

Technische Dokumentation von Soft- und Hardware in Eingebetteten Systemen

Technical Documentation of Soft- and Hardware in Embedded Systems

Beate Muranko, Rolf Drechsler, Universität Bremen

Zusammenfassung Eingebettete Systeme sind dadurch charakterisiert, dass sie Soft- und Hardwarekomponenten beinhalten. Diese Systeme sind integriert in Produkten wie Handys, Kameras, Fernseher, Autos usw. Aufgrund der Größe von Eingebetteten Systemen und der Wiederverwendung der einzelnen Komponenten erlangt die Dokumentation der Komponenten eine immer größere Bedeutung. Obwohl die Bedeutung der Dokumentation über Jahre gewachsen ist, ist es immer noch ein vernachlässigter Bereich im Entwicklungsprozess. In dieser Arbeit wird die Integration der technischen Dokumentation in Soft- und Hardwareentwicklungsprozessen diskutiert. Hierfür werden klassische Soft- und Hardwareentwicklungsmodelle bezüglich der technischen Dokumentation analysiert und evaluiert. Ferner wird ein aus der Praxis abgeleiteter Dokumentationsablauf vorgestellt, welcher in vorhandene Soft- und Hardwareentwicklungsmodelle integriert werden kann. Diese Arbeit zeigt einen Ansatz zur Integration des Dokumentationsablaufs in ein Entwicklungsmodell.

Summary Embedded Systems are characterized by the presence of software and hardware components. They are integrated into products such as mobile phones, cameras, TV sets, cars, etc. Due to the size of embedded systems and the reuse of components documentation of them becomes more important. Although the importance of documentation has increased over the years, it is still a largely neglected part of the development process. In this paper we discuss the integration of the technical documentation in the software and hardware development processes. Therefore, we analyze and evaluate classical software and hardware development models with regard to technical documentation. Furthermore, we present a workflow for documentation which is derived from practical experience and can be integrated in existing software and hardware development models. As a proof of concept we present an approach for the integration of documentation techniques into a development workflow.

KEYWORDS D.2.7. [Distribution, Maintenance and Enhancement] documentation, D.2.9. [Management] software process models, software quality assurance

1 Einleitung

Eingebettete Systeme sind in Produkten wie Handys, Kameras, Fernseher, Autos, usw. integriert. Die Untersuchung solcher Systeme kann anhand unterschiedlicher Aspekte durchgeführt werden z. B. anhand der Prozessorenanzahl. In Anlehnung an [15] werden über 79%

aller Prozessoren in Eingebetteten Systemen verwendet. Zudem verdoppelt sich für einige Produkte im Bereich der Endverbraucherelektronik die Anzahl des Programmcodes alle zwei Jahre. In der Regel beinhalten Eingebettete Systeme Soft- und Hardwarekomponenten. Einige von diesen Soft- und Hardwarekom-

ponenten können wiederverwendet werden, sodass keine Neuentwicklung notwendig ist.

Aufgrund der immensen Größe von Eingebetteten Systemen und der Wiederverwendbarkeit von Komponenten ist die Dokumentation von Soft- und Hardware von großer Bedeutung, um somit das Ver-

ständnis und die Verwendung der Komponenten zu erleichtern. Es ist zu beobachten, dass die technische Dokumentation einen ungenügenden Stellenwert und daraus folgend eine geringe wissenschaftliche Auseinandersetzung aufweist. Nach [20] wird das Schreiben der Dokumentation als ein ereignisloser Prozess empfunden, bei dem keine Erfolgserlebnisse in Form einer technischen Errungenschaft – z. B. das erste fehlerfreie Starten des Programms – erlangt werden. [13] beschreibt die Tätigkeit der Entwicklung der Soft- bzw. Hardware als einen Sachverhalt, der höher eingeschätzt wird als die technische Dokumentation des entwickelten Systems, obwohl die Akzeptanz der technischen Dokumentation in den letzten Jahren zugenommen hat¹. Ferner beklagen sowohl [13] als auch [20] das vorhandene Ausbildungsdefizit der Soft- bzw. Hardwareentwickler, da die Lehre der technischen Dokumentation in der Ausbildung nur wenig Berücksichtigung findet. Insgesamt sind ähnliche Defizite in der Forschung vorhanden. So ist z. B. in [11] die Dokumentation nur kurz erwähnt, obwohl in Bereichen wie *Design Reuse* der Dokumentation offensichtlich eine wichtige Rolle zukommen müsste.

Zum anderen ist die Durchführung von Soft- und Hardwareprojekten dadurch gekennzeichnet, dass der größte Teil der Dokumentation nicht mit dem letzten Stand der Soft- bzw. Hardware übereinstimmt, unvollständig und kompliziert zu lesen und zu verstehen ist. Da aber gegenwärtig und zukünftig die technische Dokumentation einen wesentlichen Faktor für das Gesamtprodukt darstellt, trägt sie neben dem eigentlichen Entwurfsprozess bei Soft- bzw. Hardwaressystemen eine immer größere Bedeutung. Schon heute werden unzureichende Dokumentationen als Quellen für Entwurfsfehler ausgemacht, siehe z. B. [3].

Ein weiteres Untersuchungsergebnis war die Feststellung, dass im Bereich der Dokumentation wenig Forschung betrieben wird.

