



Prozess für funktionale Sicherheit

Modellbasierte Softwareentwicklung für elektrifizierte
Antriebsstränge nach ISO 26262



Für die Realisierung sicherheitsrelevanter Fahrzeugfunktionen hat die Siemens-Division Drive Technologies ihre klassische Softwareentwicklung um die modellbasierte Entwicklung erweitert. Hierfür wurde ein eigener Prozess auf Basis von dSPACE TargetLink mit Unterstützung des strategischen dSPACE TargetLink Partners Model Engineering Solutions GmbH definiert, der den Anforderungen der Norm ISO 26262 („Road Vehicles – Functional Safety“) genügt.



Sicherheitsrelevante Software im Fahrzeug

Viele Zulieferer der Automobilindustrie entwickeln sicherheitsrelevante Software modellbasiert oder stehen vor der Herausforderung, ihre Softwareentwicklung zukünftig um modellbasierte Verfahren zu erweitern. Die modellbasierte Entwicklung bringt viele bereits belegte Vorteile mit sich wie höhere Qualität und einfachere Wartbarkeit der Software. Mit Einführung der neuen Norm ISO 26262 steht nun ein internationaler Standard für die funktionale Sicherheit von Elektrik/Elektronik-Systemen in Fahrzeugen zur Verfügung. Die ISO 26262 ist eine Ableitung der Sicherheitsnorm IEC 61508, angepasst an die spezifischen Gegebenheiten im Automobil. Im Gegensatz zur IEC-Norm geht die ISO 26262 konkret auf die modellbasierte Entwicklung ein. Dieser neue Standard stellt klare Anforderungen daran, was die Entwicklung für sicherheitsrelevante Software im Fahrzeug leisten muss, um deren funktionale Sicherheit im Hinblick auf unterschiedliche Automotive Safety Integrity Levels (ASIL, klassifiziert in Stufen von A bis D) zu gewährleisten. Die Frage, wie dies im Detail umzusetzen ist, bleibt durch den Standard weitgehend unbeantwortet. Die Praxis hat gezeigt, dass es für die Anwendung der Norm kein allgemeingültiges Vorgehensmodell gibt, sondern nur projektspezifische Lösungen zum Erfolg führen. Diese berücksichtigen die vorhandenen Prozesse und Toolketten der jeweiligen Organisation.

Elektrische Fahrzeugkomponenten von Siemens

Siemens Drive Technologies entwickelt, produziert und vermarktet Schlüsselkomponenten für elektrisch angetriebene Fahrzeuge, und zwar projektbasiert, je nach den spezifischen Anforderungen der Kunden aus der Automobilindustrie. Das Portfolio reicht dabei von Elektromotoren

(Abbildung 1) und Leistungselektronik bis zur intelligenten On-Board-Ladetechnik. Für Softwarefunktionen wurde bereits ein klassischer Entwicklungsprozess etabliert, der sich maßgeblich an dem in der Softwareentwicklung weit verbreiteten V-Modell orientiert. Dieser Entwicklungsprozess berücksichtigt die Vorgaben von Automotive SPICE (aSPICE: eine für das Automotive-Umfeld speziell angepasste Version des internationalen Standards ISO/IEC 15504 „SPICE“) und CMMI-Dev (Capability Maturity Model Integration for Development). Die Einhaltung dieser beiden Reifegradmodelle wird von nahezu allen Fahrzeugherstellern gefordert und ist daher essentiell.

