

Luft- und Raumfahrt
Raumfahrttechnik
Software

Deutsche und Englische Fassung EN 14160:2001

DIN**EN 14160**

ICS 49.140

Aerospace — Space engineering — Software;
German and English version EN 14160:2001

Aéronautique et espace — Ingénierie spatiale — Logiciel;
Version allemande et anglaise EN 14160:2001

Die Europäische Norm EN 14160:2001 hat den Status einer Deutschen Norm.

Nationales Vorwort

Die *European Cooperation for Space Standardization (ECSS)*, eine Einrichtung zur Erstellung eines kohärenten, einheitlichen Normenwerkes zur Anwendung in allen europäischen Raumfahrtprojekten, hat am 1996-11-13 vom *Europäischen Komitee für Normung (CEN)* das Mandat (M/237) erhalten, für den CEN-Subsektor T02 „Aerospace“ Europäische Normen (EN) auszuarbeiten.

Das ECSS-Dokument „ECSS-E-40A“ wurde in den Gremien des ECSS unter Mitwirkung deutscher Experten erarbeitet und nach Durchlauf der Formellen Abstimmung im CEN als Europäische Norm EN 14160:2001 angenommen.

Entsprechend Beschluss 57/9 des Technischen Ausschusses des Beirats des Normenausschuss Luft- und Raumfahrt (NL) im DIN Deutsches Institut für Normung e. V. sind die europäischen Luft- und Raumfahrt-Normungsergebnisse zweisprachig, in Deutsch und Englisch, in das Deutsche Normenwerk zu überführen. Aus diesem Grund wurde der Deutschen Fassung dieser EN-Norm die Englische Fassung hinzugefügt.

Fortsetzung 58 Seiten EN
und 58 Seiten Englische Fassung

Normenausschuss Luft- und Raumfahrt (NL) im DIN Deutsches Institut für Normung e. V.

— Leerseite —

Deutsche Fassung

**Raumfahrttechnik
Software**

Space engineering —
Software

Ingénierie spatiale —
Logiciel

Diese Europäische Norm wurde vom CEN am 11. November 2001 angenommen.

Die CEN-Mitglieder sind gehalten, die CEN/CENELEC-Geschäftsordnung zu erfüllen, in der die Bedingungen festgelegt sind, unter denen dieser Europäischen Norm ohne jede Änderung der Status einer nationalen Norm zu geben ist. Auf dem letzten Stand befindliche Listen dieser nationalen Normen mit ihren bibliographischen Angaben sind beim Management-Zentrum oder bei jedem CEN-Mitglied auf Anfrage erhältlich.

Diese Europäische Norm besteht in drei offiziellen Fassungen (Deutsch, Englisch, Französisch). Eine Fassung in einer anderen Sprache, die von einem CEN-Mitglied in eigener Verantwortung durch Übersetzung in seine Landessprache gemacht und dem Management-Zentrum mitgeteilt worden ist, hat den gleichen Status wie die offiziellen Fassungen.

CEN-Mitglieder sind die nationalen Normungsinstitute von Belgien, Dänemark, Deutschland, Finnland, Frankreich, Griechenland, Irland, Island, Italien, Luxemburg, Niederlande, Norwegen, Österreich, Portugal, Schweden, Schweiz, Spanien, der Tschechischen Republik und dem Vereinigten Königreich.



EUROPÄISCHES KOMITEE FÜR NORMUNG
EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION

Management-Zentrum: rue de Stassart, 36 B-1050 Brüssel

Inhalt

	Seite
Vorwort.....	3
Einleitung	4
1 Anwendungsbereich	5
2 Normative Verweisungen	5
3 Begriffe und Abkürzungen	6
3.1 Begriffe.....	6
3.2 Abkürzungen	8
4 Software-Engineering für Raumfahrtssysteme.....	9
4.1 Einleitung	9
4.2 Gliederung der Norm	10
4.3 Software-Engineeringprozesse	10
4.3.1 Allgemeines	10
4.3.2 Anforderungsbezogenes Engineering.....	13
4.3.3 Design-Engineeringprozess.....	13
4.3.4 Software-Verifizierung und -Validierung (-Qualifizierung).....	14
4.3.5 Engineering bei der Softwareanwendung	14
4.3.6 Softwarepflege.....	14
4.4 Zusammenhang mit Normen für das Raumfahrt-Projektmanagement und die Raumfahrt- Produktsicherung.....	15
4.4.1 Allgemeines	15
4.4.2 Softwareproduktsicherung	15
4.4.3 Softwareprojektmanagement.....	15
4.5 Verifizierung.....	16
5 Allgemeine Anforderungen	17
5.1 Einleitung	17
5.2 Systemengineering	17
5.2.1 Einleitung	17
5.2.2 Analyse der Systemanforderungen.....	17
5.2.3 Analyse der Systemstruktur	18
5.2.4 Anforderungen an Software-Verifizierung und -Validierung auf Systemebene	18
5.2.5 Softwareintegration auf Systemebene.....	19
5.2.6 Software unter Betriebsbedingungen	20
5.3 Software-Management.....	21
5.3.1 Einleitung	21
5.3.2 Software-Lebenszyklus	21
5.3.3 Schnittstellen-Management	23
5.3.4 Management der technischen und finanziellen Rahmenbedingungen	24
5.4 Anforderungsbezogenes Engineering	24
5.4.1 Einleitung	24
5.4.2 Analyse der Softwareanforderungen	24
5.4.3 Design der Softwarearchitektur auf übergeordneter Ebene.....	25
5.4.4 Software-Verifizierung und -Validierung	27
5.5 Design-Engineering	27
5.5.1 Einleitung	27
5.5.2 Design von Softwareeinheiten	27
5.5.3 Codierung und Testen	28
5.5.4 Integration.....	29
5.6 Software-Verifizierung und -Validierung (-Qualifizierung).....	31
5.6.1 Einleitung	31
5.6.2 Meilensteine.....	31
5.6.3 Verifizierungs- und Validierungsprozesse	32
5.6.4 Vorbereitende Arbeiten zur Verifizierung	32
5.6.5 Vorbereitende Arbeiten zur Validierung	33
5.6.6 Durchführung der Verifizierung.....	34
5.6.7 Durchführung der Validierung	36

	Seite
5.6.8 Gemeinsame technische Reviews	38
5.7 Software-Engineering unter Betriebsbedingungen.....	39
5.7.1 Einleitung.....	39
5.7.2 Softwareanwendung.....	39
5.7.3 Betriebliche Durchführung	39
5.7.4 Betriebstest	40
5.7.5 Systembetrieb	40
5.7.6 Anwenderbetreuung	40
5.8 Softwarepflege	41
5.8.1 Einleitung.....	41
5.8.2 Durchführung der Softwarepflege.....	41
5.8.3 Problem- und Änderungsanalyse.....	42
5.8.4 Durchführung von Änderungen	42
5.8.5 Softwarepflegereview und Abnahme	43
5.8.6 Softwaremigration.....	43
5.8.7 Aussondern von Software	44
6 Besondere Anforderungen	44
6.1 Einleitung.....	44
6.2 Software für das Raumsegment	44
6.2.1 Allgemeines	44
6.2.2 Kritische Funktionen	44
6.2.3 Systemschnittstellen	45
6.2.4 Softwareänderungen während des Flugs.....	45
6.3 Software für das Bodensegment.....	46
6.4 Wiederverwendung von Software	46
6.4.1 Allgemeines	46
6.4.2 Entwicklung von Software für Wiederverwendung	46
6.4.3 Wiederverwendung der Software von anderen Projekten	46
6.5 Schnittstelle Mensch-Maschine.....	47
6.6 Kritische Software.....	48
Anhang A (normativ) Softwaredokumentation.....	49
A.1 Einleitung.....	49
A.2 Grundanforderungen (RB)	49
A.2.1 Allgemeines	49
A.2.2 Anforderungen für das SRR	49
A.2.3 Schnittstellenanforderungsdokument (IRD)	50
A.3 Technische Spezifikation (TS)	50
A.3.1 Allgemeines	50
A.3.2 Schnittstellenkontrolldokument (ICD)	51
A.3.3 Plan zur Softwarepflege	51
A.3.4 Betriebshandbuch.....	51
A.4 Designbegründungsakte (DJF).....	52
A.5 Designdefinitionsakte (DDF).....	53
A.6 Dokumentation auf Systemebene	54
A.6.1 Einleitung.....	54
A.6.2 Dokumentation zu Softwareanwendung und -pflege, -migration und -aussonderung	54
Anhang B (informativ) Querverweise (Anforderungen).....	55
Anhang C (informativ) Verweisungen auf andere Normen.....	57
Literaturhinweise	58
Bilder	
Bild 1 — Modell der Kunde-Lieferant-Beziehung	11
Bild 2 — Überblick über Software-Entwicklungsprozesse	12
Bild 3 — Grenzen der Projektphasen	12
Bild 4 — Darstellung verschiedener Software-Lebenszyklen	13
Bild A.1 — Überblick über Software-Engineeringdokumente	49

Vorwort

Diese Europäische Norm wurde von CMC ausgearbeitet.

Diese Europäische Norm muss den Status einer nationalen Norm erhalten, entweder durch Veröffentlichung eines identischen Textes oder durch Anerkennung bis Juni 2002, und etwaige entgegenstehende nationale Normen müssen bis Juni 2002 zurückgezogen werden.

Sie basiert auf einer früheren Version¹⁾, die von der ECSS-Normungsarbeitsgruppe „Software-Engineering“ ausgearbeitet, vom Technischen ECSS-Ausschuss überprüft und vom ECSS-Lenkungsgremium verabschiedet wurde. Die Europäische Kooperation für Raumfahrtnormung (ECSS) ist eine Kooperation der Europäischen Raumfahrtagentur, nationaler Raumfahrtagenturen und europäischer Industrievereinigungen mit dem Ziel der Ausarbeitung und Pflege von einheitlichen Normen.

Diese Norm gehört zu einer Reihe von Raumfahrtnormen, die sich mit dem Management, der Technik (Engineering) und der Produktsicherung in Raumfahrtprojekten und -anwendungen befassen.

Diese Norm formuliert Anforderungen als Festlegungen dessen, was erreicht und weniger, wie die erforderliche Arbeit organisiert und durchgeführt werden soll. Dadurch können bestehende Organisationsstrukturen und -methoden beibehalten werden, wo sie effektiv sind, und sich weiterentwickeln, soweit erforderlich, ohne die Normen neu abfassen zu müssen.

Bei der Abfassung dieser Norm wurde die vorliegende EN-ISO-9000-Normenreihe berücksichtigt.

Anhang A ist normativ. Die Anhänge B und C sind informativ:

Diese Norm enthält Literaturhinweise.

Entsprechend der CEN/CENELEC-Geschäftsordnung sind die nationalen Normungsinstitute der folgenden Länder gehalten, diese Europäische Norm zu übernehmen: Belgien, Dänemark, Deutschland, Finnland, Frankreich, Griechenland, Irland, Island, Italien, Luxemburg, Niederlande, Norwegen, Österreich, Portugal, Schweden, Schweiz, Spanien, die Tschechische Republik und das Vereinigte Königreich.

Einleitung

Diese Norm greift auf Verfahren zurück, wie sie bei der Entwicklung von Raumfahrtssystemen zur Anwendung kommen, und leitet daraus Anforderungen an den Software-Engineeringprozess ab. Unter Verwendung der Anforderungen aus anderen Bereichen des Normenwerks der Europäischen Raumfahrt beschreibt diese Norm zusammenhängend und umfassend die Grundlagen für das Software-Engineering bei Raumfahrtprojekten.

Diese Norm soll Kunden bei der Formulierung ihrer Anforderungen wie auch Lieferanten bei der Reaktion auf die Kundenwünsche und die Durchführung der Arbeiten unterstützen.

Da diese Norm nicht als Lehrbuch für Computertechnik konzipiert ist, enthält sie auch keine entsprechenden Inhalte. Vielmehr werden beim Lesen und Anwenden dieser Norm allgemeine Kenntnisse der Computertechnik vorausgesetzt.

1) ECSS-E-40A

1 Anwendungsbereich

Diese Europäische Norm legt den Software-Engineeringprozess für die Raumfahrt sowie seine Schnittstellen mit den Raumfahrtnormen fest, die das Management (EN 13290) und die Produktsicherung (EN 13291) betreffen und erläutert den Umfang, in dem sie für den Engineeringprozess gelten.

Diese Norm gilt für alle Segmente eines Raumfahrtsystems, d. h., sie umfasst das Raumsegment, das Startservicesegment und das Bodensegment.

Diese Norm behandelt alle Aspekte des Software-Engineering für die Raumfahrt wie Anforderungen, Design, Produktion, Verifizierung und Validierung, Übertragung, Betrieb und Softwarepflege.

Diese Norm gilt für Software, wie sie für Raumfahrtprojekte entwickelt wird, d. h. die „Produktsoftware des Raumfahrtsystems“. Sie bezieht sich nicht auf die Entwicklung von Software für Bereiche außerhalb des Normensystems für die Raumfahrt, wie dies zum Beispiel für die Entwicklung von Softwarepaketen für handelsübliche Zwecke gilt, bei denen Software für einen größeren Markt und nicht nur für einen einzelnen Kunden entwickelt wird, wo die Grundanforderungen aus einer Marktanalyse in Verbindung mit einer Marketingstrategie entwickelt werden.