Ein Bereich, in dem die technische Dokumentation behandelt wird, ist die automatische Generierung von Source Code Dokumentation. Ein Beispiel hierfür ist zum einen das „Javadoc“ Tool² von SUN Microsystems, welches aus den Kommentaren im Source Code eine API Dokumentation in HTML erstellt. Weiterhin existiert das „CppDoc“ Tool³, welches ebenfalls eine HTML Dokumentation erstellt. Diese Dokumentation wird durch das Extrahieren der Kommentare aus den C++ Klassen generiert. Die Ausgaben von CppDoc und Javadoc sind ähnlich. Diese Tools sind für die Erstellung einer ersten Source Code Dokumentation hilfreich, aber sie liefern keinen Weg zur Erzeugung einer kompletten Dokumentation.

Zusätzlich zu den Tools existieren Initiativen, die sich mit der Thematik der Dokumentation befassen. Ein Beispiel hierfür ist die SPIRIT Consortium Initiative⁴. Die SPIRIT Consortium Spezifikation ist auf die Verbesserung und Festlegung der automatischen Integration von IP-Komponenten in Designabläufen fokussiert. Die automatische Integration von IP-Komponenten soll durch die Entwicklung von Spezifikationen in zwei Bereichen erzeugt werden. Zum einen soll eine Spezifikation für die IP Meta Beschreibung und zum anderen eine Spezifikation für das Interface der Integration von IP Generatoren und spezialisierten Werkzeugen (*point-tools*) entwickelt werden [14]. Da der Fokus auf der IP Integration liegt, deckt er einige Dokumentationsaspekte ab, aber er behandelt nicht den Dokumentationsablauf.

Unser Forschungsansatz betrachtet das Problem von einer anderen Perspektive. Wir unter-

suchen neue Aspekte der Dokumentation. Hierbei betrachten wir die Erstellung von Handbüchern und anderen wichtigen Dokumenten, die für ein Gesamtprodukt von Bedeutung sind. In der vorliegenden Arbeit behandeln wir die Berücksichtigung und Integration der technischen Dokumentation in Soft- und Hardwareentwicklungsprozessen. Um den aktuellen wissenschaftlichen Stand der technischen Dokumentation aufzuzeigen, wird eine nähere Betrachtung der klassischen Entwicklungsmodelle durchgeführt. Durch eine Analyse der Modelle und eine anschließende Auswertung wird die Integration der technischen Dokumentation veranschaulicht. Im Anschluss wird ein praxiserprobter Dokumentationsablauf erläutert, der auch für Soft- und Hardware verwandt werden kann. Basierend auf dem Dokumentationsablauf zeigen wir eine Integration der Dokumentation in ein Softwareentwicklungsmodell.

Diese Arbeit gliedert sich wie folgt: Abschnitt 2 gibt eine kurze Einführung in den theoretischen Hintergrund der technischen Dokumentation. Im darauffolgenden Abschnitt 3 werden Soft- und Hardwareentwicklungsmodelle aufgezeigt. Danach erfolgt die Analyse der einzelnen Modelle und die Bewertung der vorhandenen Integration der technischen Dokumentation. Abschnitt 4 stellt einen generellen Dokumentationsablauf skizzenhaft dar, der einer technischen Dokumentation zugrunde liegt. Im darauffolgenden Abschnitt 5 präsentieren wir eine Annäherung für eine allgemeine Lösung der Integration des Dokumentationsablaufs in ein Softwareentwicklungsmodell. Abschließend behandelt Abschnitt 6 die wesentlichen Ergebnisse dieser Arbeit und liefert einen Ausblick auf mögliche Erweiterungen.

2 Hintergrund

Dieser Abschnitt liefert den theoretischen Rahmen für das Verständnis der Arbeit. Es wird eine Darstel-

¹ Siehe <http://www.tekom.de>

² <http://java.sun.com/j2se/javadoc/>

³ <http://www.cppdoc.com/>

⁴ <http://www.spiritconsortium.com/>

lung über die begrifflichen Grundlagen „technische Dokumentation“ und „interne/externe Dokumentation“ gegeben. Ferner wird eine Auflistung der möglichen Dokumentationsbereiche aufgezeigt. Abschließend wird kurz auf die Internationale Norm DIN EN 62079 eingegangen, die bisher die allgemeine Grundlage für die Erstellung einer technischen Dokumentation liefert. Der Vollständigkeit wegen wird an dieser Stelle das Geräte- und Produktsicherheitsgesetz GPSG erwähnt, welches sich mit Sicherheitsanforderungen an Verbrauchsprodukten und technischen Arbeitsmitteln befasst⁵. Auf dieses Gesetz wird hier nicht näher eingegangen.

2.1 Technische Dokumentation

Nach [10] umfasst die technische Dokumentation alle notwendigen und zweckgerichteten Produktinformationen und ihre Benutzung, die auf elektronischen Medien oder auf Papier festgehalten werden. Ferner spezifiziert [8] die Speicherung derjenigen produktbezogenen Daten und Informationen, die von Beginn der Planung eines Produktes über dessen gesamten Lebensweg entwickelt und verwendet werden.