Anforderungen an die modellbasierte Entwicklung sicherheitsrelevanter Software

Siemens Drive Technologies hatte die Anforderung, die modellbasierte Entwicklung in die vorhandene klassische Entwicklung optimal zu integrieren, da zukünftig vermehrt sicherheitsrelevante Fahrzeugfunktionen modellbasiert mit MATLAB®/Simulink®/Stateflow® und dSPACE TargetLink® entwickelt werden sollen. Dazu mussten mehrere Kriterien erfüllt sein:

- Definieren eines ISO-26262-konformen Prozesses für die modellbasierte Entwicklung, der sich optimal in die

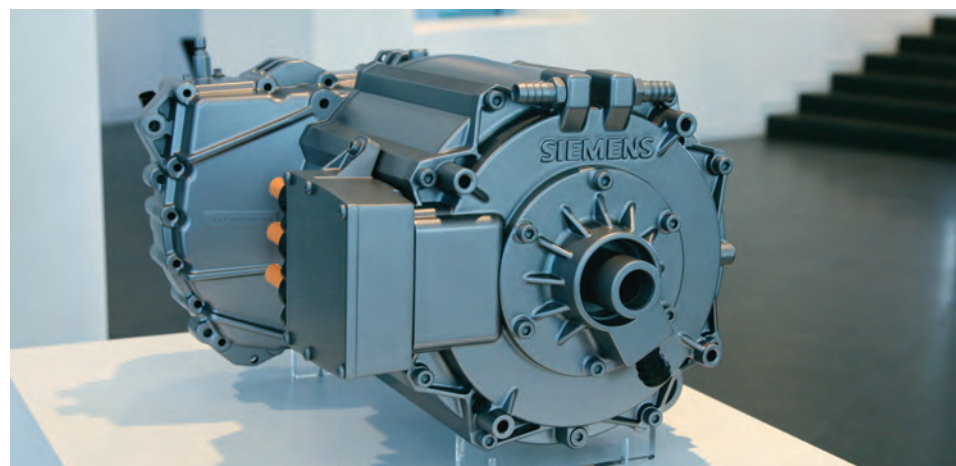
vorhandene Prozesslandschaft einfügt, die verfügbaren Werkzeuge berücksichtigt und keine der geforderten Prozesseigenschaften (z. B. aSPICE-Konformität) verletzt.

- Berücksichtigen aller Anforderungen aus ASIL A bis D mit Fokus auf Teil 6 des ISO-Standards (Produktentwicklung: Softwareebene).
- Definieren der Vorgehensweisen und Entwurfsmuster für den Entwurf von Modellarchitekturen mit Simulink und TargetLink im Rahmen der ISO 26262.
- Sicherstellen, dass alle funktionalen Anforderungen bis in den generierten Code hinein zurückverfolgt werden können (Requirements Traceability).
- Gewährleisten, dass bereits auf Modellebene gegen die Anforderungen getestet werden kann, also noch bevor der generierte Code überhaupt vorliegt.

Planung einer projektspezifischen, ISO-26262-konformen Umsetzung

Für die Umsetzung der Anforderungen hat Siemens Drive Technologies als professionellen Partner die Model Engineering Solutions GmbH (MES) beauftragt. MES ist TargetLink Strategic Partner und berät die Industrie im Bereich der Qualitätssicherung eingebetteter Software im Automobil. MES bietet dSPACE Partner-Dienstleistungen

Abbildung 1: Permanenterregter Synchronmotor für automotive Anwendungen.



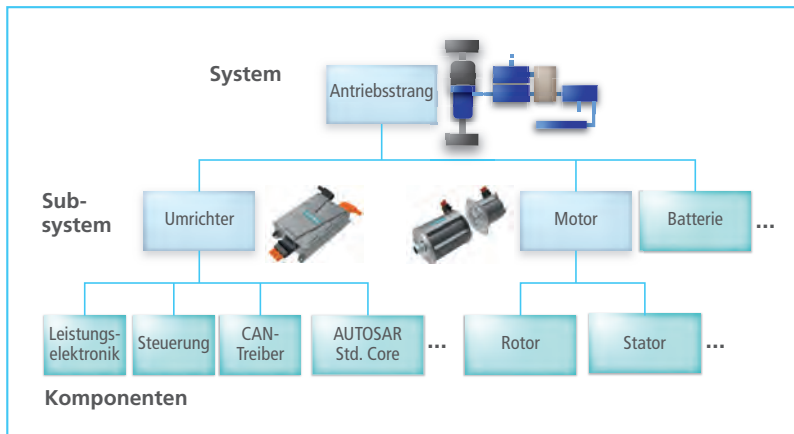


Abbildung 2: Systemzerlegung und Definition der Ebenen eines elektrischen Antriebsstrangs.

