Embedded Quality

Qualitätssicherung eingebetteter Software:

Methoden und Best-Practices

FUSIM

Prof. Dr.-Ing. K. Bender, Dipl.-Ing. P. Jack, Dipl.-Ing. A. Koç, Dipl.-Ing. I. Péter, Dipl.-Ing. G. Megyeri

Informationstechnik im Maschinenwesen, TU München

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Ein Titeldatensatz für diese Publikation ist bei
Der Deutschen Bibliothek erhältlich

ISBN 3-8316-0024-4

Printed in Germany

Herbert Utz Verlag GmbH, München
Tel.: 089/277791-00 - Fax: 089/277791-01

Vorwort

Der enorme und stetig wachsende Preis-Leistungsvorteil von mikroelektronisch statt konventionell elektromechanisch gesteuerter Produkte führt zur einer permanenten Verlagerung der Produktfunktionalität in Richtung Software, die sich bereits heute im Maschinen- und Anlagenbau, der Automatisierungs- und Produktionstechnik und der Verkehrstechnik zu einer Schlüsseltechnologie entwickelt hat. Bei solchen Produkten macht die eingebettete Software einen wesentlichen Wertschöpfungsanteil aus und bestimmt zunehmend den Kundennutzen.

Nicht nur die Funktionalität, auch die Kosten solcher Produkte haben sich von der Mechanik und Elektrik stark in Richtung Mikroelektronik und Software verlagert. Beispielsweise liegt der Anteil der Elektronik in der Automobilindustrie zwischen 25–30%. Eine noch stärkere Verlagerung ist im Bereich der Kleinserie und Unikate zu erkennen, wo die Software-Entwicklungskosten dominieren. Von den Entwicklungsaufwendungen für neue Produkte im Bereich des Maschinenbaus fallen bis zu 50% auf die darin enthaltene Software. In einigen technischen Produkten des Maschinen- und Anlagenbaus erreicht sie sogar Anteile von 75–80% der Herstellungskosten. Software ist nicht nur zu einem Innovationstreiber, sondern auch zu einem entscheidenden Wirtschaftsfaktor geworden.

Dieser technologische Druck zwingt viele Unternehmen, ihre Produktentwicklung radikal zu ändern, denn die zunehmende Funktionalität intelligenter Produkte, in denen Software eingebettet ist, führt zu steigender Komplexität und hoher Flexibilität, nicht nur der Produkte selbst, sondern auch der Entwicklungsprozesse. Durch die erweiterten Freiheitsgrade solcher eingebetteter Systeme ist es möglich, erheblich schneller und kostengünstiger auf sich unerwartet ändernde Anforderungen des Marktes reagieren zu können. Die Entwicklungsprozesse werden somit zunehmend zu einem wettbewerbsbestimmenden Faktor. Vor dem Hintergrund, dass bereits heute Software eine Kernkomponente von innovativen Produkten darstellt, hat deren Qualität einen entscheidenden Einfluss auf die Qualität des Gesamtprodukts. Für deutsche KMU wird damit die Beherrschung qualitätssichernder Entwicklungsprozesse eingebetteter Software lebenserhaltend zur Sicherung eines beständigen wirtschaftlichen Erfolgs.

Das *vorliegende Dokument* entstand im Rahmen des DFAM-Projekts *FUSIM*[1]. Es gibt einen Überblick über die Qualitätssicherung (QS) von eingebetteter Software. Dabei werden neben einem qualitätsorientierten System-Vorgehensmodell und dem Ablauf des Testprozesses, der Testfall-Entwurf, Testwerkzeuge, Emulatoren sowie Qualitätssicherungsmaßnahmen beschrieben. Im Anhang finden sich QS-Dokumente der Industriepartner, Vorlagen und Checklisten zur Qualitätssicherung. Grundlage für die vorliegende Systematisierung der Qualitätssicherung von eingebetteter Software stellten die Bestandsaufnahme bei den Unternehmen und Literaturquellen dar.

Prof. Bender

---

[1] gefördert vom AIF (AIF-Nr. 11801, DFAM-Nr. 061170)

Inhaltsverzeichnis

1 Einleitung

Besondere *Merkmale intelligenter Produkte der Automatisierungstechnik* ("Eingebettete Systeme") sind die Kombination verschiedener technischer Disziplinen (Mechanik, Pneumatik, Hydraulik, Elektrik, Elektronik und Informationstechnik), das Echtzeitverhalten, die enge Kopplung zum technischen Prozess sowie die Vielzahl unterschiedlicher Kommunikations- u. Prozessschnittstellen.