2.2 Interne/Externe Dokumentation

In Anlehnung an [9; 17] sind unter dem Begriff der internen Dokumentation alle Anweisungen und Informationen zu verstehen, die ausschließlich für die eigenen Mitarbeiter und den internen Gebrauch erzeugt werden. Beispiele hierfür sind Pflichtenhefte, Konstruktions- und Fertigungsunterlagen als auch Qualitätssicherungsdokumente. Unter dem Begriff der externen Dokumentation sind nach [9; 17] alle produktbegleitenden Kundeninformationen und Instruktionen zu verstehen, die mit dem Produkt an den Kunden ausgeliefert werden, die so genannten Benutzerinformationen. Beispiele hierfür sind

⁵ <http://www.hvbg.de/d/bgp/prod/gs/gpsg/index.html>

Betriebsanleitungen, Gebrauchsanweisungen und Hinweise.

2.3 Dokumentationsbereiche

Nach [18] können die Gesamtdokumentationsbereiche in vier Gebiete unterteilt werden: Projekt-, Entwicklungs-, Produkt- und Benutzerdokumentation. Zum Bereich der technischen Dokumentation gehören die Entwicklungs-, die Produkt- und die Benutzerdokumentation, über deren Inhalte im Folgenden eine kurze Übersicht gegeben wird. Die Projektdokumentation wird nicht weiter betrachtet, da sie kein Teil der technischen Dokumentation ist.

Entwicklungsdokumentation Im Bereich der Entwicklungsdokumentation werden die Aufgabenbeschreibungen, die Beschreibung der Lösungswege bzw. Alternativen und die Darstellung der Hilfsmittel, die zur Lösung herangezogen werden, festgehalten.

Produktdokumentation Die Produktdokumentation beinhaltet beispielsweise Informationen zur Übersichtsbeschreibung und zur Programmbeschreibung, worunter die Darstellung des Programmablaufs und des zu Grunde liegenden Datenmodells zu verstehen sind.

Benutzerdokumentation Der Bereich der Benutzerdokumentation behandelt Anleitungen für den Betrieb des Produktes – z. B. das Benutzerhandbuch, Unterlagen für die Wartung wie Instandhaltungsvorschriften, Schulungsunterlagen und Vertriebsunterlagen.

2.4 DIN EN 62079

Aus Gründen der Haftung werden Hersteller und Importeure von technischen Systemen durch Gesetze (z. B. Produkthaftungsgesetz) und Vorschriften (VDE Vorschriften, Normen) zur Bereitstellung der Dokumentation gezwungen. Die DIN EN 62079 beinhaltet für den Entwurf und die Erstellung von technischen Dokumentationen eine

allgemeine Grundlage. Der Anwendungsbereich der DIN EN 62079 „[...] umfasst Anleitungen für kleine und einfache Produkte wie z. B. eine Dose Farbe bis hin zu großen und hoch komplexen Produkten wie z. B. große Industrieanlagen“ [7], S. 8. Nach [7] kann keine allgemeingültige Norm jeden einzelnen Spezialfall abdecken. Aus diesem Grund müssen für die Erstellung einer technischen Dokumentation die spezifischen Produktnormen zusätzlich herangezogen werden. Eine Einschränkung auf einen bestimmten Produktbereich – wie z. B. Software- bzw. Hardwareentwicklung – ist nicht gegeben, sodass keine spezifische Grundlage für die Erstellung einer technischen Dokumentation in diesem Entwicklungsbereich existiert. Abschließend ist erwähnenswert, dass die DIN EN 62079 auch eine Checkliste zur technischen Überprüfung von Anleitungen enthält. Anhand dieser Checkliste kann eine Bewertung von Anleitungen vorgenommen werden. Aber auch diese lässt Details bezüglich Software- und Hardware vermissen.

3 Soft- und Hardwareentwicklungsmodelle

Die ungenügende Behandlung der technischen Dokumentation in Software- und Hardwareentwicklungsmodellen bilden den Hintergrund für die in diesem Abschnitt dargestellten klassischen Software- und Hardwareentwicklungsmodelle. Hierbei wird das Ziel verfolgt, die (mangelnde) Berücksichtigung und Integration der technischen Dokumentation in den Modellen aufzuzeigen. Es werden diejenigen Vorgehensmodelle zur Entwicklung von Software vorgestellt, die eine Dokumenterstellung als integralen Bestandteil beinhalten. Dies bedeutet, dass die Entwicklung und Pflege von Dokumenten bei der Entwicklung von Software im Vordergrund steht. Hinsichtlich der Hardwareentwicklungsmodelle wird das allgemein anerkannte Entwicklungsmodell heran gezogen, das in [5] beschrieben ist. Im Folgenden

wird das daran angelehnte Vorgehen aus [6] präsentiert, welches den klassischen Entwicklungsablauf in Phasen darstellt. Die Soft- und Hardwareentwicklungsmodelle werden skizzenhaft veranschaulicht und abschließend hinsichtlich der Integration von technischer Dokumentation bewertet.