rund um die ISO-26262-konforme Entwicklung an, u.a. GAP-Analyse (Identifizierung strategischer und operativer Lücken), Prozessmodellierung, Prozesshandbücher und Unterstützung bei der Prozessimplementierung. MES hat sich im Rahmen zahlreicher Serienprojekte einen umfassenden Überblick verschafft, wie sicherheitsrelevante Software für das Automobil in Deutschland entwickelt wird und wie die Anforderungen der ISO 26262 insbesondere in modellbasierten Softwareprojekten umzusetzen sind.

Modellbasierter Entwicklungsprozess bei Siemens Drive Technologies

Die ISO-26262-konforme Softwareentwicklung erfolgt im Bereich der elektrifizierten automotiven Antriebsstränge von Siemens Drive Technologies. Das komplexe System Antriebsstrang lässt sich in verschiedene Subsysteme und die zugehörigen Systemebenen zerlegen (Abbildung 2). Auf der Subsystemebene befinden sich der Umrichter (Inverter), der den Fluss der elektrischen Energie zwischen Batterie (Gleichspannung) und Motor (Wechselspannungen) realisiert, und der Motor, der anschließend die elektrische in mechanische Energie wandelt. Die Subsysteme wiederum bestehen aus Komponen-

ten, beispielsweise dem Stator des Motors oder der Steuerelektronik des Umrichters. Die Runtime-Software, die auf dieser Steuerelektronik ausgeführt wird, befindet sich eine Ebene unterhalb der Komponentenebene, der sogenannten Modulebene. Die Software selbst (nicht dargestellt) gliedert sich in zahlreiche Softwaremodule, die wiederum in mehrere Softwarefunktionen unterteilt werden können. Einige dieser Funktionen werden, auf Basis des hier vorgestellten Prozesses, zukünftig modellbasiert entwickelt.

Der Softwareentwicklungsprozess

Die code- und modellbasierte Softwareentwicklung erfolgt nach dem V-Modell (Abbildung 3). Ausgangssituation für die Modulentwicklung

sind die Anforderungen für ein Softwaremodul, wie beispielsweise die Momentenregelung des elektrischen Antriebs. Im Systementwicklungsprozess entstehen diese Anforderungen durch schrittweise Verfeinerung der Kundenanforderung bis auf die entsprechende Systemebene (hier: Modulebene). Den Entwurfsphasen stehen geeignete Testphasen gegenüber, um den Reifegrad der Funktionsmodelle und -software abzusichern.

Phasen des modellbasierten Entwicklungsprozesses

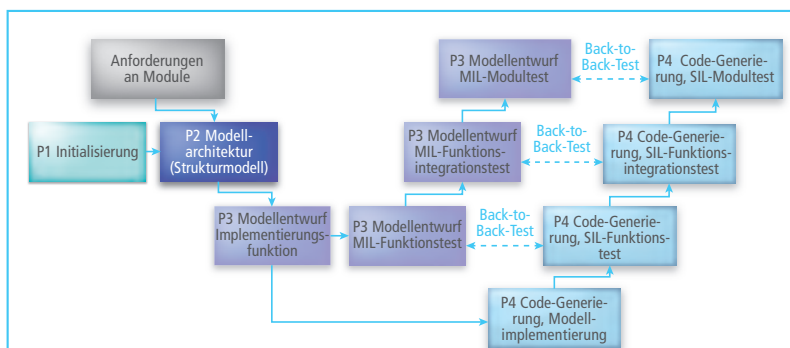
Dieser modellbasierte, ISO-26262-konforme Entwicklungsprozess besteht aus vier Kernphasen:

Phase 1 – „Initialisierung“:

Diese Phase umfasst alle vorbereitenden Maßnahmen vor Projektstart wie Kick-off-Meeting, Aufsetzen der Entwicklungsumgebung, Definition der Anforderungen an das Umgebungsmodell sowie Festlegung und Zuweisung von Rollen und Verantwortlichkeiten. Die klassischen Rollen aus der Softwareentwicklung, wie Software Engineer, Softwarearchitekt usw., wurden hier um neue Rollen für die modellbasierte Entwicklung erweitert, wie beispielsweise den Model Engineer für die Funktionsmodellierung in Simulink.

