5 Systemengineering in der Produktrealisierung

Grundsätzlich wird hier der Ansatz verfolgt, dass die Realisierung auf Basis der Spezifikation umgesetzt wird. Das heißt, die zu realisierenden Elemente entsprechen den technischen Elementen, und wenn die bisherigen Analysen und Verifikationen alle korrekt und hinreichend waren, dann sollten auch die integrierten Elemente wie spezifiziert funktionieren und die geforderten Eigenschaften bezüglich Performance und anderer Funktionalitäten den geforderten Erwartungen entsprechen. Aus den Anforderungen, wann das Produkt und wie es realisiert wird, ergeben sich die Zeitpunkte, wann bestimmte Sicherheitsaktivitäten durchgeführt werden müssen. Die Realisierung des Produktes ist in den meisten Darstellungen der V-Modelle der Boden des Vs.

5.1 Produktrealisierung

Eine Designbeschreibung ist noch kein realisiertes Produkt. Wenn eine Platine bestückt wird, wenn verschiedene Mechanikkomponenten zusammengefügt werden, wenn verschiedenen Softwareelemente integriert werden, diese generiert oder realisiert werden, oder Komponenten basierend auf unterschiedlicher Technologie zusammengeführt oder integriert werden, wird dies auf das korrekte Verhalten oder Funktionieren Einfluss haben. Besonders die Aspekte der Realisierung selbst bis hin zur Integration von Elementen oder Komponenten verschiedener Technologien ist in der ISO 26262 nur partiell adressiert. Die meisten Sicherheitsstandards stellen an die Produktrealisierung selbst gar keine Anforderungen. Es ist auch weitgehend freigestellt auf welche Art und Weise die Spezifikation ausgearbeitet ist. Die ISO 26262 fordert im Rahmen der Verifikation des Designs, dass das Produkt oder die Komponente unter anderem vollständig und konsistent spezifiziert ist. Dies kann durch Anforderungen, Architektur (Blockdiagramme, Verhaltensdiagramme, Modelle etc.) oder Designdokumente (Designspezifikation, Stücklisten, Zeichnungen etc.) erfüllt werden.

5.1.1 Produktdesign zur Realisierung

Fast alle heutigen Entwicklungsstandards gehen davon aus, dass ein Produkt auf Basis der Spezifikation entsteht. Hierbei ist jedoch darauf zu achten, dass Spezifikation mindestens aus zwei Arbeitsprodukten besteht, nämlich der Anforderungs- und der Designspezifikation. In der klassischen Mechanikentwicklung stand am Ende des Entwicklungsprozesses die Konstruktionszeichnung. Diese erhielt nach einigen Prüfungen (Verifikation) und der Fertigstellung der Produktionseinrichtung eine Freigabe zur Produktion. Setzt man voraus, dass das Design auf Basis der Anforderungen entstanden ist und daraufhin auch verifiziert und validiert wurde, dann gilt dieser Weg auch noch heute für software-basierende Systeme oder Produkte. Den Prozess hin zur Realisierung kann man als abfallenden Ast eines oder mehrerer Vs beschreiben, aber auch über Spiralen (Musterzyklus in der Automobilindustrie) oder ein Wasserfallmodell, welche auch als Grundlage für die Reifegradbestimmung für Produkte, aber auch Projekte und Prozesse dienen können. Wenn das Projekt oder der Prozess nicht vollständig, konsistent und nachvollziehbar bis hin zur Produktrealisierung abgearbeitet ist, wird man immer mit systematischen Fehlern im Produkt rechnen müssen. Welchen Einfluss solche Projekt- oder Prozessfehler auf die Eigenschaften des Produktes haben, lässt sich allgemein nicht beurteilen oder prognostizieren. Daher wird in der ISO 26262 der aufsteigende Ast des Vs genutzt, um durch systematisches Integrieren und weitere Tests zur Validation und Verifikation zu einer Produktbeurteilung zu kommen. Die Realisierung von Hardwareteilen, egal ob Mechanik, Hydraulik, Elektrik oder Elektronik, hängt dann sehr stark von den Produktionsmitteln und deren Reifegrad bezüglich der Serienfertigung ab. Die Software wird mehr in einer Laborumgebung realisiert. Um in Analogie zur Hardwareproduktion auch hier die Umsetzung und Erfüllung von nichtfunktionalen Anforderungen wie Sicherheit, Qualität und Zuverlässigkeit und so weiter zu gewährleisten, spricht man oft auch von der sogenannten Softwarefabrik (Software-Factory). Die Betrachtungsweisen sind im Grunde genommen tatsächlich sehr ähnlich.