3.1 Softwareentwicklungsmodelle

In Anlehnung an [2] gehören zu den dokumentenorientierten Softwareentwicklungsmodellen zum einen das Wasserfallmodell [4] und zum anderen das V-Modell [2]. Im Folgenden werden beide Modelle kurz beschrieben. Das Wasserfallmodell [4] beschreibt die Herstellung von Software als sequenzielle und stufenweise Reihenfolge der Entwicklungsphasen, siehe Bild 1. Nach [2; 20] ist das primäre Ziel dieses Modells, einen minimalen Managementaufwand für die Entwicklung eines Produktes aufzubringen. Bei der Entwicklung muss jede Phase komplett abgeschlossen sein, bevor die nächste Phase durchlaufen werden kann. Da das Modell ein dokumentengetriebenes Modell ist, sind die Resultate jeder Phase Dokumente, die in die darauf folgende Phase mit eingehen. Charakteristisch für das Wasserfallmodell ist, dass eine Rückkopplung zwischen den Phasen nur bei aufeinander folgenden Phasen möglich ist.

Nach [2; 12] ist das V-Modell [2] eine Erweiterung des Wasserfall-

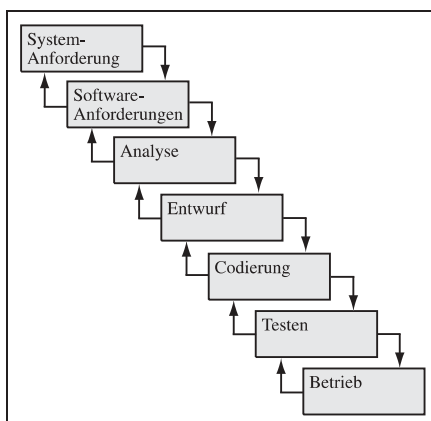


Bild 1 Wasserfallmodell.

modells, siehe Bild 2. Primäres Ziel ist es, eine Qualitätssicherung in die Entwicklung zu integrieren. Um dies zu erreichen, wurde das Wasserfallmodell um die Aktivitäten Verifikation und Validation erweitert. Hierbei wird unter Verifikation die Überprüfung der Korrektheit des Systems, d.h. die Übereinstimmung des Systems mit der Spezifikation, verstanden und durch die Validation wird die Angemessenheit des Systems an die Problemstellung gewährleistet. Durch diese beiden Erweiterungen existiert keine strikte zeitliche Abfolge zwischen den einzelnen Phasen. Als Ergebnis einer Phase liegen in diesem Modell ebenfalls Dokumente vor.

3.2 Hardwareentwicklungsmodell

Nach [5; 6] wird die Entwicklung von Hardware anhand des klassischen Vorgehensmodells durchgeführt. Im Folgenden wird dieses Modell skizzenhaft dargestellt, siehe Bild 3. Der Entwicklungsprozess der Hardware startet mit der textuellen Spezifikation, die im weiteren Schritt formalisiert wird. Diese Formalisierung bildet die Grundlage für die Simulation, die mittels einer (Hardware-) Beschreibungssprache erreicht wird. Der Programmcode wird dann automatisch in eine Netzliste übersetzt. Diese Netzliste bildet den Ausgangspunkt für die Produktion des Chips.

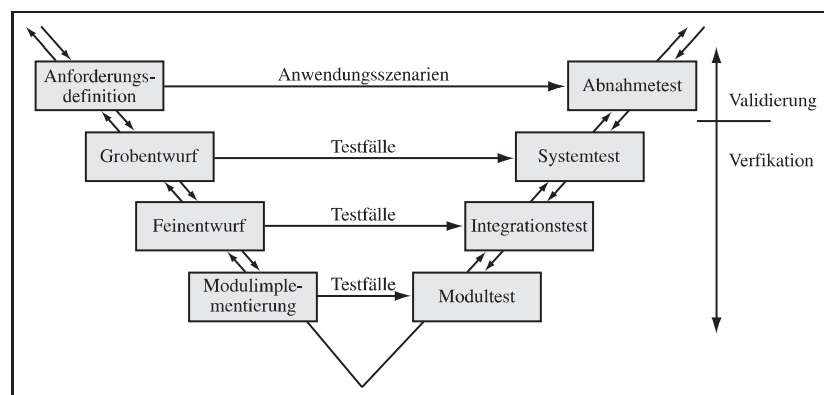


Bild 2 V-Modell.

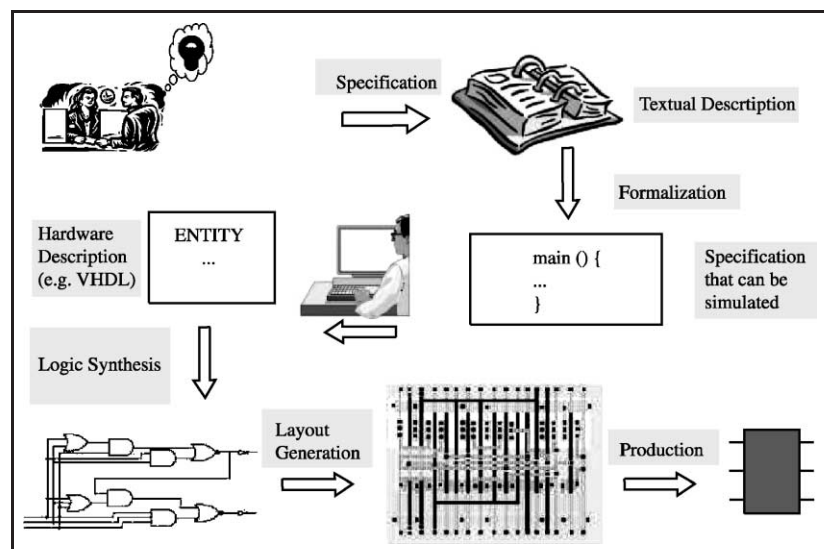


Bild 3 Hardware Modell.

