mit sehr vielen Komponenten und betreuenden Fachgruppen, die bisher auf Basis eines gemeinsamen Modells arbeiten, ergeben sich dabei völlig neue Herausforderungen auf Modellebene. Es ist nur schwer möglich, die sogenannte „Basis-Software“, die die Grundlage der einzelnen Produktlinien bildet, als eigenständige Plattform zu etablieren und in einzelnen Kunden-Projekten zu adaptieren und zu erweitern. Dabei ist es wichtig, etablierte Methoden aus der Softwareentwicklung einsetzen zu können, um die unterschiedlichen Lebenszyklen der Projekte sowie der Plattform zu berücksichtigen.

LieberLieber hat mit LemonTree speziell für die Einführung einer modellbasierten Produktlinienentwicklung mit einem Paket-Verwaltungssystem (Model Package Management System, MPMS) eine Lösung für die Modellierung mit Enterprise Architect entwickelt. Damit können verteilte Teams an verschiedenen Modellversionen arbeiten, während gleichzeitig die „Basis-Software-Plattform“ in einem konsistenten Zustand gehalten wird. Die zentrale Idee dieses Konzepts ist die Anwendung bereits existierender Ansätze, die sich beim Konfigurationsmanagement von Quellcode etabliert haben.

So lassen sich einerseits herausgeschnittene Teile des Modells aus der Plattformentwicklung ohne Verlust der Referenzen als Komponenten in der Projektentwicklung verwenden. Andererseits wird es auch möglich, Änderungen in beide Richtungen zusammenzuführen (Merge).



*Beispiel für die
Anwendung in
der Praxis*

Beispiel: Preh Car Connect

Die Produktlinien-Entwicklung unterstützt Unternehmen dabei, im Rahmen einer Plattform bereits erstellte Software wiederzuverwenden bzw. zu variieren. Die entstehenden Varianten unterscheiden sich in ihrem Funktionsumfang, die Grundbestandteile der Software kommen dabei aber immer wieder zum Einsatz. Dipl.-Informatiker Tim Michaelis, Softwarearchitekt bei Preh Car Connect:

*„Da wir im Unternehmen stark wachsen und viele Seri-
enprojekte umsetzen, lassen sich dabei Bestandteile unse-
rer eigenen Software immer wieder verwenden. Bisher ist
das allerdings nur durch aufwendiges manuelles Kopieren
insbesondere der Modelle möglich. Gemeinsam mit Lie-
berLieber arbeiten wir daher an einem Projekt, mit dem
sich solche Vorgänge automatisieren lassen. LemonTree ist
für uns ein wichtiger Baustein, um diese Entwicklungsschritte
schneller und einfacher umsetzen zu können.“*

Bei Preh Car Connect arbeiten insgesamt ca. 1.200 Mitarbeiter an acht Standorten weltweit. Der Hauptsitz und zentrale Entwicklungsstandort des Unternehmens befindet sich in Dresden. Das Unternehmen entwickelt und produziert seit mehr als 20 Jahren hochwertige Infotainmentsysteme für namhafte OEMs, darunter auch die Volkswagengruppe.

*Besondere
Highlights*

- ▶ **Versionierung auf Paketebene**
- ▶ **Paketverwaltung auf Basis von Git, etc.**
- ▶ **Update eines einzelnen Pakets mittels LemonTree-Merge**
- ▶ **Abhängigkeitsanalyse zwischen Paketen**
- ▶ **Veröffentlichung bestimmter Paketversionen**

*Nutzen für
Anwender*

Jeder Nutzer hat in diesem Szenario die Sicherheit, bestimmte Artefakte des Modells in einer definierten Version zu benutzen oder genau zu referenzieren. Er kann einzelne Versionen eines spezifischen Pakets in sein Modell importieren, ohne seine bisherigen Entwicklungen zu zerstören, da LemonTree die Teile intelligent zusammenführt. So kann ein Team einzelne Teile der Entwicklung in verschiedenen Projekten wiederverwenden.

ÜBER LIEBERLIEBER SOFTWARE

Wir sind ein Software-Engineering Unternehmen. Das Know-how unserer Mitarbeiter liegt in der modellbasierten Software- und Systementwicklung auf Basis von Tools wie Enterprise Architect von Sparx Systems.

Unsere Auftraggeber sind Unternehmen, die besonderen Wert auf die Qualität ihrer Software- und Systementwicklung legen. Diese wollen in ihren komplexen Szenarien immer den Überblick bewahren und sicherstellen, dass in ihrer Entwicklung vor allem die sicherheitsrelevanten Anforderungen gut nachvollziehbar in Modellen abgebildet sind.

Speziell für diese Aufgabe stellen wir eigene Werkzeuge zur Verfügung, wie zum Beispiel LemonTree und Embedded Engineer. Ergänzend bieten wir eine Reihe von Dienstleistungen, mit denen wir unsere Werkzeuge in den Entwicklungsprozess unserer Kunden integrieren und nutzbar machen.

Mehr Informationen finden Sie unter:

www.lieberlieber.com