

Requirements Engineering Erfahrungen in Projekten der Automobilindustrie

Frank Houdek

DaimlerChrysler AG

Postfach 23 60

89013 Ulm

frank.houdek@daimlerchrysler.com

Zusammenfassung

Die Bedeutung von Software im Automobil hat in den letzten Jahren stetig zugenommen — und dieser Trend wird sich fortsetzen und verstärken. Ein bedeutender Teil der Software wird dabei nicht vom Automobilhersteller sondern von Zulieferern entwickelt. Auf Seiten des Herstellers bleiben vor allem die Aufgaben der Spezifikation und der Abnahme.

Der Artikel fasst kurz einige der zentralen Herausforderungen im Bereich Requirements Engineering (RE) in der Automobilbranche zusammen und skizziert für ausgewählte Aspekte aktuelle Ansätze.

1 Situation und Herausforderungen

Seit den 70er Jahren findet Software zunehmend den Weg ins Fahrzeug. Wurden erst einige wenige Funktionen mit Hilfe von Software realisiert, können wir heute ein Automobil als ein hochgradig vernetztes System ansehen. In Premium-Fahrzeugen finden sich bis zu 80 untereinander vernetzte Steuergeräte, auf denen mehrere Megabyte Programmcode arbeiten. Ein Ende ist hier nicht in Sicht. Mittelfristig ist mit einer Vervielfachung des Programmcodes und damit auch der Komplexität zu rechnen.

Im Folgenden wollen wir die aktuelle Situation in diesem Umfeld etwas näher analysieren und einige zentrale Herausforderungen für das Requirements Engineering ableiten. Im Abschnitt 2 werden wir uns einigen Aspekten näher zuwenden und aktuell bearbeitete Ansätze in diesem Bereich betrachten.

Im Falle von DaimlerChrysler (aber auch anderen Automobilherstellern) wird ein wesentlicher Anteil der im Fahrzeug enthaltenen Software nicht selbst entwickelt. Vielmehr übernimmt DaimlerChrysler die Rolle des Systemintegrators und beauftragt Dritte mit der Entwicklung der Hard- und/oder Software für die einzelnen Steuergeräte. Somit werden die Lastenhefte, mit deren Hilfe die eigentliche Entwicklung beauftragt wird, zu einem zentralen Entwicklungsergebnis auf Seiten der Automobilhersteller. Die Qualität der Lastenhefte hat entscheidenden Einfluss auf die Qualität der gelieferten Komponenten und die anfallenden Aufwände für Abnahme und Systemintegration. Daher ist es beispielsweise unerlässlich, Schnittstellen und andere Funktionsdetails bereits im Lastenheft bis „auf das letzte Bit“ zu spezifizieren.

Aufgrund ihres hohen Detaillierungsgrades tendieren die erstellten Lastenhefte dazu, sehr umfangreich

zu werden. Dokumente mit mehreren hundert Seiten sind keine Seltenheit. Das korrekte und konsistente Fortschreiben dieser Dokumente wird so aufgrund des Umfangs zu einer anspruchsvollen Aufgabe. Standard-Textverarbeitungssysteme, die hierfür heute noch häufig eingesetzt werden, sind mit dieser Aufgabe in der Regel überfordert. Eine geeignete Werkzeugunterstützung erscheint hier unerlässlich. Exemplarisch sei hier DOORS erwähnt, das in der Automobilindustrie bereits in einigen Projekten erfolgreich eingesetzt wurde.

Die Entwicklung von Fahrzeugsteuergeräten startet im Regelfalle nicht auf der vielbesungenen „grünen Wiese“. Vielmehr setzt die Entwicklung eines neuen Steuergeräts auf existierenden Produkten und damit vorhandenen Lastenheften auf. Die Anforderungen für das neue Steuergerät ergeben sich aus einer Weiterentwicklung der existierenden Anforderungen und deren Anpassung an die neuen Gegebenheiten. An dieser Stelle ergibt sich die Forderung, sich systematisch mit der Wiederverwertung von Anforderungen auseinanderzusetzen und geeignete Verfahren hierfür bereit zu stellen.