3.3 Diskussion

Anhand dieser kurzen Darstellung der Merkmale der Entwicklungsmodelle [2; 4; 6] wird schnell deutlich, dass zwar die einzelnen Phasen und deren Zusammenhang klar ersichtlich sind, jedoch im Modell die technische Dokumentation vernachlässigt wird. Betrachtet man die Softwareentwicklungsmodelle [2; 4], so wird dadurch offensichtlich, dass in beiden Modellen die Dokumentation in keiner Phase explizit veranschaulicht wird. Somit lässt sich der Rückschluss ziehen, dass beide Softwareentwicklungsmodelle nicht angeben, an welchem Phasenabschnitt welche technische Dokumentation vorgenommen werden muss und wie diese Dokumentation auszusehen hat. Das klassische Hardwareentwicklungsmodell [6] verfährt noch drastischer mit der Notwendigkeit der technischen Dokumentation, da es in keinem Phasenabschnitt auf diese eingeht. Zusammenfassend wird somit ersichtlich, dass sowohl Soft- als auch Hardwareentwicklungsmodelle die technische Dokumentation nur ungenügend behandeln.

4 Dokumentationsablauf

Im Folgenden wird in Anlehnung an [19] der generelle Dokumentationsablauf, der auf der Entstehung einer technischen Dokumentation basiert, vorgestellt. Der Ablauf gliedert sich in vier Phasen, siehe Bild 4. Im weiteren Verlauf werden die Charakteristika der Phasen vorgestellt. Weiterführende Literatur und alternative Ansätze sind in [1; 9; 10; 16] zu finden.

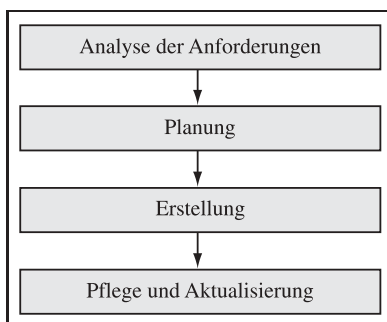


Bild 4 Dokumentationsablauf.

4.1 Analyse der Anforderungen

Als primäres Ziel gilt es, eine Analyse der Anforderungen an die technische Dokumentation durchzuführen. Danach sollte die Zielgruppe der Anleitung ermittelt werden, um somit den Detaillierungsgrad der technischen Dokumentation definieren zu können. Existiert in der Zielgruppe eine geringe Vorbildung bezüglich des Produktes, so muss eine detaillierte Dokumentation erstellt werden. Im Anschluss daran sollte sich der Autor der Dokumentation ausführlich mit dem zu dokumentierenden Produkt vertraut machen, um daraufhin die Erstfassung der Produktbeschreibung vornehmen zu können. Die Erstfassung sollte mit allen beteiligten Personen abgestimmt werden, um frühzeitig Missverständnisse zu erkennen und zu beseitigen. Als abschließender Punkt in dieser Phase ist es wichtig, eine Aufwandsabschätzung – z. B. Anzahl der Seiten oder Höhe der Kosten – durchzuführen.

4.2 Planung

In der Planungsphase findet eine Zuordnung der Zuständigkeiten an Personen statt. Ferner werden in dieser Phase das Layout, die interne Grobstruktur und der Detaillierungsgrad der Dokumentation festgelegt. Weiter ist zu prüfen, ob das Produkt den speziellen Normen genügt. An dieser Stelle muss eine Recherche durchgeführt werden und eventuell juristischer Beistand z. B. bei Absicherungen oder Garantieansprüchen in Anspruch genommen werden. Das Ergebnis dieser Phase ist eine Gesamtstruktur und ein zeitlicher Ablauf der Dokumentation.

4.3 Erstellung

Die Erstellungsphase besteht aus drei Dokumentationsfassungen: Alpha-, Beta- und Schluss-Fassung. Die jeweiligen Dokumentationsfassungen werden parallel zur Produkterstellung erzeugt. Die Alpha-Fassung beinhaltet im Wesentlichen bereits die gesamte Funktionalität des Produktes. Gemäß des Doku-

mentationsplanes werden zu jeder Komponente Beschreibungen erstellt. In dieser Phase sollten alle Werkzeuge, die bei der Erstellung des Dokumentes zum Einsatz kommen, getestet werden, um sich mit deren Funktionalität vertraut zu machen. Eine Verifikation der Alpha-Fassung wird durch die für die jeweiligen Projektteile zuständigen Mitarbeiter durchgeführt. Die hierbei entstehenden Überarbeitungsvorschläge und Verbesserungskommentare werden in die nächste Fassung, die Beta-Fassung, mit eingepflegt. Die Beta-Fassung wird wiederholt Entwicklern zur Durchsicht vorgelegt. Zusätzlich sollte eine Begutachtung von Experten und Endbenutzern durchgeführt werden. Die aus dieser Begutachtungsphase entstandenen Kommentare werden in die endgültige Fassung, die Schluss-Fassung, mit eingebaut.