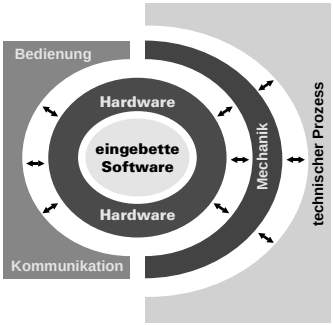

Abbildung 1 : Intelligentes Produkt (Eingebettetes System)

In diesen Produkten erfolgt eine permanente Verlagerung der Produktfunktionalität und Entwicklungskosten in Richtung Software. Diese sogenannte *eingebettete Software* macht einen wesentlichen Wertschöpfungsanteil aus und bestimmt den Kundennutzen. Als Kernkomponente eingebetteter Systeme hat sie einen entscheidenden Einfluss auf die Qualität des Gesamtprodukts [Halang94, Jansen95, Krämer94]. Für deutsche KMU wird damit die Beherrschung qualitätssichernder Entwicklungsprozesse für die eingebettete Software lebenserhaltend zur Sicherung eines beständigen wirtschaftlichen Erfolgs, denn die zunehmende Funktionalität intelligenter Produkte führt zu steigender Komplexität und hoher Flexibilität der Produkte sowie der Entwicklungsprozesse [Moitra99]. Schon heute entfallen zwei Drittel der Entwicklungskosten auf qualitätssichernde Maßnahmen [Myers99][Daich94].

1.1 Problemstellung

Obwohl die meisten Unternehmen ein definiertes bzw. standardisiertes *Vorgehensmodell* (Qualitätsmanagementsystem) für die Entwicklung technischer Systeme besitzen, geben die dort beschriebenen Entwicklungsprozesse vornehmlich die Mechanik- bzw. Maschinenbauersicht auf das Produkt wieder. Das Softwareentwicklungs-Vorgehen wird meist außer Acht gelassen bzw. nur sehr grob beschrieben. Die Folge ist, dass es keine systematische Planung und Durchführung von Qualitätsmaßnahmen über alle Entwicklungsphasen hinweg gibt.

Der Entwicklungsprozess eines *technischen Systems* durchläuft aus dem Blickwinkel der Steuerung typischerweise die Phasen Konstruktion, Steuerungsprojektierung (Elektrokonstruktion), Fertigung & Montage und Steuerungstest & Inbetriebnahme (Abbildung 2). Die zeitliche Abfolge der Phasen erfolgt streng sequentiell [Storr94].

Abbildung 2: Entwicklungsvorgehen technischer Systeme

Wenn auch die Methoden der Softwareentwicklung heute schon weit fortgeschritten sind, so berücksichtigen sie im Allgemeinen nur selten die in technischen Systemen vorhandene Informationstechnik und noch weniger bzw. gar nicht die mechanischen Komponenten oder gar den technischen Prozess, welchen man mit dem System realisieren möchte.

Dem gegenüber lassen sich etablierte Vorgehensweisen reiner Software-Systeme jedoch nicht einfach auf die eingebettete Software übertragen. Das Vorhandensein von drei Komponenten (Software, Hardware und Gerätemechanik) erfordert ein *interdisziplinäres* Entwicklungsteam und das *explorative Entwickeln* der Software. Dabei wird die eingebettete Software als der flexibelste Part verstanden, der sich den Gegebenheiten im Gesamtprojekt uneingeschränkt anpassen muss. Weiterhin existieren Unterschiede beim Test und Inbetriebnahme technischer Systeme. Bei der eingebetteten Software liegen viele Fehler im Zeitverhalten und im Zusammenspiel zwischen Software, Hardware und Mechanik. Das Echtzeitverhalten, die Nebenläufigkeit und die enge Kopplung mit dem technischen Prozess erschweren die Messbarkeit und Beobachtbarkeit. Ohne das fertige Automatisierungssystem ist die eingebettete Software nur eingeschränkt testbar.

Zudem kommt, dass das Wissen und die Fertigkeiten für den konstruktiven Anteil der Softwareentwicklung eines Unternehmens wesentlich ausgeprägter und verbreiteter sind, als zum Beispiel für den Test und die Inbetriebnahme. Erschwerend kommt hinzu, dass zum einen speziell für eingebettete Systeme Lösungen fehlen, wo geeignete Methoden und Werkzeuge bewertet werden. Da die Spannweite der Projekt- und Produktanforderungen sehr groß ist, macht die Empfehlung eines allgemeinen Vorgehensmodells wenig Sinn. Jedoch fehlt den KMU oft die Mittel und Ressourcen, um selber die Anpassung an ihre besonderen Bedürfnisse vorzunehmen.