Diese Norm behandelt auch nicht Softwareprodukte wie Managementinformationssysteme (z. B. Finanzen, Planung), technische Informationssysteme (z. B. CAD/CAM, Analysepakete) und unterstützende Softwareprodukte für Dokumentationssysteme, Datenbanksysteme und Tabellenkalkulationsprogramme. Diese werden üblicherweise im Zusammenhang mit der Beschaffung oder Anpassung vorhandener Produkte entwickelt, aber nicht im Rahmen der Entwicklung von Raumfahrtsystemen. Derartige Softwareprodukte werden jedoch häufig zur Unterstützung der Infrastruktur für Raumfahrtsysteme eingesetzt.

Aus Sicht eines speziellen Projekteinhalts sollten die in dieser Norm definierten Anforderungen angepasst werden, um sie mit den tatsächlichen Anforderungen eines bestimmten Projektprofils und spezieller Projektumstände in Übereinstimmung zu bringen.

ANMERKUNG Anpassung ist ein Prozess, in dem einzelne Anforderungen oder Spezifikationen, Normen und ähnlichen Dokumenten nach Bewertung für ein bestimmtes Projekt ausgewählt und in Ausnahmefällen auch gestrichen, modifiziert oder indem neue Anforderungen ergänzt werden.

2 Normative Verweisungen

Diese Europäische Norm enthält durch datierte oder undatierte Verweisungen Festlegungen aus anderen Publikationen. Diese normativen Verweisungen sind an den jeweiligen Stellen im Text zitiert, und die Publikationen sind nachstehend aufgeführt. Bei datierten Verweisungen gehören spätere Änderungen oder Überarbeitungen nur zu dieser Europäischen Norm, falls sie durch Änderung oder Überarbeitung eingearbeitet sind. Bei undatierten Verweisungen gilt die letzte Ausgabe der in Bezug genommenen Publikation (einschließlich Änderungen).

EN 13290-1:1999, *Raumfahrt-Projektmanagement — Allgemeine Anforderungen — Teil 1: Grundsätze und Verfahrensweise.*

EN 13290-2:2001, *Raumfahrt-Projektmanagement — Allgemeine Anforderungen — Teil 2: Projektstrukturen.*

EN 13290-3:2001, *Raumfahrt-Projektmanagement — Allgemeine Anforderungen — Teil 3: Projektorganisation.*

EN 13290-4:2001, *Raumfahrt-Projektmanagement — Allgemeine Anforderungen — Teil 4: Projektphaseneinteilung und -planung.*

EN 13290-5:2001, *Raumfahrt-Projektmanagement — Allgemeine Anforderungen — Teil 5: Konfigurationsmanagement.*

EN 13290-6:2001, *Raumfahrt-Projektmanagement — Allgemeine Anforderungen — Teil 6: Informations-/Dokumentationsmanagement.*

EN 13290-7:2001, *Raumfahrt-Projektmanagement — Allgemeine Anforderungen — Teil 7: Kosten- und Zeitplanmanagement.*

EN 13291-2-2²⁾, *Raumfahrtproduktsicherung — Allgemeine Anforderungen — Teil 2: Qualitätssicherung*.

EN 13292:1999, *Raumfahrttechnik (Engineering) — Grundsätze und Verfahrensweise*.

EN 13701, *Raumfahrtsysteme — Glossar*.

ISO 8402:1994, *Quality management and quality assurance — Vocabulary*.

ISO/IEC 12207:1995, *Information technology — Software life cycle processes*.

ISO 16091:-²⁾, *Space project management — Integrated logistic support*.

ECSS-E-10, *Raumfahrttechnik (Engineering) — Systemtechnik (Systemengineering)*.

ECSS-E-40-01²⁾, *Raumfahrttechnik (Engineering) — Software für das Raumsegment*.

ECSS-E-40-03²⁾, *Raumfahrttechnik (Engineering) — Software für das Bodensegment*.

ECSS-Q-80, *Raumfahrtproduktsicherung — Softwareproduktsicherung*.

3 Begriffe und Abkürzungen

3.1 Begriffe

Für die Anwendung dieser Europäischen Norm gelten die in EN 13701 angegebenen und die folgenden Begriffe.

3.1.1

übergeordnete Architektur

von der (den) höchsten Ebene(n) der Komponenten eines Programms oder Systems vorgegebene Struktur, deren Wechselbeziehungen sowie die Grundlagen und Regeln, die ihr Design und ihre zeitabhängige Entwicklung bestimmen

3.1.2

integrierte Software (veraltet)

ANMERKUNG 1 Dieser Begriff dient gelegentlich dafür, dass Software auf unterschiedlichen Ebenen Bestandteil eines Systems ist.

ANMERKUNG 2 Der Begriff wird auch verwendet, um hervorzuheben, dass die betreffende Software umfangreiche Anforderungen an die Hardwareschnittstelle oder das Echtzeitverhalten stellt. Er ist rein beschreibend.

3.1.3

Verantwortlicher für die Softwarepflege

Stelle, die die Softwarepflege durchführt

[ISO/IEC 12207:1995]

3.1.4

Reservenphilosophie

Grundprinzip für das Festlegen von Reserven, die Leistungsparameter und Computerausrüstung in einer bestimmten Entwicklungsphase zugeordnet werden, und wie diese Reserven im Projektablauf angewendet werden

3.1.5

Migration

Umstellung eines Softwareprodukts auf eine vollständig neue Umgebung

2) In Vorbereitung

3.1.6**Einzelvorgabe**

Belastungstest einzelner Parameter

3.1.7**Softwarekomponente**

allgemeine Bezeichnung für einen Bestandteil eines Softwaresystems

Softwarekomponenten sind zusammengesetzt und können zur Bildung neuer Komponenten zerlegt werden. In der Produktionsphase haben Komponenten die Form von Modulen, Aufgaben oder Programmen, von denen jede wiederum Konfigurationseinheiten sein können. Die Anwendung des Begriffs erfolgt allgemeiner als bei ANSI/IEEE, wo eine Komponente als ein „Grundbaustein eines Systems oder Programms“ definiert wird. Nach dieser Norm sind Komponenten jedoch keine Grundbausteine, da sie zerlegt werden können.

3.1.8**Softwareeinheit**

siehe 3.1.11 Softwareprodukt

3.1.9**softwareintensives System**

Softwareprodukt für die Raumfahrt, das vorwiegend aus Softwareelementen besteht

ANMERKUNG Bei Systemen der Raumfahrttechnik werden Untersysteme meistens von Software dargestellt. Die Mehrzahl der Schnittstellen sind hier Programmschnittstellen.

3.1.10**Überprüfbarkeit der Software**

Eigenschaft eines Systems, die erlaubt, dass Untersuchungen der Ergebnisvariablen zur Bestimmung der Anfangswerte der Zustandsvariablen ausreichen

3.1.11**Softwareprodukte**

Computerprogramme, Verfahren und gegebenenfalls zugehörige Dokumentation und Daten

3.1.12**Softwareelement**

einzelne kompilierbarer Teil eines Codes

[ISO/IEC 12207:1995]

In dieser Norm wird keine Unterscheidung zwischen einem Softwareelement und einer Datenbank vorgenommen, da für beide die gleichen Anforderungen zutreffen.

3.1.13**Belastungstest**

Test, mit dem ein System oder eine Softwarekomponente bis an die Grenzen der dafür festgelegten Anforderungen oder darüber hinaus beansprucht wird

3.1.14**Validierung**

Bestätigen aufgrund einer Untersuchung und durch Bereitstellung eines Nachweises, dass die besonderen Forderungen für einen speziellen beabsichtigten Gebrauch erfüllt worden sind

[ISO 8402:1994]

ANMERKUNG Validierung für Software: Bestätigen, dass die Grundanforderungen für Funktion und Leistung des Endprodukts vollständig erfüllt sind.

3.1.15

Verifizierung

Bestätigen aufgrund einer Untersuchung und durch Bereitstellung eines Nachweises, dass festgelegte Forderungen erfüllt worden sind

[ISO 8402:1994]

ANMERKUNG Verifizierung für Software: Bestätigen, dass für jede Aktivität ausreichende Spezifikationen und Vorgaben bestehen und dass die Ergebnisse korrekt sind und den Spezifikationen und Vorgaben entsprechen.

3.2 Abkürzungen

In dieser Norm gelten folgende Abkürzungen.

Abkürzung Bedeutung

AR	(Acceptance Review)	Abnahmereview
		ANMERKUNG Zur Bezeichnung von ARs, die nur Softwareprodukte betreffen, darf die Abkürzung SW-AR verwendet werden.
CDR	(Critical Design Review)	Kritisches Design-Review
		ANMERKUNG Zur Bezeichnung von CDRs, die nur Softwareprodukte betreffen, darf die Abkürzung SW-CDR verwendet werden.
CJF	(Change Justification File)	Änderungsbegründungsakte
COTS	(Commercial off-the-Shelf Software)	Handelsübliche (gebrauchsfähige) Software
		ANMERKUNG Bezeichnet fertige Softwareprodukte, die von einem Dritten hergestellt wurden.
CPU	(Central Processing Unit)	Zentraleinheit
DDF	(Design Definition File)	Designdefinitionsakte
DJF	(Design Justification File)	Designbegründungsakte
ICD	(Interface Control Document)	Schnittstellenkontrolldokument
IRB	(Interface Requirements Baseline)	Grundanforderungen an Schnittstellen
IRD	(Interface Requirements Document)	Schnittstellenanforderungsdokument
ISV	(Independent Software Validation)	Unabhängige Software-Validierung
ISVV	(Independent Software Verification and Validation)	Unabhängige Software-Verifizierung und -Validierung
MMI	(Man-Machine Interface)	Schnittstelle Mensch-Maschine
MOTS	(Modifiable off-the-Shelf)	Kommerzielle Standardsoftware, modifizierbar
MP	(Maintenance Plan)	Plan zur Softwarepflege
OP	(Operational Plan)	Plan zur Softwareanwendung
ORR	(Operational Requirements Review)	Betriebsanforderungsreview
PDR	(Preliminary Design Review)	Vorläufiges Design-Review
		ANMERKUNG Zur Bezeichnung von PDRs, die nur Softwareprodukte betreffen, darf die Abkürzung SW-PDR verwendet werden.

QR	(Qualification Review)	Qualifikationsreview ANMERKUNG Zur Bezeichnung von QRs, die nur Softwareprodukte betreffen, darf die Abkürzung SW-QR verwendet werden.
RB	(Requirements Baseline)	Grundanforderungen
SDE	(Software Development Environment)	Umgebung für die Softwareentwicklung ANMERKUNG Softwarewerkzeuge, die den Software-Engineeringprozess unterstützen.
SRR	(System Requirements Review)	Systemanforderungsreview ANMERKUNG Zur Bezeichnung von SRRs, die nur Softwareprodukte betreffen, darf die Abkürzung SW-SRR verwendet werden.
SW	(Software)	Software
TS	(Technical Specification)	Technische Spezifikation

4 Software-Engineering für Raumfahrtssysteme

4.1 Einleitung

Dieser Abschnitt erläutert den Aufbau dieser Norm sowie die Grundlagen des Software-Engineeringprozesses für die Raumfahrt.

Software-Engineering ist Teil des allgemeinen Engineeringprozesses für Raumfahrtssysteme. Deshalb beschreibt dieser Abschnitt die Zusammenhänge zwischen beiden Bereichen.

Die Norm zum Software-Engineering unterscheidet sich von den anderen in den Europäischen Normen für Raumfahrt behandelten Engineeringbereichen in einem wesentlichen Aspekt: Software selbst erzeugt keine Wärme, besitzt keine Masse und keine sonstigen physikalischen Eigenschaften. Software-Engineering ist eine rein intellektuelle Tätigkeit, deren wesentliches Ergebnis die Dokumentation ist. Fasst man Software als eine besondere Form elektronischer Dokumente auf, so sind alle sichtbaren Ergebnisse als Dokumentation zu betrachten.

Deshalb legt diese Norm ihren Schwerpunkt auf die Struktur und den Inhalt der erzeugten Dokumentation.

Software kann zur Realisierung hochkomplexer Funktionen benutzt werden. Ihre Eigenschaft, ein hohes Maß an Komplexität flexibel zu bewältigen, macht Software zu einem wesentlichen und in seiner Bedeutung wachsenden Bestandteil von Produkten des Raum- und Bodensegments. Bei Raumfahrtssystemen findet man Software-Engineering auf allen Funktionsebenen, angefangen von der Systemebene bis hinunter zur Firmware für einzelne Teile des Systems.

In dieser Norm wird deshalb der anforderungsbezogene Engineeringprozess, in dem die Anforderungen und Spezifikationen für Software festgelegt werden, besonders eingehend behandelt, denn dieser Prozess hat einen großen und oft unterschätzten Anteil an der Entwicklung von Software für Raumfahrtssysteme.

Wegen der Komplexität der Funktions- und Leistungsanforderungen werden für Software für das Raumsegment besonders hohe Maßstäbe an die Verifizierung und Validierung angelegt. Schließlich kann bei einer Raumfahrtmission die Funktion der Software von ausschlaggebender Bedeutung sein.

Besondere Probleme wirft auch die Softwarepflege für Raumfahrtssysteme auf, weil hier mit einer Betriebslebensdauer zu rechnen ist, die die für sonstige Computer-Softwareprodukte weit übertrifft. Beim Raumsegment kompliziert sich das Problem weiter, da Software im Allgemeinen der einzige Teil des Raumsegments ist, der nach dem Start einer umfangreichen Überarbeitung oder Reparaturen unterzogen, manchmal sogar völlig neu geschrieben wird. In Extremfällen wird sogar die Raumfahrtmission neu konzipiert, wobei nach dem Start eine neue Software realisiert wird. Für die Software für das Bodensegment gilt dies in gleicher Weise.