4.4 Pflege und Aktualisierung

Erst beim Einsatz der Dokumentation stellen sich inhaltlich unzureichende oder fehlerhafte Stellen heraus, die dann in der Dokumentation überarbeitet werden müssen. Die Pflege und Aktualisierung der Dokumentation muss permanent durchgeführt werden.

5 Integrierter Ansatz

Im vorherigen Kapitel wurde gezeigt, dass das nicht Vorhandensein von Dokumentationsprozessen in Entwicklungsmodellen ein Problem ist, welches gelöst werden muss. Unser initialer Ansatz für eine Lösung basiert auf der Integration des aufgezeigten Dokumentationsablaufs in das Wasserfallmodell⁶. Hierbei wurde das Wasserfallmodell durch den Dokumentationsablauf erweitert. Im Bild 5 ist der durch die Integration entstandene initiale Ansatz veranschaulicht.

⁶ In diesem Beitrag haben wir die Integration der technischen Dokumentation in ein statisches Software Entwicklungsmodell betrachtet. Eine Alternative hierzu ist z. B. die Integration in die Model Driven Architecture.

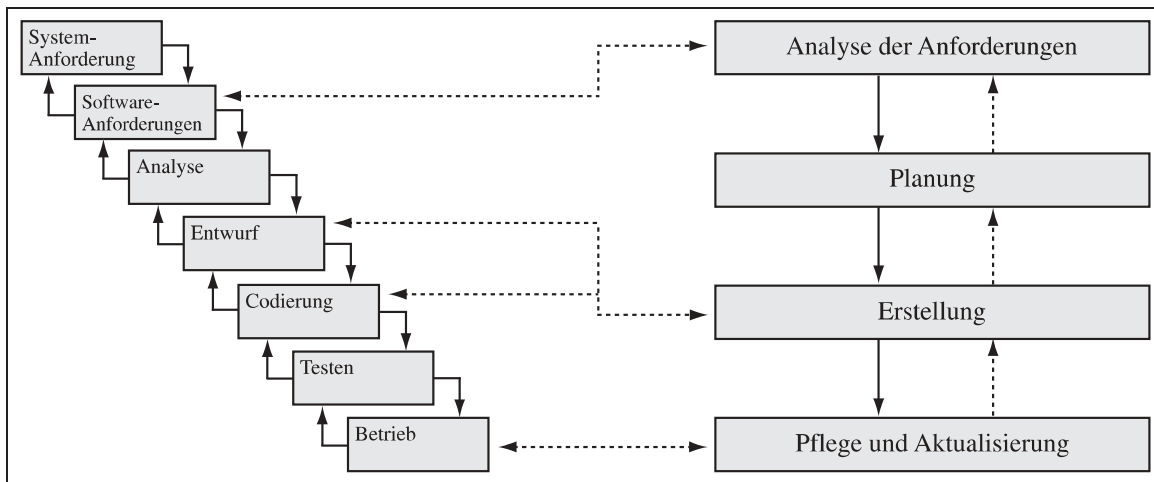


Bild 5 Initialer Ansatz.

Die rechte Seite des Ansatzes repräsentiert den Dokumentationsablauf, der durch Rückkopplungsmöglichkeiten zwischen den einzelnen Phasen erweitert worden ist. Diese Rückkopplungen gewährleisten Wechselwirkungen zwischen den Phasen. Sie sind durch Pfeile dargestellt, die jede Phase mit der vorherigen Phase verbinden. Ein Beispiel für eine Rückkopplung ist die Möglichkeit von der Phase der „Pflege und Aktualisierung“ zur Phase „Erstellung“ zurückzugelangen. Die linke Seite des Ansatzes repräsentiert das Wasserfallmodell.

Das neue Modell zeigt explizit auf, an welchen Stellen die Dokumentation in das Wasserfallmodell integriert sein muss. Die Integration der Dokumentation wird anhand von gestrichelten Pfeilen veranschaulicht, die den Dokumentationsablauf mit dem Wasserfallmodell verbinden. Im Folgenden werden die einzelnen Entwicklungsphasen des Wasserfallmodells, die die Dokumentation integrieren, veranschaulicht.

Softwareanforderungen. In der Phase der Erstellung der „Softwareanforderungen“ ist es neben der Anforderungsaufnahme für die Software ebenso wichtig die Anforderungen an die Dokumentation aufzunehmen. Hierbei muss ermittelt werden, welche Zielgruppe das Hand-

buch ansprechen soll. Dies ist wichtig, um den Detaillierungsgrad der Beschreibung in der Dokumentation einschätzen zu können. Ferner sollte an dieser Stelle herausgefunden werden, welches Format die Dokumentation haben soll – bspw. ein Buch, eine CD oder ein online-Format.

Entwurf und Codierung. In der Entwurfsphase wird der Softwareentwurf modelliert. Hierbei wird ein Überblick über die Architektur und die einzelnen Komponenten vermittelt. In der Codierungsphase wird das entwickelte Modell umgesetzt. An beiden Phasen wird die Dokumentation integriert. Die Integration beruht auf dem Hintergrund, dass die Erstellung der einzelnen Dokumentationsfassungen (Alpha-, Beta- und Schlussfassung) parallel zur Erstellung des Produktes verfasst werden müssen. Durch die Integration wird eine mit dem Produkt abgestimmte Erstellung der Dokumentation ermöglicht, d. h. Aktualisierungen im Entwurf werden direkt in die Dokumentation übernommen, sodass der aktuelle Stand auch in der Dokumentation wieder zu finden ist.

