

## Mustersynchrone Softwareentwicklung

# Sicherstellen des inhaltsgerechten Reifegrads von Software



Die Zunahme der Abhängigkeiten und die Verschmelzung der Grenzen zwischen Mechanik, Elektronik und Software vollziehen sich deutlich schneller, als sich die Unternehmensprozesse und Denkstrukturen der Entwickler diesem Prozess anpassen können.

> Sind die Entwickler heute noch spezifisch in der Tiefe ihres Fachwissens ausgebildet, so wird zukünftig die Wissensfusion in den technischen Systemen die heute bestehenden Entwicklungsprozesse in Frage stellen und ein gegenseitiges Verständnis der Disziplinen erzwingen. Dieser Trend stellt heute schon die wichtigste Herausforderung im Fahrzeugbau dar.

### Neue Anforderungen aus der Wissensfusion

Die zunehmende Wissensfusion verlangt nach tiefgreifenden Veränderungen der Entwicklungsprozesse und verändert die Erwartungshaltung an die unterschiedlichen Technologien: Die Flexibilität von Software wird wesentlich höher bewertet als die von elektronischen oder mechanischen Komponenten. Zusätzlich sind die technologischen Lebenszyklen zwischen Mechanik, Elektronik und Software elementar unterschiedlich. Die Wissensfusion führt 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 Pro-

zessmodell die Funktion des Reglers. Seine Aufgabe ist es, den Projektverlauf so zu beeinflussen, dass dieser möglichst nicht oder nur gering vom Fortschrittsplan abweicht. Daraus resultieren die Forderungen nach:

- > termingerechter Synchronisation der Entwicklungsprozesse,
- > erhöhter Transparenz der Funktionserfüllung der Software,
- > ständiger Interaktion zwischen den Wissensträgern der verschiedenen Technologien und
- > Sicherstellung der Wieder- und Weiterverwendbarkeit von Software.

### Termingerechte Synchronisation

Aufgrund der wachsenden Anforderungen an Flexibilität, Funktionalität und Updatefähigkeit werden immer größere Anteile der Produktfunktionalität in Software realisiert. Gleichzeitig steigen die Anforderungen an Sicherheit und Zuverlässigkeit extrem an.

Das in Softwareentwicklungsprozessen häufig umgesetzte V-Modell realisiert eine direkte Rückkopplung der implementierten Softwarefunktionalität in die Entwicklung mit dem Ziel, die Qualitätssicherung durch schnelle Fehlererkennung und -behebung

effizienter zu gestalten. Durch seine Fokussierung auf die Softwareentwicklung ist es jedoch nur begrenzt geeignet für die Integration eines Gesamtsystems bestehend aus Mechanik, Elektronik und Software. 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 Softwareentwicklungsprozess muss an diese spezifischen Anforderungen der technologieübergreifenden Synchronisation von Musterständen angepasst und in den Gesamtentwicklungsprozess integriert werden.

### Transparenz der Funktionserfüllung

Die Beherrschung der Komplexität und die Sicherstellung der Qualität in einem mustersynchronen Entwicklungsprozess erfordert in allen Entwicklungsphasen eine hohe Transparenz des real erreichten Projektfortschritts. Die objektive Messung des Projektfortschritts stellt somit eine elementare Anforderung an den Entwicklungsprozess dar und muss die Bewertung des Projekts aus unterschiedlichen Blickwinkeln ermöglichen. Die Erfassung des Projektfortschritts in den folgenden fünf Dimensionen liefert die geforderte Trans-

parenz als Voraussetzung für eine erfolgreiche Projektsteuerung:

- > Arbeitsfortschritt,
- > Funktionserfüllung,
- > Kumulierter Reifegrad,
- > Fehlerstatus,
- > Termintreue.

Der Arbeitsfortschritt ist ein Maß für die inhaltliche Bewertung der erzielten Ergebnisse, gemessen in „Arbeitswerten“. Mit der Funktionserfüllung werden die gleichen Ergebnisse unter dem Aspekt der erzielten Funktionalität bewertet. Der kumulierte Reifegrad misst den Implementierungsstand der Software und der Fehlerstatus ist ein Maß für die erreichte Qualität. Aktualität und Verlässlichkeit der Planung werden mit der Termintreue gemessen.

### Interaktion zwischen den Wissensträgern

Mit der Möglichkeit, Funktionen über Mechanik, Elektronik oder Software realisieren zu können, kann die Funktionslösung nicht mehr durch einen Fachbereich alleine vorgegeben werden. Um die gewünschte Wissensfusion zu erreichen, müssen die Denkweisen unterschiedlicher Disziplinen zusammenwirken. Dies wird jedoch 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 die Offenheit und Bereitschaft zur Interaktion.