Als Anforderung der ISO-Norm wird in dieser initialen Phase zum Beispiel die Auswahl der projektrelevanten

Abbildung 3: V-Modell für modellbasierte SW-Entwicklung bei Siemens Drive Technologies (Auszug).



Modellierungsrichtlinien festgelegt (vgl. ISO 26262-6, §5.4.7), die mit dem Model Examiner (MXAM) automatisiert überprüft werden können.

Phase 2 – „Modellarchitektur“:

Hier werden die Anforderungen für ein zu entwickelndes Softwaremodul oder eine Softwarefunktion in funktional zusammenhängende Einheiten gekapselt, ein geeignetes Umgebungsmodell ausgewählt und das Testkonzept für die Komponente im Hinblick auf den Modelltest verfeinert. Kernaktivität dieser Phase ist jedoch die Definition und Umsetzung der Modellarchitektur durch den Modellierer und den Softwarearchitekten. Die entworfene Modellarchitektur inkl. der Schnittstellen wird durch das Strukturmodell abgebildet, das aus leeren, miteinander verbundenen Simulink-Subsystemen besteht.

Dieses Strukturmodell definiert somit bereits das Grundgerüst für die Softwarearchitektur, die später durch TargetLink-Subsysteme und -Funktionen definiert wird. Hinsichtlich der Modell- bzw. Softwarearchitektur stellt die ISO-Norm beispielsweise folgende Anforderungen (siehe ISO 26262-6, §7.4, Tab. 3):

- Hierarchische Struktur und geringe Komplexität der Softwarekomponenten
- Geringe Größe bzw. Komplexität von Schnittstellen
- Hohe Kohäsion innerhalb der SW-Komponenten
- Eingeschränkte Kopplung der SW-Komponenten

Diese Anforderungen werden durch unterschiedliche Vorgehensweisen erfüllt, wie (1) die Anwendung von Designpatterns für die Modellarchitektur, (2) durch Review der Architektur sowie (3) toolgestützt durch

zwischen Modell und Code (vgl. ISO 26262-6, Tabelle 10 und 13). Das Testkonzept des Moduls bzw. der Funktion wird in dieser Phase entsprechend erweitert.

Phase 3 – „Modelldesign“:

In dieser Phase werden die funktionalen Anforderungen für das Modul und die Funktionen im Strukturmodell ausmodelliert. Es entsteht das sogenannte Funktionsmodell. Zudem wird das Modul entsprechend dem Testkonzept im Model-in-the-Loop-Modus (MIL) als Simulink-Simulation in „double-precision“ gegen