4.2 Gliederung der Norm

Diese Norm gliedert sich in zwei Hauptteile:

- **Allgemeine Anforderungen:** Dies sind die normativen Grundanforderungen, die für jede Tätigkeit im Software-Engineering für Raumfahrtsysteme zutreffen.
- **Besondere Anforderungen:** Dies sind zusätzliche Anforderungen für besondere Anwendungsbereiche. Diese Anforderungen gelten immer, haben aber nur für bestimmte Gebiete oder Anwendungsbereiche praktische Bedeutung. Diese Unterscheidung dient dazu, die Grundanforderungen knapp und klar zu halten.

Anhang A enthält eine Übersicht über Softwaredokumentation.

Diese Norm stützt sich weitgehend auf die Norm ISO/IEC 12207:1995, die einen allgemeinen, international anerkannten Rahmen für die Terminologie und die Beschreibung von Engineeringprozessen liefert. Ergänzend sind im Anhang B Querverweise zwischen dieser Norm und ISO/IEC 12207:1995 angegeben.

4.3 Software-Engineeringprozesse

4.3.1 Allgemeines

Die in dieser Norm beschriebenen Software-Engineeringprozesse beruhen auf der Terminologie und den Anforderungen der Normenreihe für das Raumfahrt-Projektmanagement (vor allem EN 13290-3:2001, EN 13290-4:2001, EN 13290-5:2001 und EN 13290-6:2001) sowie den allgemeinen Anforderungen an Engineeringprozesse in EN 13292:1999. Diese Anforderungen finden bei der Beschreibung des Engineeringprozesses auf der obersten Ebene Anwendung. Vor diesem Hintergrund werden die (in den folgenden Abschnitten im Einzelnen behandelten) Prozesse sowie die Schnittstelle auf der obersten Ebene zwischen den Software-Engineeringprozessen und anderen Entwicklungsprozessen in der Raumfahrt definiert.

In dieser Norm steht die Beziehung zwischen Kunde und Lieferant wie auch bei sonstigen Softwareentwicklungen im Vordergrund der Betrachtung. Die organisatorischen Aspekte dieser Beziehung werden in EN 13290-3:2001 behandelt. Ganz allgemein bezieht der Kunde gleichzeitig zwei eng miteinander verknüpfte Produkte: die Hardware und die Software für ein System, ein Untersystem, eine Produktgruppe, ein Gerät oder eine Baugruppe (siehe EN 13292:1999). Der Begriff der Beziehung zwischen Kunde und Lieferant wird rekursiv angewendet, d. h., der Kunde kann selbst Lieferant für eine übergeordnete Ebene im Raumfahrtsystem sein (siehe Bild 1). Der Software-Kunde besitzt daher zwei wichtige Schnittstellen. Die erste Schnittstelle besteht zu seinen Software- und Hardware-Lieferanten, was eine Funktionsanalyse einschließt, die für eine angemessene Zuordnung von Funktions- und Leistungsanforderungen zu seinen Lieferanten erforderlich ist. Die andere Schnittstelle besteht bei Betrachtung seiner Rolle als Lieferant für eine übergeordnete Ebene, wo er sicherstellen muss, dass die Anforderungen seiner Ebene an das System gleichermaßen berücksichtigt werden.

Der Kunde legt die Funktions- und Leistungsanforderungen für Hardware und Software auf der Grundlage von Verfahren des Systemengineering fest. Er kontrolliert auch die Schnittstelle zwischen Software und Hardware. Softwareprodukte werden für die verschiedenen Systeme definiert. Trotzdem ist es wichtig, die Programm-Schnittstellen für alle Ebenen, in denen sie auftreten, zu betrachten. Mit diesem Prozess werden die Kundenanforderungen definiert, die dann als Ausgangspunkt für das Software-Engineering dienen.

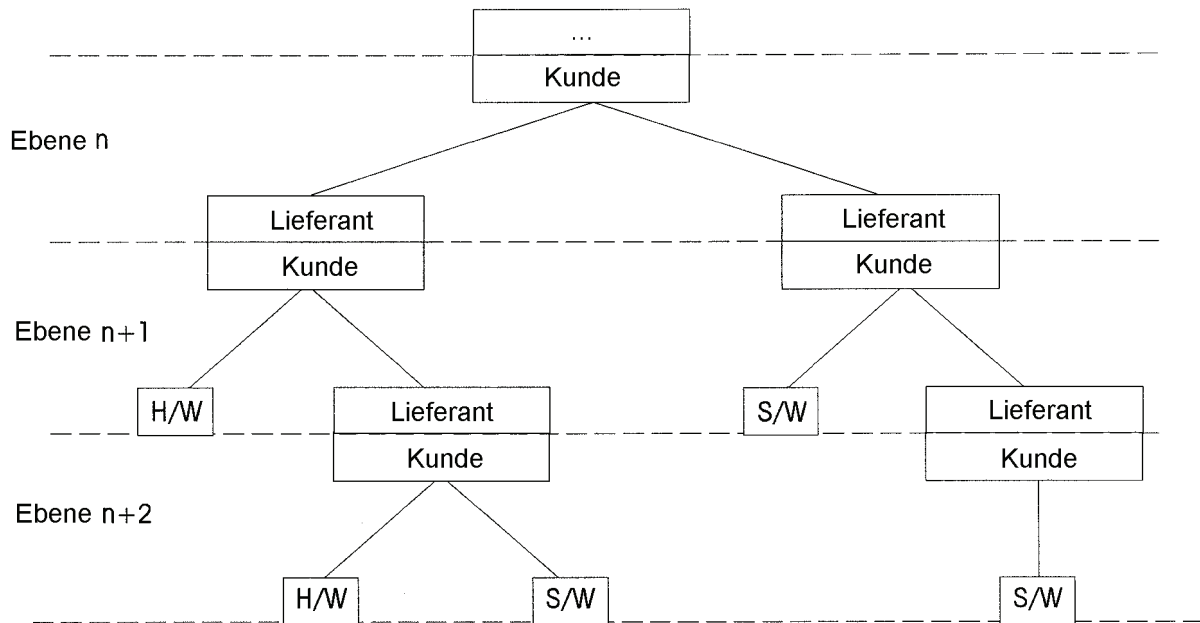


Bild 1 — Modell der Kunde-Lieferant-Beziehung

Von besonderer Bedeutung für die Wechselbeziehung zwischen Kunde und Lieferant sind Reviews. Nach EN 13290-4:2001 gelten für den Software-Engineeringprozess Reviews mit der Bezeichnung SRR, PDR, CDR, QR und AR, die für Software anwendbar sind. Reviews finden auf allen Ebenen der Beziehung „Kunde-Lieferant“ statt, wobei ihre Reihenfolge sich an der Struktur der Systemebenen orientiert. Diese Norm gilt für jegliche Ebene, ohne Berücksichtigung der Beziehung zu anderen Reviews im Raumfahrtssystem. Die Zusammenhänge zeigt Bild 2. Die häufig verwendeten Bezeichnungen für Missionsphasen (z. B. 0, A, B) gelten nur für diese, spielen aber keine direkte Rolle im Software-Engineering. Das bedeutet, dass die Software-Engineeringprozesse mit Reviews und zugehörigen Meilensteinen nach dieser Norm nicht als Missionsphasen höherer Systemebenen zu planen sind; stattdessen sollten sie lediglich mit Bezug auf die Entwicklungsprozesse der unmittelbar übergeordneten Ebene geplant werden.

Der Begriff des Engineeringprozesses ist von grundlegender Bedeutung für diese Norm, da durch ihn einerseits die Mittel zur Beschreibung der allgemeinen Grenzen und Schnittstellen für den Software-Engineeringprozess auf der Systemebene bereitgestellt werden, andererseits dem Lieferanten die notwendige Freiheit zur Realisierung der einzelnen Aktivitäten eingeräumt wird, die mit den Prozessen verbunden sind. Diese ist aufgrund der Notwendigkeit, die Arbeit nach einem klar beschriebenen Software-Lebenszyklus³⁾ zu organisieren, für das Software-Engineering besonders wichtig. Unterschiedliche Arten von Software-Lebenszyklen sind hier gleichermaßen zu berücksichtigen, sowohl im Hinblick auf eine effiziente Arbeitsorganisation wie auch aus Gründen, die mit der Wettbewerbsfähigkeit und Auswahl der Technik im Software-Engineering im Zusammenhang stehen. Die Anforderungen dieser Norm erlauben die Verwendung unterschiedlicher Arten von Software-Lebenszyklen. Bild 3 stellt die für den Prozess geltenden Grenzen der Projektphasen dar, Bild 4 zeigt Beispiele für hier in Frage kommende Varianten.

3) Software-Lebenszyklen

Es gibt umfangreiche technische Literatur zu den unterschiedlichen Arten von Software-Lebenszyklen. Diese Norm empfiehlt keinen bestimmten Lebenszyklus. Sie legt stattdessen die Anforderungen fest, die bei der Auswahl eines Lebenszyklus für ein bestimmtes Softwareprojekt gelten. Sie ist so zu treffen, dass der Software-Lebenszyklus mit der Gesamtorganisation und dem Gesamtmanagement des Raumfahrtprojekts kompatibel bleibt, und dass die Ergebnisse den Anforderungen der Raumfahrtnorm EN 13290 entsprechen. Zu den allgemein bekannten Arten von Software-Lebenszyklen gehören:

- der Wasserfall-Lebenszyklus,
- der inkrementale Lebenszyklus,
- der evolutionäre Lebenszyklus.

Die Anforderungen dieser Norm wurden auf diese Lebenszyklen abgestimmt; die Verwendung anderer Lebenszyklen ist möglich.