Verkompliziert wird diese Situation noch dadurch, dass ein Steuergerät für eine Baureihe in der Regel in verschiedenen Ausprägungen existiert (z.B. aufgrund verschiedener Märkte oder Ausstattungsvarianten). Die Vielfalt betrifft natürlich jedes Steuergerät, so dass ein einzelnes mit einer Vielzahl an variierenden Nachbar-Steuergeräten interagieren muss. Dies hat natürlich wieder einen Einfluss auf die Wiederverwertungsbestrebungen.

An der Entwicklung von Steuergeräten sind eine Vielzahl von Menschen beteiligt. Damit alle sinnvoll an der Entwicklung mitarbeiten können, müssen die Lastenhefte so verfasst sein, dass sie für alle Beteiligten verständlich sind.

2 Requirements Engineering Prozess

Das Ergebnis des Requirements Engineering Prozesses ist in der Regel ein Lastenheft, in dem detailliert Hard- und Softwareanforderungen an ein Fahrzeugsteuergerät dokumentiert sind. Aufgrund der wachsenden Komplexität der betrachteten Systeme und dem damit einhergehenden Anwachsen der Dokumente wird es zunehmend schwierig, detaillierte Lastenhefte „en block“ zu erstellen. Die Betrachtung verschiedener Abstraktionsebenen wird folglich notwen-

dig. Abbildung 1 beschreibt ein dreistufiges Abstraktionskonzept, das sich bereits in der Praxis bewährt hat.

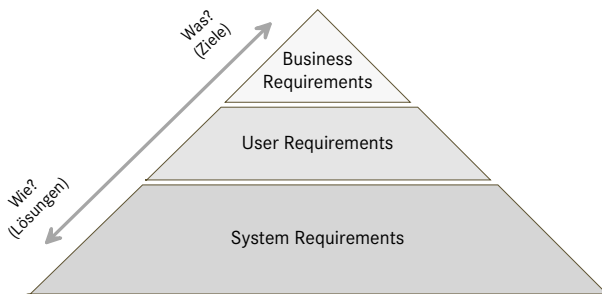


Abbildung 1: Abstraktionsebenen für Anforderungen

Nachfolgend werden die verschiedenen Ebenen und deren Zusammenspiel näher betrachtet.

Business Requirements beschreiben Anforderungen im Wesentlichen als Ziele, die mit einer Funktion erreicht werden sollen. Die Abstraktionsstufe ist hier so gewählt, dass die Beschreibung beispielsweise als Diskussionsgrundlage im Management verwendet werden kann („*Management-Folie*“). Diese Abstraktionsebene kann in einem *Vision and Scope Dokument* beschrieben werden.

Beispiel: Die neue Z-Klasse soll ein verbessertes Interaktionskonzept mit dem Fahrer aufweisen, welches intuitiv bedienbar ist.

User Requirements beleuchten die betrachtete Funktion aus einer externen Sicht, d.h. es wird hier festgelegt, wie sich das System im Zusammenspiel mit anderen Akteuren (z.B. dem Fahrer oder anderen Steuergeräten) verhalten soll. Zudem werden die wesentlichen Bestandteile festgelegt. In der Praxis haben sich hier zwei ergänzende Beschreibungsformen bewährt: Zum einen Use Cases zur Beschreibung des Funktions-Verhaltens, zum anderen Feature-Listen zur Beschreibung des Gesamtbildes.

Beispiel: (Feature:) Scroll-Wheel im Lenkrad ohne Klick. (Use Case:) Fahrer dreht am Rad. Die Selektion im Display verändert sich entsprechend der Drehrichtung.

System Requirements spezifizieren detailliert die betrachtete Funktion. Diese Abstraktionsebene entspricht den aktuell verwendeten Lastenheften.

Beispiel: Das Signal SC_WHEEL_UP (gesendet auf dem CAN Bus) bewirkt ein Verändern der Selektion im Display um eine Position nach oben. Die maximale Veränderungsrate beträgt 2 Einträge pro Sekunde. Darüber hinausgehende Signale werden ignoriert.

Über die Nachvollziehbarkeit der Verfeinerung von Business Requirements hin zu System Requirements ist zudem ein wesentlicher Grundstein für eine gezieltere Wiederverwertung von Anforderungen gelegt.

