

Requirement Engineering in der Steuergerätesoftwareentwicklung

Abdulvahap User

IAV GmbH, Gifhorn

Abstract: Mit dem zunehmenden Einzug der Elektronik ins Fahrzeug nimmt die Komplexität der Steuerungsstrukturen exponentiell zu. Durch diese Komplexität wird das Entwicklungsrisiko der Komponente Steuergerät sehr hoch. In vielen Fällen hat das Requirement Engineering großen Einfluss auf kritische Projektverläufe. Der folgende Praxis- und Erfahrungsbericht aus diesem Bereich soll die Prozessverbesserungspotentiale unter Zuhilfenahme der modellbasierten Funktions- und Softwareentwicklung aufzeigen.

1 System Engineering

Um die zuvor genannte Komplexität zu beherrschen, setzt die Steuergeräteentwicklung auf ein straffes und sehr gut organisiertes System Engineering auf. Im Besonderen liegt hierbei das Augenmerk auf der Verteilung der Steuergerätefunktionen, der Beschreibung der Schnittstellen und der Ersatzreaktionen bei Ausfall von Signalen und Komponenten. Erfahrungsgemäß haben Entscheidungen auf dieser Ebene erhebliche Auswirkungen auf die Architektur der Steuer- und Regelfunktionen im Steuergerät. Aus diesem Grund bringt der Einsatz eines Change Managements (Änderungsmanagements) und eines Informations-Netzwerkes erhebliche Zeit- und damit Kostenminimierung. Das Informations-Netzwerk gewährleistet einen zuverlässigen Informationsaustausch zwischen den Entwicklungsbeteiligten.

2 Modellbildung und Requirement Management

Für die Beschreibung der inneren Struktur eines Steuergerätes hat sich die Funktionsmodellierung bewährt und durchgesetzt. Als Methode wird heute im Automotive Bereich vorwiegend die Strukturierte Analyse (SA) mit den Tools ASCET-SD, Matlab/Simulink/Stateflow und Statemate eingesetzt. Wenn man vom Grundgedanken der Basiskomponente "Modell" im Entwicklungsprozess ausgeht, so erscheinen dessen Qualitätsmerkmale in Abb. 1, umso wichtiger je detaillierter die Funktion ausmodelliert wird.

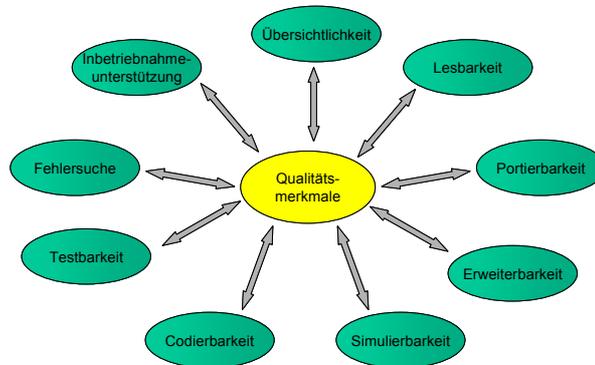


Abb. 1 : Modellierungskriterien

Ausgehend von den Zielen der Modellbildung unterteilen wir diese in drei Phasen. In der Entwurfsphase sprechen wir vom **Anforderungsmodell**. Dieses hat die Aufgabe, Funktionalitäten auf einer hohen Abstraktionsebene mit dem Auftraggeber zu diskutieren. Als Hilfsmittel können in dieser Phase aber auch Use Cases und Sequence Diagram eingesetzt werden. In der zweiten Phase wird das **Funktionsmodell** neu erstellt und zur Verifikation der Funktionalität eingesetzt. Es dient anschließend als Basis für das **Implementierungsmodell**, d. h. in dieser Phase wird der Detaillierungsgrad des Modells erneut erhöht und anschliessend auf die Zielhardware angepasst. Sämtliche Modelle werden unter dem Requirement Management Tool DOORS eingepflegt und mit anderen Anforderungsdokumenten verknüpft (verlinkt).

Von großer Bedeutung sind in jeder Phase der Modellbildung die Lesbarkeit sowie die Diskussionsfähigkeit des Modells mit dem Funktionsverantwortlichen. Letzterer ist oftmals kein Software-Entwickler.

Um dieser Anforderung gerecht zu werden, wurde bei der IAV der Stateflow-Viewer (SVI) (s. Abb. 2) entwickelt. Dieses Tool ermöglicht es, Beschriftungen von Modellelementen so zu filtern, dass wahlweise beschreibender Klartext, applikationsrelevante Informationen oder mathematisch korrekte Beschreibungen für die Programmierung sichtbar sind. Je nach Anwenderkreis kann nichtrelevante Information ausgeblendet werden.

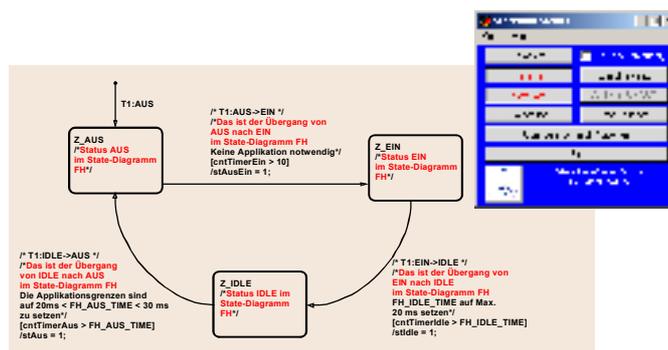


Abb. 2 : Modellierungstool Stateflow-Viewer

Auf der Kommentarebene ist somit das Modell auch mit einem reinen Funktionsentwickler diskutierbar, denn detaillierte Reviews des entstandenen Modells sind ein sehr wirksames Instrument, um in einer frühen Phase des Entwicklungsprozesses Spezifikations- und/oder Modellierungslücken bzw. Fehlfunktionen zu entdecken. Spätestens nach einem Modell-Review (Abb. 3) ist der Informationsgehalt des Modells größer als die der Spezifikationen.



