Software Corner

Ulf Pleissner

Mustersynchron entwickeln

Die Entwickler sind spezifisch in der Tiefe ihres Fachwissens ausgebildet und dementsprechend sind auch die Entwicklungsprozesse aufgebaut und etabliert. Die Wissensfusion in den technischen Systemen – ausgelöst durch das Verwischen der Grenzen zwischen Mechanik, Elektronik und Software – stellt diese etablierten Entwicklungsprozesse zunehmend in Frage.

Die Entwicklungsprozesse stehen vor tief greifenden Veränderungen. Wichtig hierbei festzuhalten ist, dass zum einen die technologischen Lebenszyklen zwischen Mechanik, Elektronik und Software elementar unterschiedlich sind und dass zum anderen insbesondere die Flexibilität von Software wesentlich höher bewertet wird, als die von elektronischen oder mechanischen Komponenten.

Dabei führt die Wissensfusion zu einem Anstieg der Freiheitsgrade in den möglichen Lösungen, die bei Projektstart weder vollständig bekannt sind, noch entschieden werden können. Diese Verlagerung von Entscheidungen in die Projekte zwingt die Unternehmen, sich von dem Plan-Steuerungsverfahren zu verabschieden und stattdessen kleine Regelkreise in der Projektabwicklung zu installieren. Der Projektleiter übernimmt in diesem Prozessmodell die Funktion des Reglers. Seine Aufgabe ist es, den Projektverlauf so zu beeinflussen, dass dieser möglichst nicht oder nur gering vom Fortschrittsplan abweicht. Aus dieser Betrachtungsweise resultieren die folgenden vier Forderungen nach:

1. Termingerechte Synchronisation

Das in Software-Entwicklungsprozessen häufig umgesetzte V-Modell realisiert eine direkte Rückkopplung der implementierten Software-Funktionalität in die Entwicklung mit dem Ziel, die Qualitätssicherung durch schnelle Fehler-Erkennung und -Behebung effizienter zu gestalten. Durch seine Fokussierung auf die Software-Entwicklung ist das V-Modell jedoch für die Integration eines Gesamtsystems bestehend aus Mechanik, Elektronik und Software nur begrenzt geeignet. Grund ist: Der Prozess der Integration und Freigabe eines Gesamtsystems ist geprägt durch eine

festgelegte Abfolge von Musterständen. Ein Musterstand entsteht durch die Integration aller Systemkomponenten zu einer funktionsfähigen Einheit mit genau definierter Funktionalität zu einem definierten Zeitpunkt. Der Software-Entwicklungsprozess muss zukünftig an diese spezifischen Anforderungen der technologie-übergreifenden Synchronisierung von Musterständen angepasst und in den Gesamtentwicklungsprozess integriert werden.

2. Transparenz der Funktions-Erfüllung

Die Beherrschung der Komplexität und die Sicherstellung der Qualität in einem Mustersynchronen Entwicklungsprozess erfordert in allen Entwicklungsphasen eine hohe Transparenz des real erreichten Projekt-Fortschritts. Die objektive Messung des Projekt-Fortschritts stellt somit eine elementare Anforderung an den Entwicklungsprozess dar und muss die Bewertung des Projektes aus unterschiedlichen Blickwinkeln ermöglichen. Der Projektfortschritt wird am besten über die fünf Punkte – Arbeitsfortschritt, Funktionserfüllung, Kumulierter Reifegrad, Fehlerstatus, Termintreue – erfasst.

3. Interaktion zwischen den Wissensträgern der verschiedenen Technologien

Das Zusammenwirken der drei Disziplinen Mechatronik, Elektronik und Software wird erschwert durch unterschiedliche Vorgehensweisen, Terminologien und erforderliches Abstraktionsvermögen. In der Praxis hat dies oft Disharmonien in der Entscheidungsfindung zur Folge und verlangt von allen am Projekt beteiligten Wissensträgern Offenheit und Bereitschaft zur Interaktion.

4. Sicherstellen der Weiter- und Wiederverwendbarkeit von Software

Die hohen Entwicklungsaufwendungen zwingen die Unternehmen dazu, bewährte Lösungen möglichst nicht nur einmal, sondern mehrfach zu nutzen. Hierbei ist zu unterscheiden zwischen Weiter- und Wiederverwendbarkeit. Unter Weiterverwendbarkeit ist die Verwendung von Softwaremodulen innerhalb eines Projektes zu verstehen. Dies soll vor allem für Teillösungen gelten, die aufgrund der musterbezogenen Systemsynchronisation benötigt werden. Im Projekt muss sichergestellt sein, dass Softwaremodule präzise doku-

mentiert und von einem Musterstand zum nächsten identisch weiterverwendet oder kompatibel weiterentwickelt werden.

Um die Chance auf Wiederverwendbarkeit (Multiplikation) in weiteren Projekten und Produkten zu gewährleisten sind die unternehmerischen Randbedingungen zu ändern. Nicht nur die Vorgabe einer modularen Softwarestruktur, sondern auch die Erzeugung eines kosten- und terminorientierten Applikationszwangs führt dazu, vorhandene Lösungen (Module) nicht unbegründet in Frage zu stellen.

Die konsequente Wieder- und Weiterverwendung von Softwaremodulen (Feature) hat wesentlichen Einfluss auf die Entwicklungszeiten, -kosten und -qualität. Sie erhöht die Produktivität bei gleichzeitiger Verringerung des Terminrisikos der musterbezogenen Systemintegration. hap



Ulf Pleissner

ist Geschäftsführender Gesellschafter der TMG Markt und Innovation sowie der TMG Technologie und Engineering.