5.1.2 Mechanik

Die Eigenschaften (Merkmale, Fähigkeiten) wurden im Rahmen der Anforderungsentwicklung identifiziert. Die technischen Elemente können nun durch Normteile (Schrauben, Muttern, Stecker etc.) oder für den Anwendungsfall bestimmte Mechanikteile realisiert werden. In allen Fällen gilt es, die als wichtig identifizierten Eigenschaften über den vollständigen Spezifikationsraum zu sichern. Das heißt, bei der Schraube und der Mutter kann der Gewindegang als relevant angesehen werden,

jedoch werden auch die anderen Eigenschaften von Bedeutung sein. Hat man kein Werkzeug, weil man einen passenden Schraubenschlüssel für die Schraube benötigt, dann kann dies für die Sicherheit tolerierbar sein. Würde jedoch der Anzugsmoment für die Schraube als wichtige Eigenschaft identifiziert, dann wird die Wahl des Werkzeugs oder die Überwachung der Arbeit mit dem Werkzeug womöglich zu einer Sicherheitsaktivität. Die Produktion (zum Beispiel die Realisierung der Verbindung durch Einschrauben der Schraube) muss in der Linie überwacht werden, um zu zeigen, dass die Anforderungen umsetzbar sind (meist bei den ersten Prototypenbauten) und auch korrekt umgesetzt werden. Für sicherheitsrelevante Eigenschaften von den Spitzenprodukten werden meist „Besondere Merkmale" festgelegt. Diese werden oft in der Serienproduktion für jedes einzelne Produkt geprüft und die Prüfergebnisse werden in Datenarchiven abgelegt.

Traditionell hat man vier Musterphasen in der klassischen Entwicklung der Automobilindustrie gesehen. Für diese Muster werden folgende Ziele bei mechanischen Teilen betrachtet:

- A-Muster: belegen die Geometrie, Haptik oder Funktionserfüllung
- B-Muster: stellen Funktionalität und Dauerlauffähigkeit auf dem Prüfstand und im Prototyp sicher
- C-Muster: stellen Funktionalität, Dauerlauffähigkeit, Kompatibilität und Integrationsfähigkeit in der Zielanwendung (Motor, Fahrzeug) sicher
- D-Muster: (Erstmuster-) freigabefähiges Muster

Allgemein werden auch Entwicklungsprozesse in Phasen beschrieben, die sich an den historischen Musterphasen orientieren.

Das A-Muster (Konzeptphase) wird oft bereits in sehr frühen Phasen der Entwicklung betrachtet und mit dem Angebot des Zulieferers bereits eingefordert.

Das B-Muster (Designphase) ist oft das Muster, an dem das Design verifiziert und validiert (DV) wird, das heißt, alle Eigenschaften sollen bereits gesichert sein. Als Konsequenz daraus sollten alle Anforderungen verifiziert vorliegen, die Architektur und das Design müssen ebenfalls analysiert und verifiziert sein. Für die Produktion müssen die Konzepte detailliert bereitstehen, sodass man üblicherweise die Prozess-FMEA vorliegen hat und die sicherheitsrelevanten Aktivitäten bereits auch in der Produktion einplant. Sämtliche Sicherheitsfunktionen, Sicherheitsmechanismen und Merkmale müssen also korrekt implementiert, realisiert, getestet und verifiziert sein.

Das C-Muster (Industrialisierungsphase) soll die Integrationsfähigkeit und Kompatibilität in der Zielumgebung zeigen können, damit müssen zum Beispiel alle Toleranzen von Zulieferteilen und die möglichen Toleranzen, die sich aus der Produktion ergeben, gesichert sein. Es wird daher oft gefordert, dass das C-Muster bereits auf der

Serienproduktionsanlage produziert wird. Bei Neuprodukten sind die Anlagenteile jedoch noch nicht verkettet, sodass nicht alle Schnittstellen in der Produktion als abgesichert betrachtet werden können.

Das D-Muster (Serienphase) muss in bestimmter Stückzahl, in einer dem Serienstand entsprechenden Produktionsanlage und in einer der Serie entsprechenden Taktung der Arbeitsschritte produziert werden. Das produzierte Muster ist dann allgemein die Grundlage des Fahrzeugherstellers für die Serienfreigabe des Produktes.

Als Schluss für die Sicherheit müssten alle Produktmerkmale, die als sicherheitsrelevant identifiziert sind, bereits in der B-Musterphase mit ihren Eigenschaften gesichert sein. In heutigen Entwicklungszyklen ist dies sehr selten möglich, jedoch leiten sich aus den Eigenschaften der mechanischen Komponenten oft erst die Anforderungen und Eigenschaften für die Elektronik und Software ab. Auch für die mechanischen Teile der elektronischen und elektrischen Elemente, wie Stecker, Gehäuse, Platine und so weiter, ist diese Vorgehensweise sehr zu empfehlen.

5.1.3 Elektronik

Grundsätzlich wird die Elektronik genau wie die Mechanik auch auf Basis der Designdokumente realisiert. Die Elektronik wird auch wie die Mechanik in verschiedenen Musterständen produziert, das heißt, die Beschreibung der Musterphasen für die Elektronik ist vergleichbar. Wie bereits bei der Mechanik beschrieben, sind viele Designparameter für die Elektronik abhängig von mechanischen Teilen wie Platine, Stecker und Gehäuse. Das heißt, diese Toleranzen oder tolerierbaren Abweichungen sind Grundlage für die Auslegung der Elektronik. Gerade bei geometrischen Eigenschaften sind viele Eigenschaften zu berücksichtigen.