### 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 unterscheidet man zwischen Weiter- und Wiederverwendbarkeit. Unter Wiederverwendbarkeit ist die Verwendung von Softwaremodulen innerhalb eines Projektes zu verstehen. Dies soll vor allem für Teillösungen gelten, die aufgrund der musterbezogenen Systemsynchronisation benö-

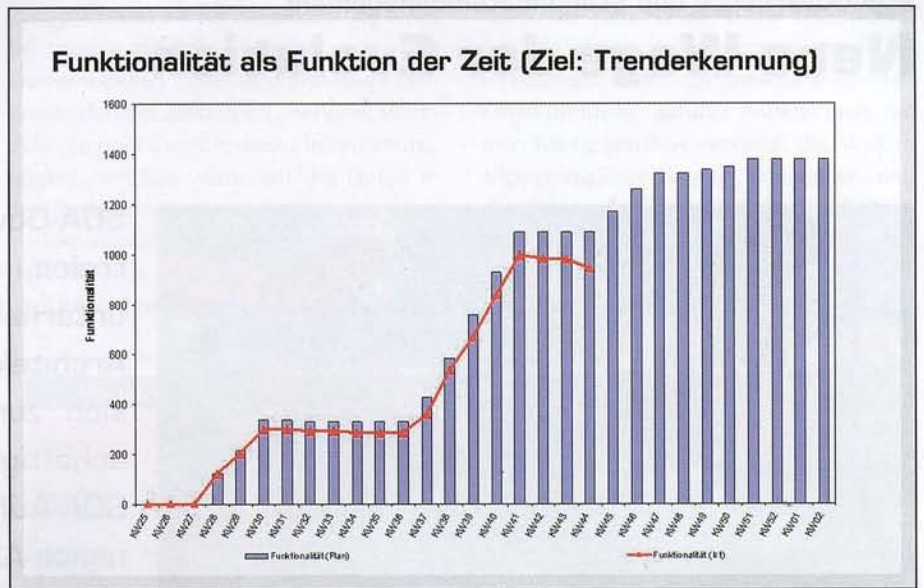


Bild 1: Funktionalität als Funktion der Zeit (Ziel: Trenderkennung).

tigt werden. Im Projekt muss sichergestellt werden, dass Softwaremodule präzise dokumentiert 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 Entwicklungszeit, -kosten und -qualität. Sie erhöht die Produktivität bei gleichzeitiger Verringerung des Terminrisikos der musterbezogenen Systemintegration und muss deshalb integraler Bestandteil zukünftiger Softwareentwicklungsprozesse sein.

Dipl.-Wirtsch.-Ing. Ulf Pleissner  
info@tmg-karlsruhe.de

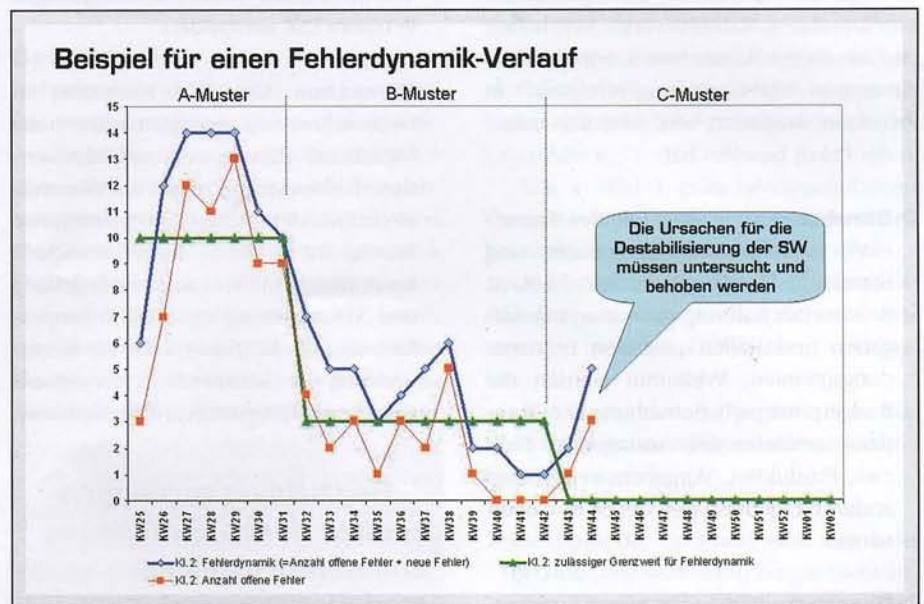


Bild 2: Beispiel für einen Fehlerdynamik-Verlauf. (Bilder: TMG Karlsruhe)