Soll ein bereits beschriebenes Feature auch in einer anderen Baureihe genutzt werden, so können im Grunde alle zugehörigen Detailanforderungen mit übernommen werden.

Die Verfeinerung von einer Abstraktionsebene zur nächsten ist nicht immer naheliegend. Vielmehr liegen zwischen den einzelnen Abstraktionsebenen bewusste Entwurfsentscheidungen, die sinnvollerweise dokumentiert werden.

Bisherige Erfahrungen haben gezeigt, dass das Denken in diesen drei Ebenen auch von Nicht-RE-Spezialisten gut angenommen wird. Durch die Trennung der verschiedenen Zielsetzungen, die den Ebenen zugrunde liegen werden die einzelnen Dokumente wesentlich besser verständlich. Durch den Einsatz von Use Cases wird zudem der Weg von Anforderungen hin zu aussagekräftigen Testfällen vereinfacht, da durch die Use Cases ein Kontext für einzelne Testschritte geschaffen wird.

Abbildung 2 stellt das Zusammenspiel der verschiedenen Abstraktionsebenen graphisch dar.

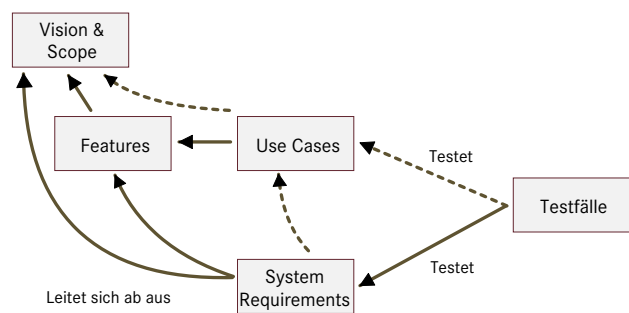


Abbildung 2: Zusammenspiel der Abstraktionsebenen

3 Resümee

Der vorliegende Kurzaufsatz hat versucht, ein kleines Schlaglicht auf Herausforderungen und mögliche Lösungsansätze im Bereich Requirements Engineering in der Automobilindustrie zu geben. Neben den hier angesprochenen Ansätzen und Erfahrungen zum Umgang mit wachsender Komplexität gibt es noch eine Vielzahl weiterer Tätigkeitsfelder. Exemplarisch seien hier genannt:

- Nutzeradäquate Beschreibung von Anforderungen (Was ist die „geeignete“ Beschreibungsform für Anforderungen?)
- Zusammenspiel von Requirements Engineering und Projektmanagement
- Geeignete werkzeugtechnische Unterstützungen (z.B. unter Verwendung von DOORS)
- Systematischer Umgang mit Änderungen

Bei allen Änderungen und Verbesserungen, die an dieser Stelle versucht werden, darf freilich nie der

Kosten-/Nutzenaspekt aus den Augen gelassen werden. So einleuchtend diese Forderung ist, so oft wird diese außer Acht gelassen. Vor allem der Nutzen kann oft nicht wirklich vorhergesagt werden. Bisweilen ergibt sich der Nutzen, der dazu führt, dass ein Ansatz die Pilotphase überdauert, an Stellen, die anders eingeschätzt wurden. Ein kurzes Beispiel soll dies illustrieren. In einem Pilotprojekt wurde DOORS als Requirements Management Werkzeug eingeführt mit dem Ziel, textuelle Anforderungen mit existierenden Modellierungen zu verknüpfen, um so die Konsistenz zwischen diesen beiden Beschreibungsformen sicher zu stellen. Im Verlauf des Projekts haben sich jedoch andere Aspekte des Werkzeugeinsatzes als deutlich nutzbringender herausgestellt: Der Einsatz eines zentralen Repositories, das mit DOORS einhergeht, hat das Problem unterschiedlicher Versionsstände beseitigt; die automatische History-Funktion macht nun Änderungen deutlich transparenter und die Annotation von Status-Attributen hilft deutlich bei der Projektverfolgung.

Literatur

- [1] S. Lauesen. *Software Requirements — Styles and Techniques*. Addison-Wesley, London, 2002.
- [2] S. Robertson und J. Robertson. *Mastering the Requirements Process*. Addison-Wesley, Harlow, 1999.