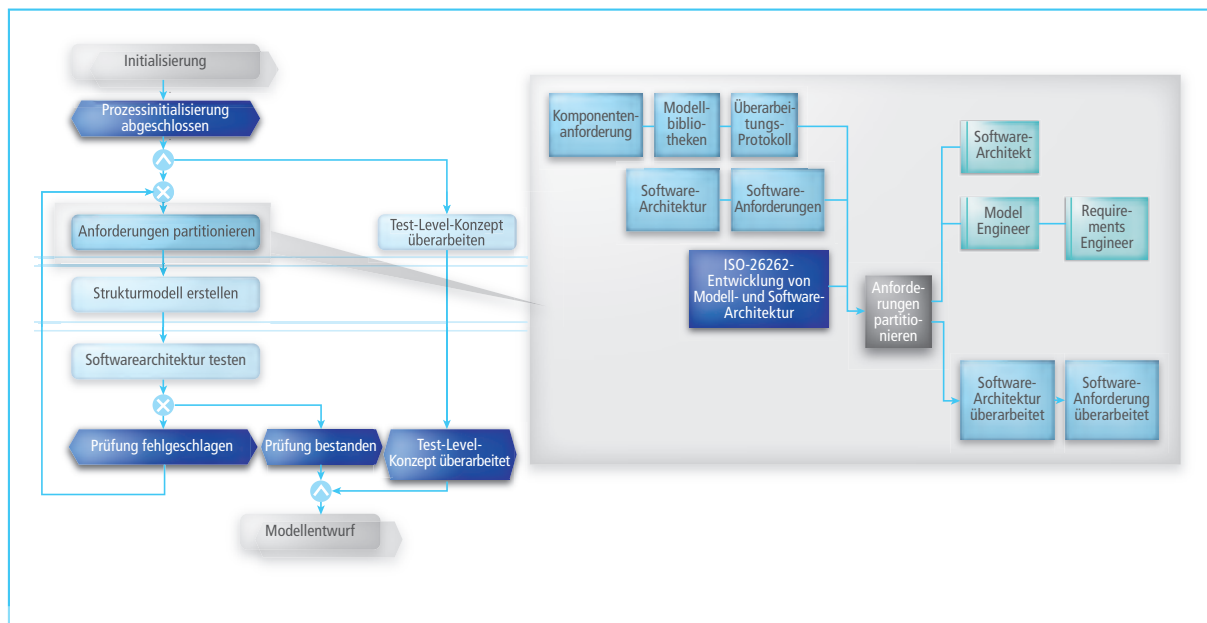
„Eine durchgängige Toolkette, bestehend aus spezialisierten Entwicklungswerkzeugen, ist für einen ISO-26262-konformen Entwicklungsprozess unabdingbar.“

Dr. Ingo Stürmer, MES

den Einsatz von M-XRAY zur Messung und Bewertung der Modellkomplexität. Die ISO 26262 stellt weiterhin Anforderungen an das Testen der Softwaremodule und -funktionen wie beispielsweise anforderungsbasiertes Testen, Schnittstellentest und Back-to-Back-Test

die zugehörigen Anforderungen getestet. Dies hat den Vorteil, dass bereits auf einer abstrakten Ebene nachgewiesen werden kann, ob das Verhalten des Modells den Anforderungen entspricht – bevor der Code überhaupt vorliegt (Ziel: frühzeitiges Testen). Das Funktionsmodell stellt

Abbildung 4: Beispielmodellierung der Phase 2 „Modellarchitektur“ mit Detaildarstellung des Schritts „Anforderungen partitionieren“.



somit eine ausführbare Spezifikation der funktionalen Anforderungen dar. Die ISO Norm stellt hierbei insbesondere Anforderungen an das Design der Softwaremodule und -funktionen, wie z.B. die Verwendung einer geeigneten Notation. Dabei erfüllt bereits die Verwendung von Simulink bzw. TargetLink die Anforderung an eine semi-formale Notation für ASIL-B- bis -D-Komponenten (vgl. ISO 26262-6, Tabelle 7). Neben diesen High-Level-Anforderungen an die Notation werden aber ganz konkrete Maßnahmen zur Fehlerprävention gefordert, wie beispielsweise die Vermeidung impliziter Typkonvertierungen (vgl. ISO 26262-6, Tabelle 8).

Phase 4 – „Codegenerierung“:

In der Implementierungsphase wird das Funktionsmodell mit den für die Codegenerierung mit TargetLink benötigten Informationen angereichert (Funktionspartitionierung, Zahlendarstellung, Datentypen, lokale/globale Variablen etc.). Projektabhängig kann diese Phase weitere Unterphasen beinhalten, wenn beispielsweise zuvor ein reines Simulink-Funktionsmodell verwendet wurde, das zunächst noch in ein TargetLink-Modell umgewandelt werden muss. Die Testergebnisse des generierten Codes als Software-in-the-Loop-Simulation (SIL) werden im Rahmen eines Back-to-Back-Tests gemäß ISO 26262 mit den Ergebnissen des MIL-Tests des Funktionsmodells verglichen. Sobald der generierte Code getestet vorliegt, werden die modellbasierten Entwicklungsaktivitäten wieder mit der codebasierten Softwareentwicklung zusammengeführt, d. h. beispielsweise, dass die Testaktivitäten für den Integrationstest mehrerer Module oder Funktionen bis zum Systemtest wieder den gängigen Vorgehensweisen für die Codeverifikation folgen. Hier kommen die oben genannten Testmethoden wie anforderungsbasiertes Testen, Schnittstellentest und Back-to-Back-Test wieder