Das Gehäuse muss so konstruiert sein, dass es ins Fahrzeug passt, Schutz gegen Feuchtigkeit und Schmutz gewährleistet wird, Kabel zugeführt werden können, EMV-Anforderungen erfüllt und die innen entstehenden Wärmemengen an die Umgebung abgegeben werden können. Bei der Realisierung der Elektronik wird eine wirkliche Abgrenzung zur Mechanik nicht darstellbar sein. Daher wird das testen der Muster und der Serienteile einer der wesentlichen Sicherheitsaktivitäten sein.

5.1.4 Software

Software wird im Gegensatz zur Hardware nicht in einem Werk produziert, sondern in der Entwicklung realisiert. Wo die Softwarerealisierung beginnt, wird in verschiedenen Standards unterschiedlich interpretiert. Die ISO 26262 geht davon aus, dass

die SW-Unit das kleinste Element in der Architektur ist und dass zum Beispiel das C-File über Codegeneratoren oder Compiler dann diese kleinste Einheit darstellt. Der Binärcode ist das Äquivalent des Quellcodes, der in den Mikrokontroller nach dem Linken übertragen wird. In diesem Vorgang können noch beliebige Fehler entstehen, sodass der implementierte Code nicht dem Quellcode entspricht. Oft wird nach einem sicheren Compiler gefragt, jedoch ist auch bei der Zuführung des Quellcodes an den Compiler, bei der Definition der Rechnerumgebung sowie beim Linken ein Potential für Fehler vorhanden. Weiter wird bei der modellbasierenden Softwareentwicklung oft bereits in einem Softwaremodul, welches aus mehreren SW-Units besteht, über Codegeneratoren der Binärcode generiert. Spätestens hier stellt sich die Frage, wie die SW-Units miteinander interagieren und ob die Schnittstellen zwischen den SW-Units nicht doch durch die Rechnerumgebung negativ beeinflusst werden können.

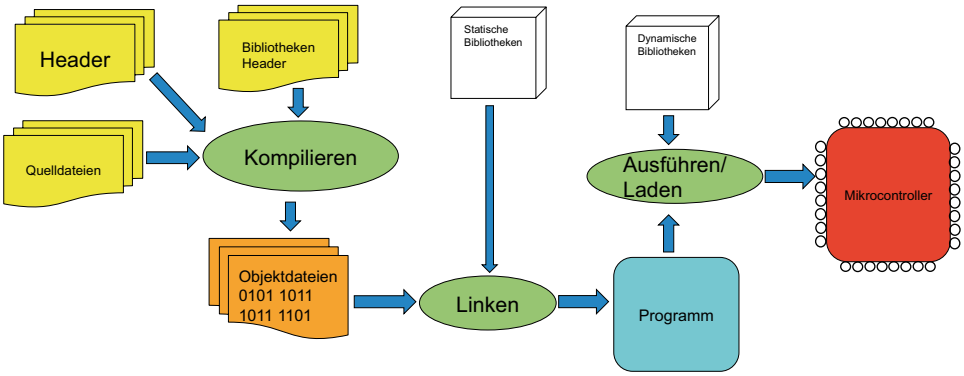

Bild 5.1 Prinzip der C-Codierung

Neben dem Kompilieren ist das Linken, Einbinden der Bibliotheken (Header, statische und dynamische Bibliotheken) und das Laden des Programms ein Vorgang mit möglichen Fehlerquellen hin zum ausführbaren Code im Mikrokontroller. Auch die Editoren oder die Aufbereitung der Dateien (zum Beispiel das Preprozessing des Quellcodes) können zu Fehlereinflüssen führen. Die Werkzeuge, die diese Aktivitäten unterstützen, können nicht immer gegen mögliche Fehler abgesichert werden. Aus diesem Umstand entstand der Bedarf der Tool-Qualifizierung in Teil 8 Kapitel 12 der ISO 262626. Später stellte man fest, dass auch andere Tools Fehler in sicherheitsrelevanten Funktionen bewirken können. Geht man wirklich mit der Architektur herunter bis auf die SW-Unit und kann die entsprechenden SW-Units durch Coverage Tests vollständig testen, so kann der negative Tool-Einfluss ausgeschlossen werden, sofern ein Hardwareeinfluss durch den Rechner auch hinreichend gewährleistet ist. In der Praxis werden auf der Hardwareseite, zum Beispiel durch falsche Handhabung der Tools sowie unzureichende Testmöglichkeiten, hier einige Punkte offen gelas-

sen. Für ASIL C und D Software, wozu die ISO 26262 neben Plausibilisierung oder gar Diversität (ASIL D) eine Daten- und Steuerflussanalyse mit „Doppelplus (++)“ fordert, wird man diese Lücken meist nur durch redundante Implementierungen kompensieren können.