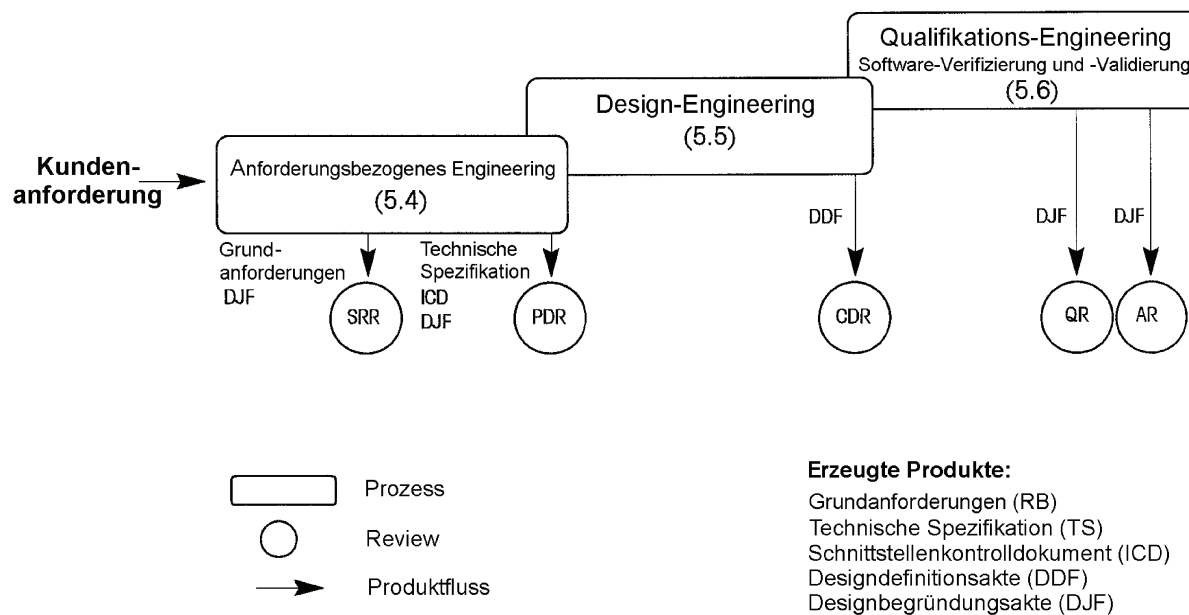


Bild 2 — Überblick über Software-Entwicklungsprozesse

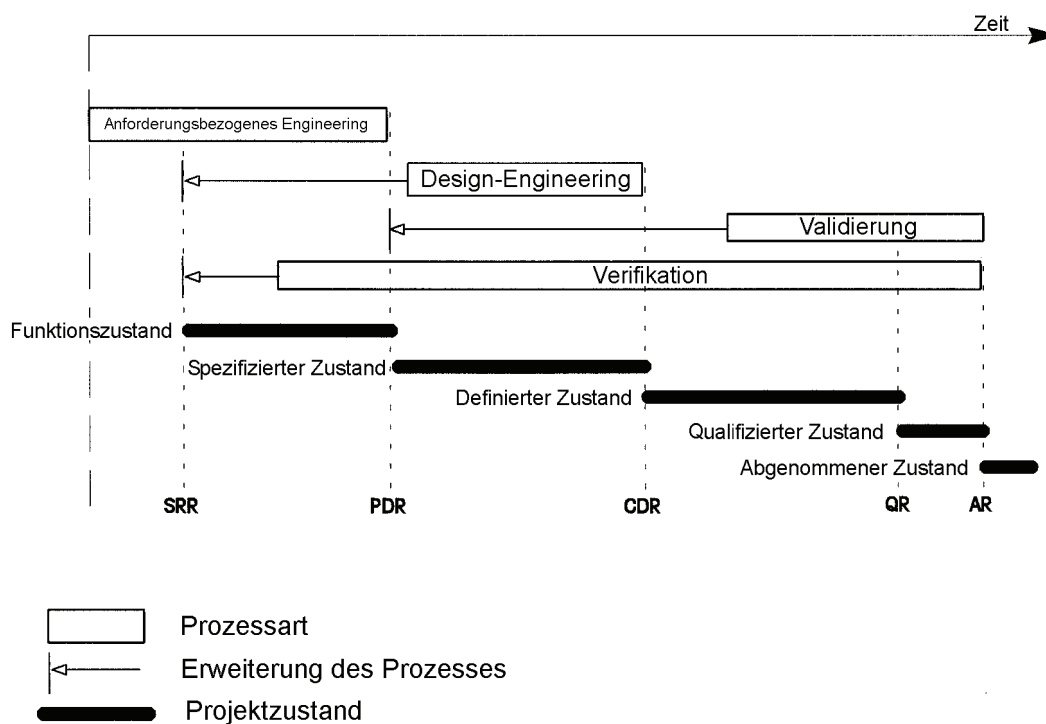


Bild 3 — Grenzen der Projektphasen

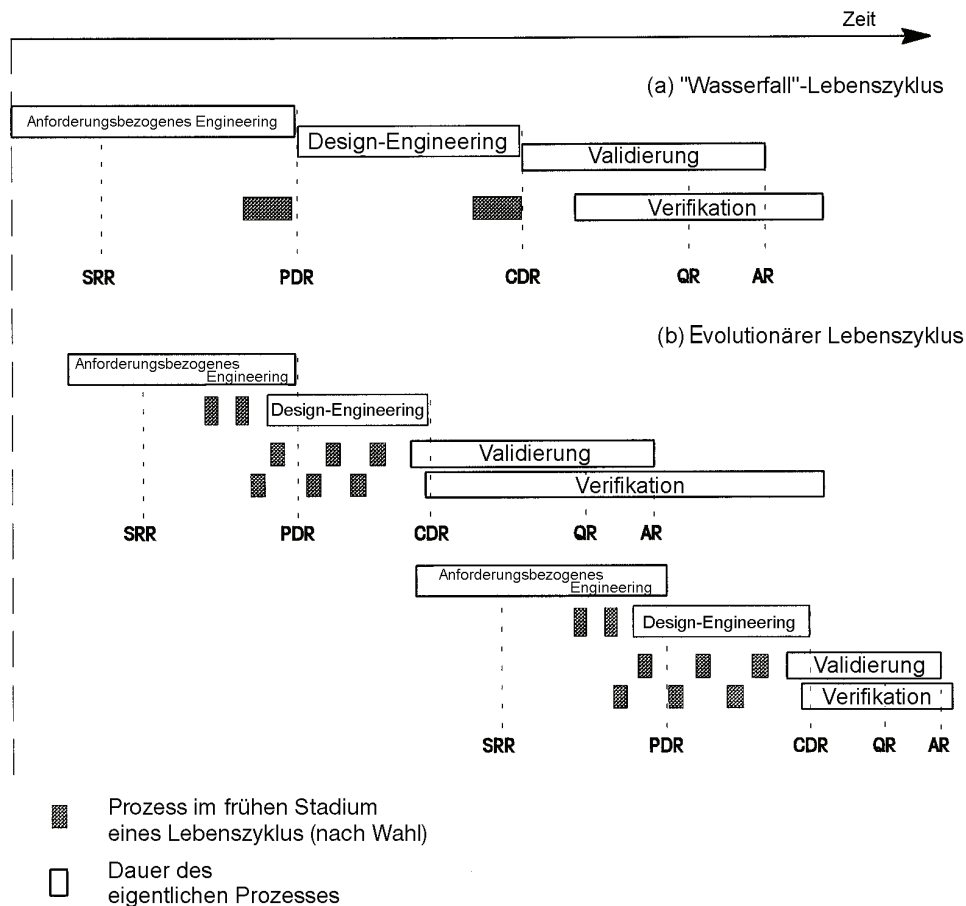


Bild 4 — Darstellung verschiedener Software-Lebenszyklen

4.3.2 Anforderungsbezogenes Engineering

Im ersten Teil des anforderungsbezogenen Engineeringprozesses werden die Informationen erzeugt, die als Input für das Systemanforderungsreview (SRR) notwendig sind. Daraus ergeben sich die Grundanforderungen für Funktion und Leistung bei der Softwareentwicklung sowie vorläufige Anforderungen an die Schnittstellen.

Der zweite Teil des Prozesses umfasst die Erstellung einer technischen Spezifikation (TS) als Reaktion des Lieferanten auf die Grundanforderungen. Dies darf gleichzeitig mit oder nach Aufstellung der Grundanforderungen erfolgen. Das Ergebnis ist der Software-Produktbaum. Die TS muss eine genaue und schlüssige Beschreibung der Funktionen, Leistungen, Kosten und der Pläne für Ablauf und Durchführung für alle Ebenen der zu entwickelnden Software enthalten. Mit diesem Prozess wird das vorläufige Schnittstellenkontrolldokument (ICD) erzeugt.

Im Rahmen desselben Prozesses sind die Ergebnisse aller wichtigen Kompromisse, Machbarkeitsstudien, Entscheidungen zur Eigenentwicklung oder zum Fremdbezug sowie unterstützender technischer Beurteilungen in einer Designbegründungsakte (DJF) aufzuzeichnen.

Der anforderungsbezogene Software-Engineeringprozess wird durch das vorläufige Designreview (PDR) abgeschlossen. Der Input für das PDR ist die technische Spezifikation, das vorläufige ICD und die DJF. Mit dem PDR wird das Design der Architektur auf übergeordneter Ebene überprüft. Fordert der Kunde ein zusätzliches Review der Softwarearchitektur, so darf dieses in die Projektorganisation eingebunden werden.

Der Zustand des Softwareprojekts nach dem PDR wird als „spezifizierter Zustand“ bezeichnet.

4.3.3 Design-Engineeringprozess

Der in ECSS-E-10 beschriebene „Design- und Konfigurations-Engineeringprozess“ wird in der Softwareentwicklung als der „Design-Engineeringprozess“ bezeichnet.

Dieser Prozess sollte nicht vor dem SRR beginnen. Er darf zwar vor dem PDR beginnen, jedoch werden erst nach dem Review die Ergebnisse des anforderungsbezogenen Engineeringprozesses geprüft und als Grundlage für den Input in den Design-Engineeringprozess festgelegt.

In diesem Prozess wird in Erfüllung der in der TS, im ICD und in der DJF enthaltenen Anforderungen das Design aller Elemente des Software-Produktbaums erzeugt, was in der Designdefinitionsakte (DDF) zu dokumentieren ist. Die DDF enthält die Ergebnisse aller Ebenen des Design-Engineering einschließlich von Verzeichnissen mit Softwarecodes.

In der DJF werden im Rahmen des Prozesses auch die Gründe für wichtige Entscheidungen und Analysen genannt sowie Testergebnisse angegeben, die zeigen, dass das Design alle Anforderungen erfüllt. Die Ergebnisse dieses Prozesses sind gleichzeitig der Input für das kritische Designreview (CDR), das die Designphase abschließt. Bei großen Softwareprojekten müssen alle Software-Untersysteme einem CDR unterzogen werden, bevor sie in die nächsthöhere Systemebene eingebunden werden. Umfangreiche Softwareprojekte sollten in kleinere, überschaubare Einheiten unterteilt werden, die dann wie andere Untersysteme in Raumfahrtprojekten behandelt werden.

Der Zustand des Softwareprojekts nach dem CDR wird als „definierter Zustand“ bezeichnet.

4.3.4 Software-Verifizierung und -Validierung (-Qualifizierung)

Der Prozess der Software-Verifizierung und -Validierung kann jederzeit nach dem SRR beginnen.

Dieser Prozess dient als Bestätigung dafür, dass den Kundenanforderungen entsprochen wurde, dass alle Anforderungen erfüllt wurden und dass alle für das Design geltenden Randbedingungen berücksichtigt wurden.

Das Ergebnis dieses Prozesses wird in die DJF aufgenommen.

Der Prozess muss ein Qualifikationsreview (QR) mit der DJF als Vorgabe umfassen.

Der Zustand der Softwareprojekts nach dem QR wird als „qualifizierter Zustand“ bezeichnet.

Ein Teil dieses Prozesses ist die Übergabe der Software an den Kunden und deren Abnahme durch diesen. Abschluss bildet hier das Abnahmereview (AR), das nach dem QR durchgeführt werden darf. Das AR ist ein formaler Vorgang, mit dem das Softwareprodukt in seiner betrieblichen Umgebung bewertet wird. Es sollte durchgeführt werden, nachdem das Softwareprodukt fertig gestellt, dem Kunden übergeben und im Betrieb installiert wurde. Der Prozess der Software-Validierung ist mit dem AR abgeschlossen.

Der Zustand des Softwareprojekts nach dem AR wird als „abgenommener Zustand“ bezeichnet.

ANMERKUNG Der Begriff „Qualifizierungsengineering“ wird für Projekte zur Lieferung von Hardware oft gleich bedeutend mit „Verifizierungsengineering“ benutzt. In dieser Norm wird die Bezeichnung angewendet, um „die Gesamtheit aller mit der Verifizierung und Validierung“ zusammenhängenden Vorgänge zu nennen. Dies erfolgt in Übereinstimmung mit anderen Raumfahrtnormen außerhalb des Software-Engineering und soll Verwechslungen mit Tätigkeiten beim Verifizierungsengineering, das bei Softwareprojekten vielfach zur Anwendung kommt, vermeiden helfen.

4.3.5 Engineering bei der Softwareanwendung

Der Prozess der Softwareanwendung kann im Anschluss an das AR beginnen. Da die Software Teil eines Raumfahrtssystems ist, müssen sich die Einbindung und das Management der Softwareprodukte an den Anforderungen an das Gesamtsystem orientieren. Der Prozess ist nicht unmittelbar an die Missionsphase E gebunden, sondern er wird von den Anforderungen der jeweiligen Systemebene bestimmt, auf der das Softwareprodukt zur gegebenen Zeit im Einsatz ist.

4.3.6 Softwarepflege

Der Prozess der Softwarepflege beginnt nach dem Abschluss des AR.

Er kommt zum Tragen, wenn das Softwareprodukt als Folge einer Fehlerkorrektur, des Auftretens eines Problems, der Durchführung einer Verbesserung, eine Anpassung oder eine Änderung des Codes oder der zugehörigen Dokumentation erfährt. Der Prozess endet mit der Zurückziehung des Softwareprodukts.

ANMERKUNG Die Softwareanalyse (als allgemeiner Engineeringprozess, wie er in den Normen für das Raumfahrt-Projektmanagement behandelt wird) kommt beim anforderungsbezogenen Engineering wie auch Design-Engineeringprozessen zur Anwendung. Dieser Prozess bleibt ohne Ergebnis. Vielmehr werden die Ergebnisse der Analyse in die Ergebnisse des anforderungsbezogenen Engineering und des Design-Engineering einbezogen.

4.4 Zusammenhang mit Normen für das Raumfahrt-Projektmanagement und die Raumfahrt-Produktsicherung

4.4.1 Allgemeines

In diesem Abschnitt wird auf die Schnittstellen dieser Norm mit Normen für die Raumfahrt, und zwar den Normenreihen für das Projektmanagement und die Produktsicherung, eingegangen.

4.4.2 Softwareproduktsicherung

Anforderungen an die Softwareproduktsicherung sind in ECSS-Q-80 festgelegt, dem Basisdokument der Normenreihe für die Raumfahrt-Produktsicherung.

4.4.3 Softwareprojektmanagement

4.4.3.1 Allgemeines

Die für das Management von Raumfahrtprojekten geltenden Anforderungen werden in den Normen für das Raumfahrt-Projektmanagement festgelegt. In den folgenden Abschnitten wird beschrieben, in welchem Umfang diese Normen für das Management von Softwareprojekten gelten.

Darüber hinaus enthält 5.3 Querverweise auf andere Normen, die nicht der Management-Reihe angehören, weil sie nur für das Software-Projektmanagement gelten.

4.4.3.2 EN 13290-1:1999: Grundsätze und Verfahrensweisen

EN 13290-1:1999 ist ein Dokument übergeordneter Art, in dem Grundsätze des Projektmanagements und allgemeine Anforderungen festgelegt sind, die für alle Bereiche eines Raumfahrtprojekts einschließlich der Software gelten.

Die Norm behandelt auch das Risikomanagement. Einige Risikofaktoren wie die Überschreitung der reservierten Speicherkapazität oder der CPU-Auslastung gelten insbesondere für Software.

Die in der vorliegenden Norm verwendeten Begriffe „Kunde“ und „Lieferant“ werden in EN 13290-1:1999, 5.2, festgelegt.

4.4.3.3 EN 13290-2:2001: Projektstrukturen

Die Festlegungen von EN 13290-2:2001 gelten für Software unter Berücksichtigung ihrer spezifischen Eigenschaften.

Die Produkte eines Softwareprojekts sind in der Regel Dokumente (einschließlich Code), können aber bei softwareintensiven Systemen auch Geräte umfassen.

Die „Modellmatrix“ in EN 13290-2:2001, 5.3, behandelt Werkstoffmodelle und ist deshalb für Software nicht von Bedeutung. Diese Modelle dürfen jedoch nicht mit logischen und physikalischen Softwaremodellen verwechselt werden, die im Rahmen einer Softwarespezifikation bzw. beim Softwaredesign entwickelt werden können.