zum Einsatz, die für den Integrationstest spezifische Überdeckungsmaße erfüllen sollen, wie Function Coverage und Call Coverage (siehe ISO 26262-6, Tabelle 15).

Prozessmodellierung der modellbasierten Entwicklung nach ISO 26262

Entwicklungsprozesse werden bei Siemens Drive Technologies einheitlich und werkzeuggestützt mit ARIS modelliert. Alle notwendigen Prozessschritte zur modellbasierten Entwicklung von Programmcode gemäß ISO 26262 werden so vollständig dokumentiert. Die Dokumentation erfolgt in mehreren sukzessiv aufeinander aufbauenden Dokumentationsstufen, die im Folgenden kurz beschrieben werden.

Die Phase „Modellarchitektur“ (Abbildung 4) des Entwicklungsprozesses lässt sich verkürzt als sog. ereignisgesteuerte Prozesskette darstellen (Abbildung 4, links). Erst die detaillierte Darstellung (Abbildung 4, rechts) zeigt, welche Rolle beispielsweise im Prozessschritt „Anforderungen partitionieren“ auf Basis welcher Dokumente den Prozessschritt im Sinne der Entwicklungsergebnisse und der Anforderungen der ISO 26262 durchführt.

Prozesshandbuch für die modellbasierte Entwicklung

Neben der grafischen Darstellung der Prozessschritte für die modellbasierte

Softwareentwicklung wird bei Siemens Drive Technologies der Entwicklungsprozess ausführlich in einem Prozesshandbuch erläutert. Das Handbuch beinhaltet definierte Prozessschritte mit detaillierten Handlungsanweisungen für alle beteiligten Personen. Zusätzlich enthält das Handbuch separate Kapitel zu den für die modellbasierte Entwicklung relevanten Themen wie z. B. modellbasiertes Testen, Design der Modellarchitektur und Anforderungen der ISO 26262 an den Softwareentwicklungsprozess.

Für die Beschreibung jedes einzelnen Prozessschritts wurde dabei eine Struktur gewählt, die die Informationen so aufbereitet, dass sich der Prozessanwender oder ein anderer Interessierter schnell orientieren kann und Informationen leicht findet. Die für das Prozesshandbuch gewählte Kapitel- und Unterkapitelstruktur beinhaltet für jede Prozessphase fünf Punkte (Abbildung 5), wie hier am Beispiel „Modellarchitektur“ erläutert:

1. Ziele:

Definiert, welche Ziele durch die Prozessphase erreicht werden sollen. Für die Phase „Modellarchitektur“ bedeutet dies die funktionale Dekomposition der Anforderungen in Teile, die durch eine Hierarchie von Softwarefunktionen realisiert werden können. Hier muss bereits die Softwarearchitektur berücksichtigt wer-



Abbildung 5: Struktur fundamentaler Themen und Prozessschritte für jede Prozessphase des ISO-26262-konformen Entwicklungsprozesses bei Siemens Drive Technologies.

den, die später durch das TargetLink-Modell definiert wird, um einen zeit- und aufwändigen Umbau des Funktionsmodells in ein Implementierungsmodell zu vermeiden. Ein weiteres Ziel ist die Verfeinerung des Testkonzepts für das Modul (Abbildung 4).