Betrieb. Erst im laufenden Betrieb stellen sich fehlerhafte Stellen heraus, die dann in der Entwicklung überarbeitet werden müssen. Findet eine Überarbeitung statt, so muss

die Dokumentation mit der Überarbeitung abgestimmt werden.

Um die Softwareentwicklung mit der dazugehörigen Dokumentation in einem einzigen Modell zu veranschaulichen, wurden die Phasen der Dokumentation in das Wasserfallmodell mit eingebaut. Somit erhält man einen Gesamtüberblick der Softwareentwicklungsphasen inklusive der Dokumentation, siehe Bild 6.

Anhand dieses zusammengefassten Modells ist schnell ersichtlich, dass nicht alle Phasen einer Softwareentwicklung durch die Dokumentation ergänzt werden. Um nun eine bessere Übersicht der dokumentationsrelevanten Phasen zu erhalten, wurden nur die betreffenden Phasen betrachtet. Diese Phasen wurden separat in einem weiteren Modell aufgezeigt und mit den Phasen der Dokumentation verbunden. Das so entstandene Modell ist im Bild 7 veranschaulicht.

6 Zusammenfassung und abschließende Bewertung

Ziel dieser Arbeit war es, die Berücksichtigung und Integration der technischen Dokumentation in Soft- und Hardwareentwicklungsprozessen, die für die Entwicklung von Eingebetteten System verwendet werden können, aufzuzeigen. Um sich diesem zentralen Sachverhalt zu nähern, wurde in den einzelnen Abschnitten das benötigte

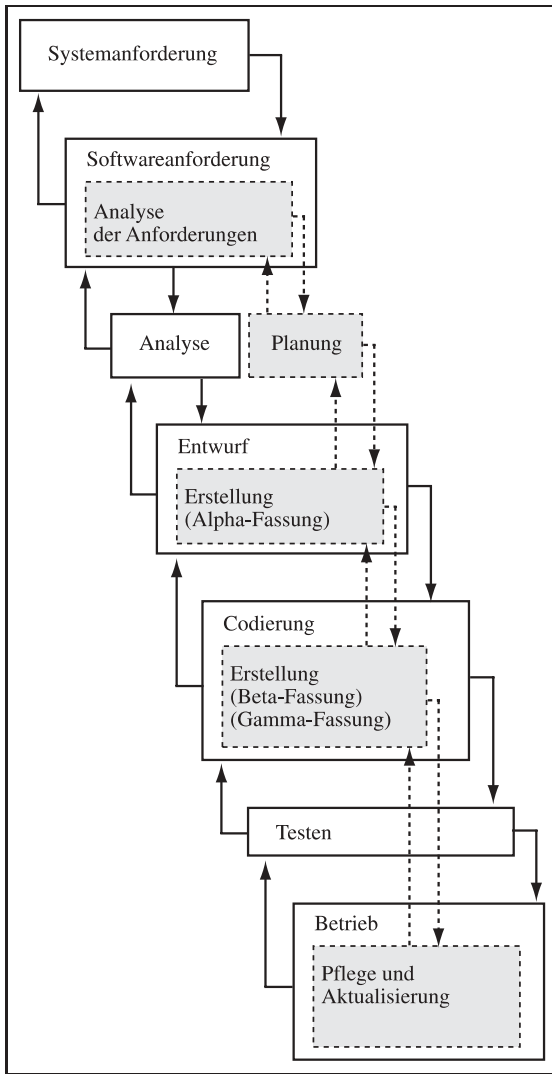


Bild 6 Zusammengeführtes Modell.

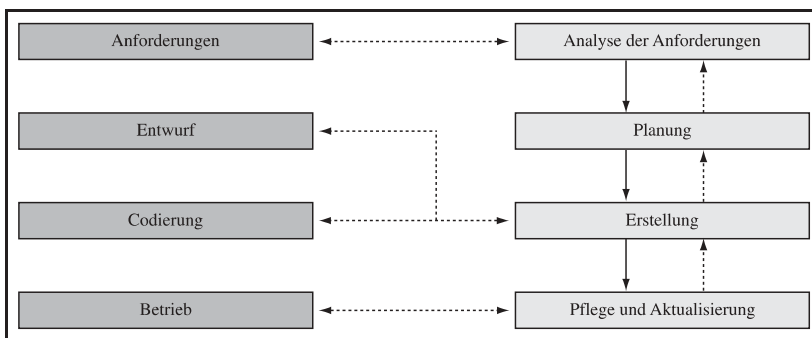


Bild 7 Allgemeiner Ansatz.

Hintergrundwissen behandelt. Die Herangehensweise erfolgte von der Betrachtung der einzelnen Dokumentationsbereiche über die DIN EN 62079 Norm bis hin zu einer Analyse der wissenschaftlichen Soft- und Hardwareentwicklungs-

modelle. Die daraus gewonnenen Erkenntnisse zeigen auf, dass die technische Dokumentation sowohl in der DIN EN 62079 Norm als auch in den Entwicklungsmodellen unzureichend und oberflächlich behandelt wird.