4.4.3.4 EN 13290-3:2001: Projektorganisation

EN 13290-3:2001 weist allen Projektbeteiligten ihre Aufgabe und Zuständigkeit zu und legt darüber hinaus Anforderungen an Softwareprojekte fest.

4.4.3.5 EN 13290-4:2001: Projektphaseneinteilung und -planung

EN 13290-4:2001 legt Anforderungen hinsichtlich Planung und Phaseneinteilung für das gesamte Raumfahrtprojekt fest. Einige Anforderungen betreffen die Softwareentwicklung, da sie auf jeder Ebene der Projektorganisation von Bedeutung sind.

Projektphasen, wie sie in EN 13290-4:2001 beschrieben sind, sind (Missions-) Phasen übergeordneter Art, die der Strukturierung des gesamten Raumfahrtprojekts dienen. Sie gelten nicht rekursiv für die Softwareentwicklung und dürfen nicht mit den Phasen verwechselt werden, die der Strukturierung der Lebenszyklen bei der Softwareentwicklung dienen und für die in dieser Norm keine besonderen Festlegungen getroffen wurden.

Genauso gelten die in EN 13290-4:2001 beschriebenen Reviews nicht direkt für Software, selbst wenn der Begriff des Reviews für alle Ebenen eines Raumfahrtprojekts gilt.

Die Akronyme „SRR“, „PDR“, „CDR“, „QR“ und „AR“ werden in EN 13290-4:2001 erläutert und dienen zur Bezeichnung technischer Reviews, wie sie in der Softwareentwicklung entsprechend 4.3 eingesetzt werden.

Diese Reviews müssen projektabhängig auf Reviews übergeordneter Ebenen abgestimmt werden. Abschnitt 6 enthält Schnittstellenanforderungen für bestimmte Arten der Software. 5.3 enthält softwarespezifische Anforderungen in Bezug auf die Phaseneinteilung und Reviews.

4.4.3.6 EN 13290-5:2001: Konfigurationsmanagement

In EN 13290-5:2001 sind auch Anforderungen hinsichtlich der Softwareentwicklung enthalten.

Ein Aspekt des Software-Konfigurationsmanagements ist, dass alle Konfigurationseinheiten als Dokumente (als Code) betrachtet werden können. Deshalb steht hinsichtlich der Software EN 13290-5:2001 in engem Zusammenhang mit EN 13290-6:2001. Danach wird die Software-Konfigurationssteuerung zu einem stark automatisierten Prozess, der sich in Bezug auf den allgemeinen Ansatz vereinfachen lässt.

4.4.3.7 EN 13290-6:2001: Informations-/Dokumentationsmanagement

Das Informations- und Dokumentationsmanagement verschafft insbesondere allen am Projekt Beteiligten den Zugang zu Informationen und sichert deren Kohärenz. Dies gilt auch für Softwareprojekte, was sich in entsprechenden Anforderungen in der Norm niederschlägt.

4.4.3.8 EN 13290-7:2001: Kosten- und Zeitplanmanagement

EN 13290-7:2001 enthält hier auch Anforderungen für Softwareprojekte; dabei spielen Anforderungen an das Zeitplanmanagement bei Software eher eine Rolle als Anforderungen hinsichtlich der Kosten.

4.4.3.9 ISO 16091: Integrierte Logistikunterstützung

ISO 16091 betrifft hauptsächlich große oder softwareintensive Systeme.

4.5 Verifizierung

Die Verifizierung unterstützt die Festlegung von Anforderungen, das Design, die Codierung, die Integration, den Betrieb und die Softwarepflege. Die Anforderungen nach ECSS-E-10A, 4.6, gelten mit folgender Einschränkung auch für Software:

„Mit Hilfe der Verifizierung wird festgestellt, ob Softwareprodukte die gestellten Anforderungen und Bedingungen erfüllen. Die Verifizierung sollte so früh wie möglich in das anforderungsbezogene Engineering und das Design-Engineering einbezogen werden.“

Beim Verifizierungsengineering gibt es unterschiedliche Grade der Unabhängigkeit der damit beauftragten Person. Es kann sich um ein und dieselbe Person, um eine andere Person in derselben Organisation oder um Dritte handeln. In Fällen, in denen die Verifizierung von einer Stelle durchgeführt wird, die vom Lieferanten unabhängig ist, wird von einer unabhängigen Software-Verifizierung und -Validierung gesprochen. Die einzelnen Anforderungen an die Verifizierung für die verschiedenen Software-Engineeringprozesse sind in Abschnitt 5 angegeben.

5 Allgemeine Anforderungen

5.1 Einleitung

Dieser Abschnitt legt Anforderungen an das Engineering von Software für Raumfahrtssysteme fest. Diese gelten für alle Raumfahrtprojekte, für die Software erzeugt wird.

Jede Anforderung erhält eine Abschnittsnummer. Dem Text der Anforderung folgt gegebenenfalls eine weiterführende Erläuterung des Ziels. Im Abschnitt 'Erwartetes Ergebnis' wird für jede Anforderung das zugehörige Ergebnis angegeben. Für jedes Ergebniselement (z. B. „a“, „b“) wird/werden in Klammern als Akronym/e das/die Zieldokument/e, in das das Ergebnis einfließen soll, zusammen mit dem/den entsprechenden Review/s angegeben. So bedeuten zum Beispiel [DDF, DJF; QR] ein Ergebnis, das in die Designdefinitionsakte und die Designbegründungsakte Eingang finden soll und das Ergebnis für das Qualifikationsreview benötigt wird.

5.2 Systemengineering

5.2.1 Einleitung

Dieser Abschnitt beschreibt Aufgaben, die im Verantwortungsbereich des Kunden liegen. Dieser ist für die Lieferung eines Systems verantwortlich, in das die entwickelte Software integriert ist (siehe das in 4.3 beschriebene Modell der Kunde-Lieferant-Beziehung).

Es werden nur die Tätigkeiten des Kunden beschrieben, die zusätzliche Anforderungen für die Softwareentwicklung erfordern und zwar:

- Analyse der Systemanforderungen;
- Analyse der Systemstruktur;
- Anforderungen an Software-Verifizierung und -Validierung für die einzelnen Systemebenen;
- Integration der Software auf den einzelnen Systemebenen;
- Softwareanwendung.

Dokumentation auf Systemebene ist eine Vorbedingung für das anforderungsbezogene Engineering der Software. Die Anforderungen in diesem Abschnitt sollen die Vollständigkeit und Richtigkeit der Dokumentation des Kunden auf Systemebene sicherstellen und eine umfassende und verifizierte Grundlage für die Anforderungen an das Softwareprojekt liefern.

5.2.2 Analyse der Systemanforderungen

5.2.2.1 Diese Analyse umfasst:

- Spezifikation der Systemanforderungen;
- Risikoanalyse für System und Funktionen.

5.2.2.2 Die Systemanforderungen sind aus einer Analyse der für das zu entwickelnden System vorgesehenen Anwendung zu liefern. Alle Anforderungen müssen dokumentiert werden.

ERWARTETES ERGEBNIS:

- a) Anforderungen an Funktion und Leistung des Systems [RB; SRR];
- b) Anforderungen an Softwareanwendung und -pflege [RB; SRR];
- c) Anforderungen an Schnittstellen [IRD; SRR];
- d) Für das Design geltende Randbedingungen und Anforderungen an Verifizierung und Validierung [RB; SRR];
- e) Benennung anwendbarer Normen des Software-Engineering niedrigerer Ebenen [RB; SRR].

5.2.2.3 Für das System und die Funktionen ist eine Risikoanalyse durchzuführen.

ERWARTETES ERGEBNIS:

- a) Entwicklung umfassender Anforderungen an Sicherheit und Zuverlässigkeit der zu erzeugenden Software [RB; SRR];
- b) Ermittlung und Analyse kritischer Funktionen [RB; SRR].

5.2.3 Analyse der Systemstruktur

5.2.3.1 Im Rahmen des Systemdesignprozesses sind die Architektur und das Design (einschließlich Hardware, Software und Betriebshandbuch) des Systems zu beschreiben: Dies lässt sich als übergeordnete Analyse der Aufteilung des Systems bezeichnen. Das Systemdesign wird aus einer Analyse der Anforderungen an das System und an seine Funktionen abgeleitet. Die Erfüllung aller Anforderungen an das Systemdesign im Hinblick auf die Systemanforderungen muss überprüft werden. Alle Systemanforderungen sind den einzelnen Elementen des Systemdesigns zuzuordnen (und müssen zu diesen rückverfolgbar sein).

5.2.3.2 Es ist eine übergeordnete Analyse des Systems vorzunehmen. Diese Analyse muss alle Hardware- und Softwareeinheiten sowie das Betriebshandbuch umfassen. Dabei muss sichergestellt werden, dass den einzelnen Einheiten die Systemanforderungen zugeordnet werden. Daraus müssen sich später die Hardwarekonfiguration, die Softwarekonfiguration und das Betriebshandbuch entwickeln lassen. Die Analyse der Systemstruktur und die jeweils zugeordneten Systemanforderungen müssen dokumentiert werden.

ERWARTETES ERGEBNIS:

- a) Analyse der Systemstruktur mit Festlegung von Einheiten [RB; SRR];
- b) Anforderungen an Software- und Hardware-Schnittstellen [IRD; SRR];
- c) Liste der Einheiten der Systemkonfiguration [RB; SRR];
- d) Rückverfolgbarkeit der Systemstruktur [DJF; SRR].

5.2.4 Anforderungen an Software-Verifizierung und -Validierung auf Systemebene

5.2.4.1 Der allgemeine Ansatz für den Prozess der Verifizierung wird in ECSS-E-10A, Abschnitte 4 und 5, beschrieben, wo die mit der Verifizierung wie auch der Validierung zusammenhängenden Tätigkeiten behandelt werden.

ANMERKUNG 1 Es wird vorausgesetzt, dass bei allen Raumfahrtprojekten immer bestimmte Verifizierungs- und Validierungsarbeiten durchgeführt werden. Daher gelten die Festlegungen von ISO/IEC 12207, nach denen die Notwendigkeit von Validierung und Verifizierung zu ermitteln ist, hier nicht (Anpassung von ISO/IEC 12207).

ANMERKUNG 2 Die Prozessverifizierung durch den Lieferanten gilt als Teil des Managements und ist daher nicht Gegenstand des Software-Engineering (Anpassung von ISO/IEC 12207:1995, 6.4.2.2). Darüber hinaus legen die Normen für die Raumfahrt-Produktsicherung Anforderungen fest, die sich auf die Bewertung der Prozesse beim Lieferanten beziehen und deshalb in dieser Norm nicht wiederholt werden.

5.2.4.2 Der Kunde muss die Anforderungen an die Softwarequalifizierung nach 5.6 an die Systemanforderungen anpassen.

ZIEL: Festlegung der Anforderungen an den Verifizierungs- und Validierungsprozess des Kunden auf Systemebene und Vorbereitung der Softwareabnahme und -integration durch Übernahme der Anforderungen an die Verifizierung und Validierung in die Grundanforderungen.

ERWARTETES ERGEBNIS: Anforderungen an den Verifizierungs- und Validierungsprozess [RB; SRR].

5.2.4.3 Der Kunde muss die Grundanforderungen überprüfen.

Wenn das Produkt des Kunden ein Hardwareprodukt mit integrierter Software ist, muss es nach den Anforderungen der Normen ECSS-E-10 ausgeführt werden. Ist das Produkt des Kunden ein reines Softwareprodukt, so muss der Kunde diese Norm in seiner Eigenschaft als Lieferant für eine höhere Ebene im Produktbaum anwenden.

ERWARTETES ERGEBNIS: Begründung der Anforderungen [Kunden-DJF auf Systemebene].

ANMERKUNG Dieses Ergebnis ist ein Sonderfall: Das Ergebnis steht nicht im Zusammenhang mit der Schnittstelle 'Kunde-Lieferant' und stellt deshalb keinen festzulegenden Meilenstein des Software-Engineeringprozesses dar. Stattdessen fließt das Ergebnis in die kundeneigene System-DJF ein und sollte vom Kunden nur in seiner Eigenschaft als Lieferant auf der nächsthöheren Ebene im Produktbaum verwendet werden. Das Ergebnis wird hier nur der Vollständigkeit halber aufgeführt.

5.2.5 Softwareintegration auf Systemebene

5.2.5.1 Die Softwareintegration umfasst folgende Aufgaben, die der Kunde in seiner Eigenschaft als Lieferant innerhalb des Gesamtsystems im Rahmen seiner Tätigkeit auf einzelnen Systemebenen ausführen oder unterstützen muss:

- Ermittlung des Grads der Überprüfbarkeit der Software zur Erleichterung der Softwareintegration;
- Festlegen der erforderlichen Steuer- und Datenschnittstellen für die Integration auf Systemebene;
- Festlegen von Anforderungen an die Datenträger für die Integration;
- Festlegen funktioneller Anforderungen zur Unterstützung der Softwareintegration;
- Festlegen von Vorgaben auf Systemebene, die zur Vorbereitung der Softwareintegration durch den Kunden erforderlich sind;
- Ermitteln der für das Software-Engineering erforderlichen Ergebnisse durch den Lieferanten zur Vorbereitung der Softwareintegration auf Systemebene;
- Festlegen des Lieferanten für die Softwareintegration auf Systemebene.

Die Integration von Softwareprodukten auf Systemebene findet erst im Anschluss an das CDR des Softwareprodukts statt, während das Engineering, das Design und die Planung für das Produkt zur späteren Integration auf Systemebene schon vor dem CDR abgeschlossen sind.