2. Eingangskriterien:

Definiert, welche Informationen und Arbeitsprodukte (Work Products) zu Beginn der Aktivität verfügbar sein müssen. Um die Ziele der Phase „Modellarchitektur“ zu erreichen, müssen z.B. die im übergeordneten Systementwicklungsprozess erstellte Komponenten-Anforderungs-Spezifikation und die initiale Version der Software-Anforderungs-Spezifikation vorliegen.

3. Aktivitäten (im Detail):

Beschreibt die Prozessschritte detailliert – was ist von wem durchzuführen, welche Arbeitsprodukte werden benötigt und welche erstellt. Bei der Beschreibung der einzelnen Aktivitäten wird, wie bereits im Kontext zu Abbildung 4 erläutert, insbesondere auf die ISO-26262-relevanten Anforderungen verwiesen. Diese beziehen sich beispielsweise auf die Softwarearchitektur, die Testmethoden, den Reviewprozess usw.

4. Arbeitsprodukte:

Enthält die Zusammenfassung aller in den einzelnen Schritten der Pro-

zessphase entstehenden Arbeitsprodukte (Work Products), also auch jener mit ISO-26262-Relevanz, wie z.B. das Test Level Konzept und die Überarbeitungsprotokolle.

5. Erfolgskriterien:

Führt die Kriterien auf, die erfüllt sein müssen, damit der Prozessschritt als

„Der vom TÜV SÜD anerkannte TargetLink-Referenzworkflow hat es uns erleichtert, einen ISO-26262-konformen modellbasierten Entwicklungsprozess rund um dSPACE TargetLink aufzusetzen.“

Dr. Heiko Zatocil, Siemens

„erfolgreich abgeschlossen“ bewertet und der nächste Schritte begonnen werden kann. Für die Modellarchitektur wären dies unter anderem, dass die Modellarchitektur im Strukturmodell umgesetzt ist und letzteres mit dem Umgebungsmodell erfolgreich integriert wurde.

Ein Glossar mit allen wichtigen Begriffen, speziell die der modellbasierten Entwicklung, erleichtert den Einstieg. Die einzelnen Kapitel des Handbuchs sind in sich geschlossen und können auch unabhängig von der Beschreibung des Prozesses gelesen werden. Ein weiterer wichtiger Bestandteil des Prozesshandbuchs ist die Erläuterung, wie und mit welchen Werkzeugen

die von der Norm empfohlenen Methoden für die modellbasierte Entwicklung umgesetzt werden sollen.

ISO-26262-konformer Prozess und Toolkette

Zentral für die praxisrelevante Definition und Umsetzung des Prozesses war die externe Unterstützung durch MES: Hier wurde mit dem Wissen, was funktioniert und was nicht, wertvolles Know-how von extern eingeholt. Ebenso trugen Erfahrungen aus unterschiedlichen Siemens-Projekten zum Erfolg bei. Dazu wurden Experten für Prozessmodellierung eingebunden, die den umfassenden Systementwicklungsprozess bei Siemens gestalten. Dabei wurde deutlich, dass eingeführte Prozesse, die transparent organisiert und für sicherheitsrelevante Aspekte optimiert sind, schon elementare Anforderungen eines ISO-26262-gerechten Vorgehens erfüllen können.

In gleicher Weise müssen die verwendeten Software-Tools geeignet sein, hohe Produktqualität und ISO-Konformität zu gewährleisten. So ist der Code-Generator TargetLink bereits für den Einsatz in sicherheitsrelevanten Entwicklungsprojekten bis hin zu ASIL D durch den TÜV SÜD qualifiziert. Grundlage hierfür ist die Einhaltung des von dSPACE veröffentlichten TargetLink-Referenzworkflows, der die Best Practices für die Modell- und Codeabsicherung für TargetLink beschreibt. Dieser Workflow wurde im Siemens-Entwicklungsprozess integriert. Von MES wurde der Model Examiner (Functional Safety Solution) zur Richt-

linienkonformität der Modelle und M-XRAY zur Messung und Beherrschung der Modellkomplexität eingesetzt. Damit wurden bereits durch den Einsatz von Werkzeugen wichtige Anforderungen der ISO-Norm an die modellbasierte Entwicklung erfüllt (vollständige Abdeckung von Teil 6, §5.4.7, Tabelle 1). Es hat sich gezeigt, dass die bei Siemens eingesetzte Werkzeugkette geeignet ist, den in der Norm definierten Prozess projektspezifisch mit Leben zu füllen. Die bei Siemens entwickelte Werkzeugkette realisiert darüber hinaus bidirektionale Traceability von den Anforderungen über den hier beschriebenen Code-Generator bis in den Test.