Dem gegenüber steigt mit der ständigen Funktionsverlagerung in Richtung Software auch deren Komplexität. Dadurch nimmt die Qualität der Software eine zentrale Stellung in der Produktentwicklung ein. Um die sich daraus ergebenden Herausforderungen zu bewältigen und die Risiken zu mindern, bedarf es eines Vorgehensmodells, dass den Entwicklungsprozess aus Softwaresicht beschreibt und hilft, qualitätssichernd vorzugehen.

1.2 Zielsetzung

Das Handbuch gibt einen Überblick über die Qualitätssicherung von Software. Dabei wird aus der Literatur und den „Best-Practices“ der Unternehmen ein Vorgehensmodell vorgestellt, in dem zu jeder QS-Phase die möglichen Methoden und Werkzeuge zugeordnet werden. Insbesondere ist ein systematischer Testprozess in seinen Einzelteilen dokumentiert. Hier ist auch eine übersichtliche Bewertung der Methoden und Werkzeuge wiederzufinden.

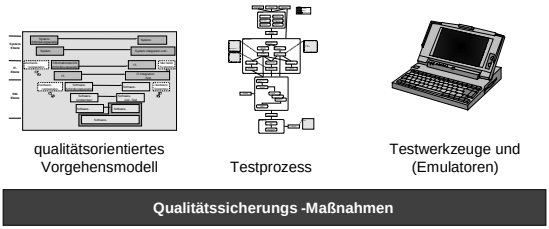

Abbildung 3: Zielsetzung

Das *Vorgehensmodell*, welches die qualitätsdominante Stellung der Software berücksichtigt, hilft in einem interdisziplinären Team die eigenen Entwicklungsprozesse zu ordnen und zu strukturieren. Klassische Verfahren des Testfallentwurfs erfüllen die Anforderungen der Praxis hinsichtlich der Effizienz, der graphischen Darstellung und Handhabbarkeit bei umfangreichen Testproblemen nicht zufriedenstellend. Das hier vorgestellte Konzept für den *Testfall-Entwurf* basiert auf der Klassifikationsbaum-Methode zur Testdaten-Ermittlung und einer erweiterten Beschreibung von Messsage Sequence Charts (MSC) für die Testablauf-Beschreibung. *Testwerkzeuge und QS-Maßnahmen*, klassifiziert und bewertet stellen ebenfalls eine wichtige Hilfe, bei der Auswahl für das eigene Unternehmen. Die zahlreichen *Dokumente und Vorlagen zur Qualitätssicherung*, die sich bei den Projektpartnern bewährt haben, können aufwandsarm in die eigenen Handbücher übernommen werden.

1.3 Aufbau des Handbuchs

Im Kapitel 2 wird ein allgemeines System-Vorgehensmodell beschrieben, in der die Qualitätssicherungsaspekte von Software im Vordergrund stehen. Da die Verbesserung des Testmethodik und -systematik im Mittelpunkt des Projektes FUSIM steht, erfolgt eine Strukturierung und detaillierte Beschreibung eines Testprozesses in Kapitel 3. Aufbauend dazu wird im Kapitel 1 auf Testwerkzeuge und In-Circuit-Emulatoren eingegangen. Neben der Klassifikation und Angabe über Verfahren zur Auswahl und Bewertung ist hier auch eine Übersicht über gängige Testwerkzeuge und Emulatoren zu finden. Kapitel 1 beschreibt den Testfall-Entwurf mit Hilfe der Kombination aus der Klassifikationsbaummethode und erweiterten Message-Sequence-Charts. Die Frage, welche Qualitätssicherungs-Maßnahme in welcher Entwicklungsphase erfolgen kann, wird in Kapitel 6 behandelt. Hier werden QS-Maßnahmen kurz beschrieben – ausführliche Beschreibung und Bewertung (Anhang B) – und in das Vorgehensmodell eingeordnet. Abgeschlossen wird mit den Ergebnissen der Bestandsaufnahme bei den FUSIM-Projektpartnern (Kapitel 7).

In den Anhängen sind Beschreibungen und Vorlagen wiederzufinden: Anhang A enthält neben einer Kurzbeschreibung der Werkzeuge auch die Kontaktadresse, um weitergehendere Informationen zu erhalten. Die Qualitätssicherungsmaßnahmen sind im Anhang B nach den Gesichtspunkten Einordnung, Ziel, Prüfobjekt und benötigte Unterlagen, Vorgehensweisen, Bewertung und weitergehende Informationen beschrieben. Anhang C wiederum beinhaltet Dokumente und Vorlagen der Projektpartner, die verallgemeinert und weitgehend anonymisiert wurden. Weitergehende Dokumente, wie QS-Dokumente des V-Modells oder der IEEE, Implementierungsregeln und Checklisten befinden sich im Anhang D.