5.2.5.2 Wird ein Softwareprodukt in ein System integriert, müssen alle für die Softwareintegration erforderlichen Anforderungen hinsichtlich der Überprüfbarkeit der Software vom Kunden festgelegt werden.

ERWARTETES ERGEBNIS: Anforderungen an die Überprüfbarkeit von Software [RB; SRR].

5.2.5.3 Ist die Software Teil eines Systems, müssen alle Schnittstellen zwischen der Software und dem System vom Kunden festgelegt werden, in statischer wie dynamischer Hinsicht, für den Nennzustand wie auch den Zustand verminderter Leistungsfähigkeit (z. B. Verhalten bei einem Fehler).

Externe Schnittstellen für systemintegrierte Software können sein:

- Schnittstellen zu anderer Software desselben Systems (z. B. Betriebssystem, Dateien, Datenbank-Managementsystem oder andere Anwendungen);
- Schnittstellen zu bestimmten Hardwarekonfigurationen;
- Schnittstellen für die Datenübermittlung (z. B. zum Erstellen bestimmter Netzwerkprotokolle).

ERWARTETES ERGEBNIS: Anforderungen an Schnittstellen auf Systemebene [IRD; SRR].

5.2.5.4 Der Kunde muss das Medium für die Schnittstellendaten angeben und die Anforderungen entsprechend formulieren.

Beispielsweise können die Schnittstellendaten so definiert und strukturiert werden, dass sie automatisch in die Umgebung für die Softwareentwicklung des Lieferanten einfließen. Zwischen Faktoren wie Menge der Softwarepakete im System, der Möglichkeit der Weiterentwicklung der Schnittstellen und der Anzahl der Schnittstellen-Datensätze muss ein Kompromiss gefunden werden.

ERWARTETES ERGEBNIS: Datenschnittstellen auf Systemebene [IRD; SRR].

5.2.5.5 Der Kunde muss gegebenenfalls besondere Randbedingungen für den Lieferanten für die Unterstützung der Softwareintegration in das System festlegen.

Für die Integration von Software sollten einige Randbedingungen vorgegeben werden, z. B.:

- Spezifikation des zu verwendenden Betriebssystems;
- Spezifikation der einzusetzenden handelsüblichen (gebrauchsfertigen) Software (z. B. Datenbank, MMI-Generator);
- Spezifikation der anzuwendenden Umgebung für die Softwareentwicklung.

ERWARTETES ERGEBNIS: Randbedingungen für die Softwareentwicklung [RB; SRR].

5.2.5.6 Gegebenenfalls hat der Kunde dem Lieferanten bestimmte Vorgaben zur Integration der Software in das System zu machen und die entsprechenden Grundanforderungen zu formulieren.

Für die Softwareintegration kann der Kunde dem Lieferanten bestimmte Vorgaben zur Validierung der Software in einer Umgebung machen, die für den späteren Einsatz repräsentativ ist. Diese können sein:

- Versuchsaufbau oder Computermodell;
- Simulator für die Hardware- oder Softwareumgebung.

ERWARTETES ERGEBNIS: Produkte zur Unterstützung der Softwareintegration auf Systemebene [IRD; SRR].

5.2.5.7 Der Kunde muss die vom Lieferanten zu liefernden Ergebnisse zur Integration der Software in das System festlegen und die entsprechenden Grundanforderungen formulieren.

Für die Softwareintegration kann der Kunde Prototypversionen oder vorläufige Versionen fordern. Funktionalität und Lieferdaten für diese Versionen sind festzulegen.

ERWARTETES ERGEBNIS: Anforderungen an die Vorbereitung der Softwareintegration auf Systemebene [IRD; SRR].

5.2.5.8 Der Kunde muss sich auf die Unterstützung des Softwarelieferanten für die Softwareintegration auf der Systemebene einstellen und entsprechende Grundanforderungen festlegen.

Entsprechende Maßnahmen können sich auf Schulung, Softwarepflege, Konfiguration und die Unterstützung bei der Durchführung von Tests beziehen.

ERWARTETES ERGEBNIS: Anforderungen zur Erleichterung der Softwareintegration auf Systemebene [IRD; SRR].

5.2.6 Software unter Betriebsbedingungen

5.2.6.1 Da die Software Bestandteil des Raumfahrtssystems ist, richtet sich die Terminplanung und das Management bei der Anwendung der Software nach den Anforderungen an das Gesamtsystem, die auch für Softwareprodukte gelten.

5.2.6.2 Der Kunde muss dem Lieferanten für alle Softwareprodukte den späteren Nutzer benennen (nachfolgend als „Betreiber“ bezeichnet).

5.2.6.3 Der Kunde muss Systemanforderungen für den Einsatz der Softwareprodukte festlegen. Der Betreiber hat entsprechende Anforderungen für den betrieblichen Einsatz zu formulieren und diese Antwort entsprechend 5.7 zu dokumentieren. Die Antwort des Betreibers ist mit dem Kunden in einem Betriebsanforderungsreview abzustimmen, mit dem die Umsetzung der Pläne in die Praxis freigegeben wird.

5.3 Software-Management

5.3.1 Einleitung

Die meisten der für das Management und die Steuerung von Softwareprojekten für Raumfahrtsysteme geltenden Anforderungen sind in der Normenreihe für das Raumfahrt-Projektmanagement zu finden. Sie werden hier deshalb nicht wiederholt. Darüber hinaus findet auch ECSS-Q-80 mit Anforderungen an die Softwareproduktsicherung für die Steuerung von Softwareprojekten Anwendung. Die in diesem Abschnitt festgelegten Anforderungen berücksichtigen das Engineering und die Steuerung der Softwareentwicklung bei Raumfahrtprojekten und sind somit als Bindeglied zwischen den oben erwähnten Normen für die Raumfahrt und den Tätigkeiten im Software-Engineering zu sehen.

Das Management und die Steuerung, die Gegenstand dieses Abschnitts sind, zielen auf:

- den Software-Lebenszyklus;
- das Schnittstellen-Management;
- das Management der finanziellen und technischen Rahmenbedingungen.

Die Anforderungen in diesem Abschnitt gelten für jede Softwareart bei Raumfahrtprojekten.

Wie genauer in den folgenden Abschnitten beschrieben, muss die Software bei bestimmten Meilensteinen mindestens den Softwarereviews SRR, PDR, CDR, QR und AR unterzogen werden. Weitere Reviews (z. B. Review des Projektplans vor dem PDR) dürfen vom Kunden gefordert werden und sollten die in 5.6.8 festgestellten Anforderungen erfüllen.

5.3.2 Software-Lebenszyklus

5.3.2.1 Um eine effektive Planung im Allgemeinen sowie Zeitplanung im Besonderen sicherzustellen, ist der Lebenszyklus für die Softwareentwicklung in Phasen zu unterteilen, von denen jede ihre Meilensteine besitzt.

5.3.2.2 Der Softwarelieferant muss einen Lebenszyklus für die Softwareentwicklung nach 4.3 festlegen und einhalten, der alle Tätigkeiten von der Aufstellung der Anforderungen bis zur Inbetriebnahme der Software umfasst. Bei der Festlegung des Lebenszyklus müssen bestimmte Methoden der Softwareentwicklung (z. B. Verwendung eines Datenbank-Managementsystems, Wiederverwendung der Produkte) eingesetzt, und projektspezifische Risiken (z. B. häufig wechselnde Spezifikationen, enge Zeitvorgaben) sowie Fragen der Abstimmung mit der übergeordneten Ebene beachtet werden.

Die Wahl des Software-Lebenszyklus muss auf die Anforderungen an das Gesamtprojekt abgestimmt sein, wobei das in 4.3 und EN 13290-4:2001 beschriebene Prozessmodell zu verwenden ist.

ERWARTETES ERGEBNIS: Festlegung des Lebenszyklus für die Softwareentwicklung eines Projekts als Teil des Software-Entwicklungsplans [TS; PDR].

5.3.2.3 Der Lebenszyklus für die Softwareentwicklung muss für jede Phase und die zugehörigen Meilensteine sowohl die Vorgaben wie auch die Ergebnisse festlegen.

Das Ergebnis für jede Phase kann vollständig oder in Zusammenfassung erfasst werden und muss das Ergebnis der Verifizierung der Ergebnisse jeder Phase einschließen.

Meilensteine sind die vom Kunden geforderten gemeinsamen technischen Reviews (SRR, PDR, CDR, QR und AR) sowie die Reviews auf Lieferantenebene.

Die Ergebnisse der Reviews sind Dokumente, die einer Prüfung unterzogen und bei der Festlegung des Software-Lebenszyklus gesondert aufgeführt werden.

5.3.2.4 Für den Software-Lebenszyklus muss die Schnittstelle zwischen Entwicklung und Softwarepflege (z. B. zu erstellende Dokumente, zu wartende Werkzeuge) bestimmt werden.

ZIEL: Festlegung des erforderlichen Inputs für die Softwarepflege während der Softwareentwicklung. Siehe 5.8.

ERWARTETES ERGEBNIS: Elemente des Softwarepflegeplans [TS; PDR].

5.3.2.5 Das Material, das dem SRR unterzogen wird, muss die vom Kunden freigegebenen Grundanforderungen an die Software umfassen.

Diese Anforderungen ergeben sich aus der Analyse der Systemanforderungen und einer vom Kunden durchgeführten Systemanalyse. Sie stellen die Anforderungen des Kunden an die zu entwickelnde Software dar, d. h.:

- Kundenanforderungen;
- Anforderungen an die äußeren Schnittstellen der Software.

ERWARTETES ERGEBNIS:

- a) Abnahme der Grundanforderungen durch den Kunden [RB; SRR];
- b) Bericht über das SRR-Meilensteinreview [DJF; SRR].

5.3.2.6 Am Anfang des Lebenszyklus für die Softwareentwicklung muss eine Phase stehen, in der die technische Spezifikation für die Software erstellt wird.

ZIEL: Aufstellung der technischen Grundspezifikation für das Projekt als Reaktion des Softwarelieferanten auf die Grundanforderungen für das System. Die Spezifikation umfasst alle an das Softwareprodukt zu stellende technischen Anforderungen und soll in einer frühen Phase des Projektablaufs stehen.

ERWARTETES ERGEBNIS:

- a) Technische Spezifikation der Software [TS; PDR];
- b) Design der Softwarearchitektur auf übergeordneter Ebene [DDF; PDR];
- c) Schnittstellenkontrolldokument [ICD; PDR];
- d) Festlegen von Kompromissen für das Design auf übergeordneter Ebene [DJF; PDR].

5.3.2.7 Als Abschluss der Spezifikationsphase muss der Softwarelieferant ein vorläufiges Design-Review (PDR) abhalten, zu dem der Kunde einzuladen ist.

ZIEL:

- Übereinstimmung mit dem Kunden oder seinen Vertretern, dass alle den Systemanforderungen entsprechenden Grundanforderungen in der technischen Spezifikation enthalten sind.
- Review der Softwarearchitektur auf übergeordneter Ebene.

ERWARTETES ERGEBNIS: Abnahme der technischen Spezifikation und Softwarearchitektur auf übergeordneter Ebene durch den Kunden [TS, DDF, ICD, DJF; PDR].

5.3.2.8 Am Ende der Designphase muss der Softwarelieferant ein kritisches Design-Review (CDR) abhalten, zu dem der Kunde einzuladen ist.

ZIEL: Während des CDR werden die Designdefinitionsakte, das Betriebshandbuch und die zugehörige Designbegründungsakte überprüft.

Die Vollständigkeit des Verifizierungs- und Validierungsplans und die Verfügbarkeit der notwendigen Mittel (z. B. Testspezifikation, Simulatoren) müssen überprüft werden.

ERWARTETES ERGEBNIS:

- a) Abnahme der Designdefinitionsakte (z. B. Architekturdesign und detailliertes Design, Code) durch den Kunden [DDF; CDR];
- b) Abnahme der Designbegründungsakte (z. B. Ergebnisse von Einzel- und Integrationstests) durch den Kunden [DJF; CDR];
- c) Abnahme des Designs der Schnittstellen und des Integrationsplans auf Systemebene durch den Kunden [DDF, DJF; CDR];
- d) Abnahme des Betriebshandbuchs durch den Kunden [TS; CDR].

5.3.2.9 Um sicherzustellen, dass das Softwareprodukt mit der technischen Spezifikation übereinstimmt, müssen am Ende des Lebenszyklus die Ergebnisse überprüft (verifiziert) und validiert werden.

ZIEL: Durch Verifizierung und Validierung in einer für Betriebsbedingungen typischen Umgebung sicherstellen, dass das Softwareprodukt vor der Integration in das System der technischen Spezifikation entspricht.

5.3.2.10 Der Softwarelieferant muss zum Nachweis, dass das Softwareprodukt alle festgelegten Anforderungen erfüllt, ein Qualifikationsreview (QR) durchführen.

ZIEL: Nachweis, dass die Software alle festgelegten Anforderungen erfüllt, und insbesondere, dass die Ergebnisse der Verifizierung und Validierung erlauben, den Softwareprodukten den „qualifizierten Zustand“ zuzuordnen.

Während des QR werden eine Zusammenfassung von Testberichten und das Betriebshandbuch überprüft, ebenso wie die Einheitlichkeit der gesamten Softwaredokumentation (TS, DDF, ICD, DJF, Betriebshandbuch).

ERWARTETES ERGEBNIS: Abnahme des „qualifizierten Zustands“ durch den Kunden [DJF; QR].

5.3.2.11 Nach dem Qualifikationsreview muss der Kunde ein Abnahmereview abhalten.

ZIEL: Abnahme der Software unter Berücksichtigung der vorgesehenen Betriebsumgebung.