Frühere Tests und deutlich verbesserte Entwicklungssicherheit

Mit der neuen ISO 26262 wird die modellbasierte Entwicklung als qualitativ hochwertige Vorgehensweise bei der Entwicklung sicherheitsrelevanter Software für das Automobil hervorgehoben.

Die Prozesserweiterung der codebasierten um die modellbasierte Entwicklung bringt wesentliche Vorteile, da die Software frühzeitiger und mit besserer Methoden- und Werkzeugunterstützung abgesichert werden kann. Eine firmenspezifische Anpassung der internen Prozesse an die Anforderungen der Norm ist hierfür notwendig. Siemens Drive Technologies hat bereits erste sicherheitsrelevante Softwarekomponenten nach dem neuen Vorgehensmodell erfolgreich entwickelt. Die modellbasierte Entwicklung wird neben der klassischen Entwicklung eine immer wichtigere Rolle einnehmen und findet durch die Einhaltung ISO-26262-konformer Prozessschritte zunehmende Verbreitung. ■

David Brothanek, Dr. Martin Jung, Verena Jung, Michael Krell, Reinhard Pfundt, Dr. Elke Salecker, Dr. Ingo Stürmer, Dr. Heiko Zatocil



David Brothanek
David Brothanek ist Senior Manager und Spezialist mit Schwerpunkt Prozessmodellierung, -optimierung und -implementierung bei der Headframe IT GmbH in Essen, Deutschland.



Dr. Martin Jung
Dr. Martin Jung ist Leiter des Bereichs Software- und Systementwicklungsberatung der develop group in Erlangen, Deutschland, und hält einen Lehrauftrag für Softwarearchitektur an der FAU Erlangen, Deutschland.



Verena Jung
Verena Jung ist Teamleiterin für den Bereich Integration und Test und unter anderem verantwortlich für die Koordination der Testaktivitäten in der Software- und Komponentenentwicklung bei der Siemens AG in Erlangen, Deutschland.



Michael Krell
Michael Krell ist Functional Safety Manager bei der Siemens AG in Erlangen, Deutschland. Er unterstützt bei der Umsetzung der ISO-Anforderungen in den Projekten.



Reinhard Pfundt
Reinhard Pfundt ist als Software-Manager verantwortlich für die Planung und Koordination der Software für Frequenzrichter, DC/DC-Wandler und On-Board-Ladegeräte bei der Siemens AG in Erlangen, Deutschland.



Dr. Elke Salecker
Dr. Elke Salecker ist Senior Software Consultant bei der Model Engineering Solutions GmbH in Berlin, Deutschland. Sie ist Expertin für modellbasierte Softwareentwicklung nach ISO 26262 und unterstützt Kunden bei der Prozessdefinition und -implementierung.



Dr. Ingo Stürmer
Dr. Ingo Stürmer ist Gründer und CEO der Model Engineering Solutions GmbH in Berlin, Deutschland. Er ist anerkannter Spezialist für Entwicklungsprozesse mit dSPACE TargetLink und hilft Kunden, ihre modellbasierten Entwicklungsprozesse zu optimieren und firmenspezifische Toolketten nach ISO 26262 zu qualifizieren.



Dr. Heiko Zatocil
Dr. Heiko Zatocil ist Leiter der Funktionsentwicklung bei der Siemens AG in Erlangen, Deutschland, und treibt dort die modellbasierte Softwareentwicklung maßgeblich voran. Er initiierte und koordinierte die Erstellung des beschriebenen Prozesses.