Abb. 3 : Modell-Reviews

Umfasst ein Team eine größere Anzahl von Mitarbeitern, evtl. auch standortübergreifend, so müssen neben der generellen Anwendung von Modellierungsrichtlinien weitere Maßnahmen zur einheitlichen Vorgehensweise, wie die Nutzung von Namenskonventionen und Abkürzungslisten, Anwendung finden. Um den Informationsfluss zum Gesamtmodell besser zu kanalisieren, werden zusätzliche Dokumente wie Offene Punkte Listen (OPL), Change Request Listen (CRQ) sowie ein Data Dictionary (DD) benötigt. Letzteres beschreibt Signale, Funktionen und Module.

Zusätzlich entsteht zur Projektlaufzeit weiteres undokumentiertes Wissen. Dies wird dauerhaft durch eine detaillierte Verlinkung zwischen Lasten-/Pflichtenheft und den entsprechenden Modellabbildungen unter DOORS festgehalten. Weiterhin kann man diese Detailverlinkung anwenden auf Code-Zeilen, Testfälle, Testresultate und Dokumentationen.

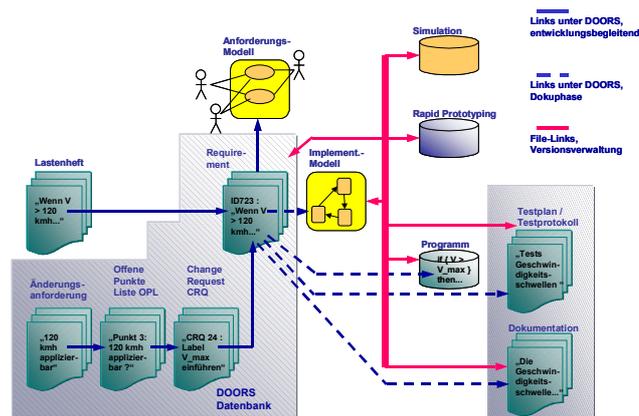


Abb. 4 : Verlinkung von Projektinformationen

In der IAV GmbH werden DOORS und Matlab/Simulink/Stateflow für das Requirement Engineering eingesetzt. Die Kopplung erfolgt mittels der Matlab-Toolbox RMI (Requirement Management Interface), die von der IAV problemspezifisch modifiziert wurde. RMI ermöglicht ein Abbild des Modells im Anforderungsmanagementsystem (DOORS). Abb. 4 stellt die detaillierte Verlinkung des Systems dar.

Damit bleibt das Projekt auch Jahre nach Projektende wartbar. Ein Austausch von Bearbeitern während der Projektlaufzeit ist weniger kritisch. Somit hat man zu jedem Zeitpunkt im Projekt einen sehr guten Überblick über den aktuellen Projektstand. Erfahrungsgemäß ändern sich noch während der Entwicklungsphase die Anforderungen. Ursachen können Interpretationsfehler der Spezifikation, Aufdeckung von Widersprüchen der Funktionalität während der Modellierungsphase oder der Austausch von Aggregatteilen sein. Es kommt also eine zeitliche Abfolge von Änderungsanforderungen und deren Bearbeitung im Gesamtsystem als zusätzliche Dimension hinzu.

Dieses Problem lässt sich mit dem Einsatz von DOORS und einem Change Management System wie StarTeam lösen (s. Abb. 5). Aufgrund des aufgeführten Bearbeitungsprozesses ist es möglich, zu jedem Zeitpunkt des Projektverlaufes über sogenannte Baselines in DOORS und Modellversionen in StarTeam, Anforderungen bzw. Änderungen einer Modellversion zuzuordnen.

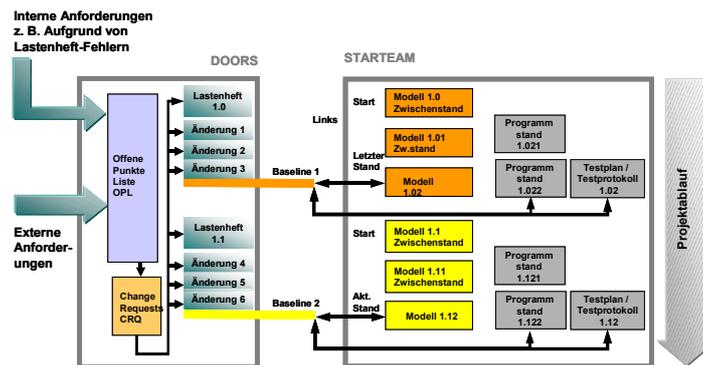


Abb. 5 : Versionierung von Projektinformationen

Mit der konsequenten Anwendung des Requirement Engineering im Entwicklungsprozess und dem Einsatz der vorgestellten Toolkette lässt sich zuverlässig ein transparentes sowie wartbares Modell erzeugen. Aufgrund der langjährigen Erfahrung der IAV GmbH im Bereich der Entwicklung von Automotive-Seriensoftware wird verdeutlicht, dass eine solche Verfahrensweise wesentlich zur Prozesssicherheit und damit zur Verbesserung von Termintreue und Kostenvorhersagbarkeit beiträgt.

Literaturverzeichnis

- [Sc03] Schultze, A.: Automotive Electronics, Requirement Engineering – Effektives Verbesserungspotenzial bei der Entwicklung von Steuergerätesoftware, 2003.
- [Au03] User, A.: Deutsches Zentrum für Luft- und Raumfahrt e.V. - Requirement Engineering in der Steuergeräteentwicklung, 2003