ERWARTETES ERGEBNIS: Billigung des abgenommenen Zustands durch den Kunden [DJF; AR].

5.3.2.12 Jeder Lieferant muss die Software-Engineering-Normen festlegen, die er in seinen Bereichen anwenden wird. Diese Normen müssen vom Kunden als für die zu entwickelnde Anwendung geeignet akzeptiert werden.

ZIEL: Bestimmung der für das Projekt anwendbaren Software-Engineering-Normen.

ERWARTETES ERGEBNIS:

- a) Dokumentationsnormen [RB; SRR];
- b) Designnormen [RB; SRR];
- c) Normen für Verifizierung und Validierung [RB; SRR].

5.3.3 Schnittstellen-Management

Schnittstellen müssen bei Festlegung der Grundanforderungen in einem Dokument definiert werden, in dem die auf verschiedene Elemente des Produktbaums des Systems anwendbaren Anforderungen festgelegt sind.

Die Verfahren im Schnittstellen-Management müssen den Anforderungen nach EN 13290-5:2001 entsprechen.

ZIEL: Festlegung von Verfahren, die die Einheitlichkeit der Systemschnittstellen sicherstellen.

ERWARTETES ERGEBNIS:

- a) Schnittstellen-Managementverfahren [RB; SRR];
- b) Teil des Konfigurationsmanagementplans [RB; SRR].

5.3.4 Management der technischen und finanziellen Rahmenbedingungen

5.3.4.1 Die in diesem Abschnitt beschriebenen Rahmenbedingungen für die Software stehen mit der Hardwarekonfiguration (Kapazität von Arbeitsspeicher und Festspeicher) und den Leistungsanforderungen in Zusammenhang.

5.3.4.2 Die für die Software zu beachtenden Rahmenbedingungen und Reservephilosophie müssen vom Kunden in den Grundanforderungen festgelegt werden.

ZIEL: Festlegung der Rahmenbedingungen, die vom Lieferanten zu beachten sind.

ERWARTETES ERGEBNIS: Rahmenbedingungen und Reservenphilosophie für das Projekt [RB; SRR].

5.3.4.3 Der Lieferant muss die Rahmenbedingungen in finanzieller und technischer Hinsicht regeln und deren Status bei jedem Meilenstein angeben.

Die Bedingungen müssen durch Analyse in den frühen Entwicklungsphasen des Projekts festgelegt und durch Prüfung der Software im Zuge der Softwareimplementierung bestätigt werden.

Hypothesen, auf denen die Analyse fußt, müssen bei Angabe der Ergebnisse beschrieben werden.

ERWARTETES ERGEBNIS: Status der finanziellen und technischen Rahmenbedingungen [DJF; PDR, CDR, QR, AR].

5.4 Anforderungsbezogenes Engineering

5.4.1 Einleitung

Der Engineeringprozess zum Festlegen der Anforderungen an die Software umfasst:

- Analyse der Anforderungen an die Software;
- Design der Softwarearchitektur auf übergeordneter Ebene;
- Software-Verifizierung und -Validierung.

5.4.2 Analyse der Softwareanforderungen

5.4.2.1 Die Analyse für jede Softwareeinheit umfasst:

- Aufstellen und Dokumentieren der Anforderungen;
- Kennzeichnen der Anforderungen;
- Bewerten der Anforderungen.

5.4.2.2 Der Lieferant muss Anforderungen einschließlich derjenigen an die Softwarequalität aufstellen und dokumentieren.

ERWARTETES ERGEBNIS: (technische Grundanforderungen der Spezifikation)

- a) Spezifikationen von Funktion und Leistung einschließlich von Hardwarekennwerten sowie Umgebungsbedingungen, unter denen die Softwareeinheit arbeiten wird [TS; PDR];
- b) externe Schnittstellen zur Softwareeinheit [ICD; PDR];
- c) Verifizierungsplan (unter Bezug auf 5.6.4.6) [DJF; PDR];
- d) Spezifikationen zur Sicherheit, auch in Bezug auf Softwareanwendung und -pflege, Umwelteinflüsse und mögliche Verletzungsgefahr für Bedienungspersonal [TS; PDR];
- e) Spezifikationen zur Datensicherheit, auch in Bezug auf Faktoren, die sensitive Daten gefährden könnten [TS; PDR];
- f) Spezifikationen zur Arbeitsphysiologie (Ergonomie), auch in Bezug auf Faktoren wie Handbetrieb, Schnittstelle zwischen Maschine-Mensch, Einschränkungen für das Personal und Bereiche, in denen eine erhöhte Aufmerksamkeit erforderlich ist und die anfällig für menschliche Fehler sind und Schulung erfordern [TS; PDR];
- g) Datendefinitionen und Anforderungen an Datenbanken [TS; PDR];
- h) Anforderungen an Installation und Abnahme der gelieferten Software am/an Ort/den Orten von Softwareanwendung und -pflege [TS; PDR];
- i) Benennung von Software-Engineering-Normen niedrigerer Ebenen, die angewendet werden und mit denen sich die Übereinstimmung der Software mit dieser Norm dokumentieren lässt [TS; PDR].

5.4.2.3 Jede Anforderung ist im Hinblick auf ihre Rückverfolgbarkeit einzeln zu kennzeichnen.

5.4.2.4 Der Lieferant muss die Anforderungen unter Berücksichtigung der folgenden Kriterien im Rahmen der „Verifizierung der Softwareanforderungen“ (5.6.6.2) bewerten. Die Ergebnisse sind zu dokumentieren.

- a) Rückverfolgbarkeit der Systemstruktur;
- b) Übereinstimmung mit den Systemanforderungen;
- c) interne Übereinstimmung;
- d) Verifizierbarkeit;
- e) Realisierbarkeit des Softwaredesigns;
- f) Realisierbarkeit von Softwareanwendung und -pflege.

ERWARTETES ERGEBNIS:

- a) Matrix für die Rückverfolgbarkeit der Anforderungen [DJF; PDR];
- b) Bericht über die Verifizierung der Anforderungen [DJF; PDR].

5.4.3 Design der Softwarearchitektur auf übergeordneter Ebene

5.4.3.1 Für jede Softwareeinheit umfasst das Design:

- Strukturierung der Softwareanforderungen;
- Entwicklung und Dokumentation des Designs der (externen und internen) Softwareschnittstellen auf übergeordneter Ebene;
- Entwicklung und Dokumentation von vorläufigen Versionen des Betriebshandbuchs;

- Entwicklung und Dokumentation von vorläufigen Testanforderungen und eines Plans zur Softwareintegration;
- Bewertung der Softwarearchitektur auf übergeordneter Ebene;
- Durchführung eines vorläufigen Designreviews.

5.4.3.2 Der Lieferant muss die Anforderungen an die Softwareeinheit so strukturieren, dass sich die Struktur der Software auf übergeordneter Ebene beschreiben lässt und Softwarekomponenten identifizieren lassen. Es muss sichergestellt sein, dass alle Anforderungen an die Softwareeinheit den Softwarekomponenten zugeordnet und dann weiter aufgeschlüsselt werden, um ein verfeinertes Design zu ermöglichen. Die Architektur der Softwareeinheit auf übergeordneter Ebene ist zu dokumentieren.

ERWARTETES ERGEBNIS:

- a) Softwarearchitekturdesign [DDF; PDR];
- b) Design der Architektur auf übergeordneter Ebene für Matrix zur Rückverfolgbarkeit von Anforderungen [DJF; PDR].

5.4.3.3 Der Lieferant muss ein Design auf übergeordneter Ebene für die externen Schnittstellen zur Softwareeinheit und zwischen deren Softwarekomponenten entwickeln und dokumentieren.

ERWARTETES ERGEBNIS:

- a) Vorläufiges Design für externe Schnittstellen (auf übergeordneter Ebene) [ICD; PDR];
- b) vorläufiges Design für interne Schnittstellen (auf übergeordneter Ebene) [ICD; PDR].

5.4.3.4 Der Lieferant muss vorläufige Versionen des Betriebshandbuchs entwickeln und dokumentieren.

ERWARTETES ERGEBNIS: Vorläufige Version des Betriebshandbuchs [TS; PDR].

5.4.3.5 Der Lieferant muss vorläufige Testanforderungen und einen vorläufigen Plan für die Softwareintegration entwickeln und dokumentieren.

ERWARTETES ERGEBNIS: (Vorläufiger) Plan zur Softwareintegration [DJF; CDR].

5.4.3.6 Der Lieferant muss die Architektur der Softwareeinheit und das Schnittstellendesign unter Berücksichtigung der folgenden Kriterien bewerten. Die Ergebnisse der Bewertung sind zu dokumentieren. Die Verifizierung des Architekturdesigns muss im Rahmen der „Verifizierung des Softwaredesigns“ (5.6.6.3) erfolgen.

- a) Rückverfolgbarkeit der Anforderungen an die Software zur Softwareeinheit
- b) Stimmigkeit mit den Anforderungen für die Softwareeinheit
- c) Stimmigkeit der Softwarekomponenten untereinander
- d) Eignung der anwendbaren Designmethoden und -normen
- e) Realisierbarkeit des verfeinerten Designs
- f) Realisierbarkeit von Softwareanwendung und -pflege

ERWARTETES ERGEBNIS: Bericht über die Verifizierung von Architektur und Schnittstellen [DJF; PDR].

5.4.3.7 Der Lieferant muss ein vorläufiges Design-Review (PDR) nach 5.3.2.7 durchführen. Ein erfolgreiches Review bildet die Grundlage für die Entwicklung der Softwareeinheit.

ERWARTETES ERGEBNIS: PDR-Meilensteinbericht [DJF; PDR].

5.4.4 Software-Verifizierung und -Validierung

Die technischen Spezifikationen müssen Festlegungen zur Verifizierung und Validierung des Softwareprodukts enthalten. Sie basieren auf Grundanforderungen des Kunden (5.2.4.2) für Verifizierung und Validierung.

Die entsprechenden Prozesse sind:

- a) Verifizierungsprozess (5.6.4) und
- b) Validierungsprozess (5.6.5).

ERWARTETES ERGEBNIS:

- a) Plan zur Software-Verifizierung — Risiken und Leistung [DJF; PDR];
- b) Plan zur Software-Verifizierung — Methoden und Werkzeuge [DJF; PDR];
- c) Plan zur Software-Verifizierung — Organisation [TS; PDR];
- d) Plan zur Software-Validierung — Leistung und Unabhängigkeit [DJF; PDR];
- e) Plan zur Software-Validierung — Methoden und Werkzeuge [DJF; PDR];
- f) Plan zur Software-Validierung — Unabhängige Validierung [DJF; PDR];
- g) Plan zur Software-Validierung — Organisation [TS; PDR].

5.5 Design-Engineering

5.5.1 Einleitung

Der Engineeringprozess für das Softwaredesign umfasst:

- Design der Softwareeinheiten;
- Codierung und Test;
- Softwareintegration.

5.5.2 Design von Softwareeinheiten

5.5.2.1 Für jede Softwareeinheit umfasst das Engineering:

- Design der Softwarekomponenten;
- Entwicklung und Dokumentation des Schnittstellendesigns;
- Aktualisierung des Betriebshandbuchs;
- Definition und Dokumentation einer Testspezifikation und eines Testplans für jede Softwareeinheit;
- Aktualisierung der Testspezifikation und des Zeitplans für die Integration;
- Bewertung des verfeinerten Softwaredesigns und der Testspezifikation.

5.5.2.2 Der Lieferant muss ein verfeinertes Design für jede Softwarekomponente entwickeln. Jede Softwarekomponente muss in Softwareeinheiten aufgeschlüsselt werden, die codiert, kompiliert und getestet werden können. Dabei muss sichergestellt werden, dass alle Anforderungen an die Software von den Komponenten den Softwareeinheiten zugeordnet werden. Das Design muss dokumentiert werden.

ERWARTETES ERGEBNIS: Dokumente für das Design der Softwarekomponenten [DDF; CDR].

5.5.2.3 Der Lieferant muss ein verfeinertes Design für die externen Schnittstellen der Softwareeinheiten, der Schnittstellen zwischen den Komponenten sowie zwischen den Softwareeinheiten entwickeln und dokumentieren. Das Schnittstellendesign muss eine Codierung ohne Rückgriff auf weitere Informationen erlauben.

ERWARTETES ERGEBNIS:

- a) Kontrolldokumente für externe Schnittstellen (Aktualisierung) [ICD; CDR];
- b) Kontrolldokumente für interne Schnittstellen (Aktualisierung) [ICD; CDR].

5.5.2.4 Der Lieferant muss das Betriebshandbuch soweit erforderlich aktualisieren.

ERWARTETES ERGEBNIS: Betriebshandbuch (Aktualisierung) [TS; CDR].

5.5.2.5 Der Lieferant muss die Testanforderungen und den Plan für das Testen der Softwareeinheiten festlegen und dokumentieren. Die Testspezifikationen müssen die Beanspruchung der Software bis zu den Anforderungsgrenzen einschließen.

ERWARTETES ERGEBNIS: Testplan für Softwareeinheiten [DJF; CDR].

5.5.2.6 Der Lieferant muss die Testanforderungen und den Plan für die Softwareintegration aktualisieren.

ERWARTETES ERGEBNIS: Testplan für die Softwareintegration [DJF; CDR].

5.5.2.7 Der Lieferant muss das Softwaredesign und die Testspezifikation unter Berücksichtigung der folgenden Kriterien im Rahmen der „Verifizierung des Softwaredesigns“ (5.6.6.3.) bewerten. Die Ergebnisse der Bewertung sind zu dokumentieren.