In diesem Aufsatz haben wir einen aus der Praxis abgeleiteten Dokumentationsablauf vorgestellt und diesen im Kontext des Wasserfallmodells weiter betrachtet. Eine Integration der beiden Modelle veranschaulicht unseren Lösungsansatz. Eine mögliche Vertiefung ist die Erweiterung unseres Ansatzes auf andere Entwicklungsmodelle, wie beispielsweise das V-Modell und das hierzu erweiterte V-Modell XT.

Literatur

- [1] T. T. Barker: Writing Software Documentation: A Task-Oriented Approach, Longman, 2003.
- [2] H. Balzert: Lehrbuch der Softwaretechnik: Software-Management Software-Qualitätssicherung Unternehmensmodellierung, Spektrum, 1998.
- [3] B. Bentley: Validating the Intel Pentium 4 Microprocessor. In: *Proc. DAC 01*, 2001, pp. 244–248.
- [4] B. W. Boehm: Software-Engineering Economics, Englewood Cliffs: Prentice Hall, 1981. In: H. Balzert: Lehrbuch der Softwaretechnik: Software-Management Software-Qualitätssicherung Unternehmensmodellierung, Spektrum, 1998.
- [5] G. De Micheli: Synthesis and optimization of digital circuits, McGraw-Hill, 1994.
- [6] R. Drechsler, A. Breiter: Hardware Project Management - What we Can Learn from the Software Development Process for Hardware Design?. In: *4th Conf. of Informatics and Information Technologies 03*, 2003.
- [7] DIN EN 62079: Erstellen von Anleitungen: Gliederung, Inhalt und Darstellung, VDE, 2000.
- [8] C. H. Gabriel: Europa-Projekt Didos, *Tekom-Nachrichten* 15.02.92. In: H. P. Krings: Wissenschaftliche Grundlagen der Technischen Kommunikation, Gunter Narr, 1996.
- [9] H. P. Hahn: Technische Dokumentation leichtgemacht, Hanser, 1996.
- [10] W. Hoffmann, B. G. Hölscher, U. Thiele: Handbuch für technische Autoren und Redakteure, VDE, 2002.
- [11] M. Keating, P. Bricaud: Reuse Methodology Manual: For System-on-a-Chip Design, Kluwer, 2002.

- [12] H. Krcmar: Informationsmanagement, Springer, 2003.
- [13] F. Lehner: Software-Dokumentation, Logos, 1999.
- [14] C. K. Lennard: Industrially Proving the SPIRIT Consortium Specifications for Design Chain Integration. In: DATE Design Automation and Test in Europe, 2006.
- [15] P. Marwedel: Embedded System Design, Kluwer, 2003.
- [16] J. Price, H. Korman: How to Communicate Technical Information: A Handbook of Software and Hardware Documentation, Addison-Wesley, 1993.
- [17] G. Pötter: Die Anleitung zur Anleitung: Leitfaden zur Erstellung technischer Dokumentationen, Vogel, 1994.
- [18] H. J. Scheibl: Wie dokumentiere ich ein DV-Projekt?: Dokumentationsverfahren in Theorie und Praxis, expert, 1985.
- [19] A. Sikora, R. Drechsler: Software-Engineering und Hardware-Design: Eine systematische Einführung, Kapitel 11, Hanser, 2002.

- [20] E. Wallmüller: Software-Qualitätsmanagement in der Praxis: Software-Qualität durch Führung und Verbesserung von Software-Prozessen, Hanser, 2001.



1



2

1 Dipl.-Inf. Beate Muranko studierte Informatik an der Universität Bremen. Seit September 2005 ist sie wissenschaftliche Mitarbeiterin an der Universität Bremen. Ihre Forschungsinteressen liegen insbesondere in der technischen Dokumentation und formalen Spezifikation von Schaltungen und Systemen.
 Adresse: Universität Bremen, Bibliothekstraße 1, 28359 Bremen, Germany, Tel.: +49-421-2184733,

Fax: +49-421-2187385, E-Mail: bmuranko@informatik.uni-bremen.de

2 Prof. Dr. Rolf Drechsler hat an der Johann-Wolfgang-Goethe-Universität in Frankfurt am Main Mathematik und Informatik studiert. Im Jahr 1995 schloss er sein Studium mit der Promotion ab. Von 1995 bis 2000 war er als Wissenschaftlicher Assistent an der Albert-Ludwigs-Universität in Freiburg beschäftigt, wo er 1999 habilitierte. Im Anschluss arbeitete er als Senior Engineer in der Abteilung Corporate Technologies der Siemens AG, München, bis er im Oktober 2001 die Professur für Rechnerarchitektur am Fachbereich 3, Mathematik und Informatik, der Universität Bremen annahm. Seine wissenschaftlichen Interessen umfassen den Schaltkreis- und Systementwurf mit dem Schwerpunkt Formale Verifikation.
 Adresse: Universität Bremen, Bibliothekstraße 1, 28359 Bremen, Germany, Tel.: +49-421-2187389, Fax: +49-421-2187385, E-Mail: drechsle@informatik.uni-bremen.de