- a) Rückverfolgbarkeit des Architekturdesigns der Softwareeinheit;
- b) Übereinstimmung des Softwaredesigns mit dem Architekturdesign;
- c) Übereinstimmung zwischen Softwarekomponenten und Softwareelementen;
- d) Eignung der verwendeten Testverfahren und Normen zur Softwareintegration;
- e) Durchführbarkeit der Tests;
- f) Realisierbarkeit von Softwareanwendung und -pflege.

ERWARTETES ERGEBNIS:

- a) Design-Verifizierungsbericht [DJF; CDR];
- b) Matrix für die Rückverfolgbarkeit des Designs [DJF; CDR].

5.5.3 Codierung und Testen

5.5.3.1 Für jede Softwareeinheit umfasst das Codieren und Testen:

- Entwicklung und Dokumentation von Softwareelementen, Testverfahren und Testdaten;
- Testen jedes Softwareelements und jeder Datenbank;
- Aktualisierung des Betriebshandbuchs;
- Aktualisierung der Testanforderungen und des Zeitplans für die Integration;
- Bewertung des Softwarecodes und der Testergebnisse.

5.5.3.2 Der Lieferant muss Folgendes entwickeln und dokumentieren:

- a) die Codierung jedes Softwareelements;
- b) Testverfahren und Daten für das Testen jedes Softwareelements.

ERWARTETES ERGEBNIS:

- a) Dokumente und Code für das Design der Softwarekomponenten (Aktualisierung) [DDF; CDR];
- b) Testplan für Softwareeinheiten (Aktualisierung) [DJF; CDR].

5.5.3.3 Der Lieferant muss im Rahmen der „Verifizierung des Softwarecodes“ (5.6.6.4) jedes Softwareelement testen und feststellen, ob es seine Anforderungen erfüllt. Die Testergebnisse sind zu dokumentieren.

ERWARTETES ERGEBNIS:

- a) Dokument und Code für das Design der Softwarekomponenten (Aktualisierung) [DDF; CDR];
- b) Berichte über Einzeltests [DJF; CDR].

5.5.3.4 Der Lieferant muss das Betriebshandbuch soweit erforderlich aktualisieren.

ERWARTETES ERGEBNIS: Betriebshandbuch (Aktualisierung) [TS; CDR].

5.5.3.5 Der Lieferant muss die Testanforderungen und den Plan für die Softwareintegration aktualisieren.

ZIEL: Abstimmung der Testanforderungen und des Integrationsplans auf die Ergebnisse des Codedesignprozesses.

ERWARTETES ERGEBNIS: Softwareintegrationsplan (Aktualisierung) [DJF; CDR].

5.5.3.6 Der Lieferant muss den Softwarecode und die Testergebnisse im Rahmen der „Verifizierung des Softwarecodes“ (5.6.6.4) anhand der folgenden Kriterien bewerten. Die Ergebnisse der Bewertung sind zu dokumentieren.

- a) Rückverfolgbarkeit zwischen dem Design der Softwareeinheit und dem Code der Softwareelemente;
- b) Übereinstimmung mit den Anforderungen und dem Design der Softwareeinheit;
- c) Stimmigkeit der Einzelanforderungen untereinander;
- d) Angaben über den Umfang der Tests an allen Softwareeinheiten;
- e) Eignung im Hinblick auf die einschlägigen Codierv Verfahren und Normen;
- f) Durchführbarkeit der Softwareintegration und Tests;
- g) Realisierbarkeit von Softwareanwendung und -pflege.

ERWARTETES ERGEBNIS:

- a) Softwarecode-Verifizierungsbericht [DJF; CDR];
- b) Matrix für die Rückverfolgbarkeit des Softwarecodes [DJF; CDR].

5.5.4 Integration

5.5.4.1 Die folgenden Abschnitte gelten für die Integration des Softwareprodukts, wie es vom Lieferanten zum Kunden geliefert wird. Im Rahmen des Integrationsprozesses muss der Validierungstest für das integrierte Produkt vorbereitet werden.

Die Integration des Softwareprodukts steht in Beziehung zur Integration der Software in das Raumfahrtssystem. Auf der nächsthöheren Ebene im Produktbaum findet die Integration auf Systemebene statt und zwar üblicherweise nach Abschluss des Qualifikationsreviews. Abhängig vom Lebenszyklus auf Systemebene und von der angenommenen Risikoabgrenzung zwischen Lieferant und Kunden darf die Integration jedoch früher durchgeführt werden (siehe 5.2.5.7), aber nicht früher als das kritische Designreview für die Software.

Für jede Softwareeinheit umfasst die Integration:

- Entwicklung eines Integrationsplans;
- Integration und Test der Softwareelemente und -komponenten;
- Aktualisierung des Betriebshandbuchs;
- Bewertung von Integrationsplan, Design, Code, Testverfahren, Testergebnissen und Betriebshandbuch;
- Durchführung eines kritischen Designreviews (CDR) vor Beginn der Validierung.

5.5.4.2 Der Lieferant muss einen Integrationsplan für die Integration der Softwareelemente und -komponenten in die Softwareeinheit entwickeln. Der Plan muss Testanforderungen, Verfahren, Daten, Zuständigkeiten und einen Zeitplan enthalten. Er ist zu dokumentieren.

ERWARTETES ERGEBNIS: Testplan für die Softwareintegration [DJF; CDR].

5.5.4.3 Der Lieferant muss die Softwareelemente und -komponenten und den Test nach Entwicklung der Datengruppen gemäß Integrationsplan integrieren. Dabei muss sichergestellt werden, dass jede Datengruppe die Anforderungen für die Softwareeinheit erfüllt und dass die Einheit nach Abschluss aller mit der Integration zusammenhängenden Arbeiten integriert ist. Die Integration und die Testergebnisse sind zu dokumentieren.

ERWARTETES ERGEBNIS: Testbericht über die Softwareintegration [DJF; CDR].

5.5.4.4 Der Lieferant muss das Betriebshandbuch soweit erforderlich aktualisieren.

ERWARTETES ERGEBNIS: Betriebshandbuch (Aktualisierung) [TS; QR].

5.5.4.5 Der Lieferant muss den Integrationsplan, das Design, den Code, die Testverfahren, die Testergebnisse und das Betriebshandbuch unter Berücksichtigung nachfolgender Kriterien bewerten. Die Ergebnisse der Bewertung sind zu dokumentieren.

Die „Verifizierung der Softwareintegration“ (5.6.6.5.) soll sich beziehen auf:

- a) Rückverfolgbarkeit zu den Systemanforderungen;
- b) Übereinstimmung mit den Systemanforderungen;
- c) interne Übereinstimmung;
- d) Angabe, inwieweit die Anforderungen für die Softwareeinheit getestet werden können;
- e) Eignung der verwendeten Testnormen und -methoden;
- f) Übereinstimmung mit den erwarteten Ergebnissen;
- g) Durchführbarkeit des Software-Validierungstests;
- h) Realisierbarkeit von Softwareanwendung und -pflege.

ERWARTETES ERGEBNIS: Verifizierungsbericht über die Softwareintegration, der die Berichte unter a), b), c) und d) einschließt:

- a) Verifizierungsbericht über die Softwareintegration [DJF; CDR];
- b) Verifizierungsbericht über die Softwaredokumentation [DJF; CDR];

- c) Bewertung der Vollständigkeit des Tests und Einhaltung des Codes [DJF; CDR];
- d) Bestätigung der Durchführbarkeit für Validierungstests, Softwareanwendung und -pflege [DJF; CDR];
- e) Testspezifikation für die Software-Validierung [DJF; CDR].

5.5.4.6 Der Lieferant muss ein kritisches Design-Review (CDR) nach 5.3.2.8 durchführen. Alle dafür erforderlichen Ergebnisse müssen im Rahmen der „Verifizierung der Softwaredokumentation“ (5.6.6.6) in Vorbereitung auf das CDR aufbereitet und geprüft werden.

ZIEL: Erarbeitung der Grundlagen der Designdokumentation vom Lieferanten für das Projekt für den Übergang vom „spezifizierten Zustand“ in den „definierten Zustand“ und durch Erreichen des Meilensteins „Design ist abgeschlossen“.

ERWARTETES ERGEBNIS: CDR-Meilensteinbericht [DJF; CDR].

5.6 Software-Verifizierung und -Validierung (-Qualifizierung)

5.6.1 Einleitung

Bei Verifizierungs- und Validierungsprozessen gibt es unterschiedliche Grade der Unabhängigkeit der daran beteiligten Personen. Es kann sich um ein und dieselbe Person, eine andere Person derselben Organisation oder um Dritte handeln. In Fällen, in denen die Prozesse von einer Stelle durchgeführt werden, die vom Lieferanten unabhängig ist, wird von einer unabhängigen Software-Verifizierung und -Validierung gesprochen bzw. von einer unabhängigen Software-Validierung, wenn die Unabhängigkeit nur für die Validierung gilt.

Die Anforderungen dieses Abschnitts sind unterteilt; der erste Teil betrifft Engineeringtätigkeiten für die QR- und AR-Meilensteine (5.6.2), der zweite Teil (ab 5.6.3) enthält Anforderungen, die für andere Prozesse von Bedeutung sind.

ANMERKUNG 1 Es wird vorausgesetzt, dass bei allen Raumfahrtprojekten bestimmte Verifizierungs- und Validierungsarbeiten durchgeführt werden. Daher gelten die Festlegungen von ISO/IEC 12207, nach denen die Notwendigkeit dieser Tätigkeiten zu ermitteln ist, hier nicht (Anpassung von ISO/IEC 12207).

ANMERKUNG 2 Die Prozessverifizierung des Lieferanten gilt als Teil des Managements und ist daher nicht Gegenstand des Software-Engineering (Anpassung von ISO/IEC 12207:1995, 6.4.2.2).

5.6.2 Meilensteine

5.6.2.1 Das Qualifikationsreview (QR) muss nach 5.3.2.10 durchgeführt werden.

ZIEL: Nachweis dafür, dass die Software alle Anforderungen erfüllt und besonders, dass es die Ergebnisse dieser Verifizierung und Validierung erlauben, den Softwareprodukten den „qualifizierten Zustand“ zuzuordnen.

ERWARTETES ERGEBNIS:

- a) Vorläufiges Datenpaket für Softwareabnahme [DJF; QR];
- b) Vorläufige Dokumentation der Softwarefreigabe [DDF; QR];
- c) Vorläufige Softwarelieferung auf festgelegtem Datenträger [DDF; QR];
- d) Bericht über die Bewertung von Softwaredesign und Test [DJF; QR];
- e) Validierungstestbericht [DJF; QR];
- f) Bewertung der Testspezifikation [DJF; QR];
- g) Aufzeichnungen zum Qualifikationsreview [DJF; QR].

5.6.2.2 Das Abnahmereview (AR) muss nach 5.3.2.11 durchgeführt werden. Dabei muss der entsprechende Prozess des Softwarelieferanten die mit der Abnahme im Zusammenhang stehenden Arbeiten des Kunden in Vorbereitung auf das Review unterstützen.

ZIEL: Sicherstellen, dass der Kunde angemessene Unterstützung des Lieferanten bei der Ausführung seiner Abnahme- und Integrationsarbeit in Vorbereitung auf das AR erhält und zwar im Rahmen der „Unterstützung von Abnahmereviews und Abnahmetests“ (5.6.7.8).

ERWARTETES ERGEBNIS:

- a) Endgültiges Datenpaket für Softwareabnahme [DJF; AR];
- b) Abnahmetestdokumentation [DJF; AR];
- c) Aufzeichnungen zum Abnahmereview [DJF; AR];
- d) Dokumentation der Softwarefreigabe [DDF; AR];
- e) Softwarelieferung auf festgelegtem Datenträger [DDF; AR].

5.6.3 Verifizierungs- und Validierungsprozesse

Da auf folgende Abschnitte an anderen Stellen dieser Norm Bezug genommen wird, wird das Ziel des Prozesses im Einzelfall nicht ausdrücklich angegeben.

Das mit der Software-Verifizierung und -Validierung verbundene Engineering umfasst:

- Vorbereitende Arbeiten zur Verifizierung;
- Vorbereitende Arbeiten zur Validierung;
- Durchführung der Verifizierung;
- Durchführung der Validierung;
- gemeinsame technische Reviews.

5.6.4 Vorbereitende Arbeiten zur Verifizierung

5.6.4.1 Die Arbeiten umfassen:

- Bestimmung des für das Projekt zu leistenden Verifizierungsaufwands;
- Festlegen des Verifizierungsprozesses;
- Auswahl der für die Verifizierung verantwortlichen Organisation;
- Festlegen der mit dem Lebenszyklus zusammenhängenden Tätigkeiten sowie der Softwareprodukte, die eine Verifizierung erfordern;
- Entwicklung und Dokumentation eines Verifizierungsplans.

5.6.4.2 Bestimmung des Verifizierungsaufwands

Der für die Verifizierung zu leistende Aufwand ist festzulegen wie auch der Grad der organisatorischen Unabhängigkeit der damit einhergehenden Tätigkeiten. Ebenso müssen die Normen EN 13290-1:1999, 5.3 (Risikomanagement) und ECSS-Q-80A, 3.2.2 (Softwarezuverlässigkeit und -sicherheit) und 3.2.5.p (unabhängige Software-Verifizierung und -Validierung) auf Anwendbarkeit geprüft werden. Die Projektanforderungen sind auf innewohnende Risiken zu prüfen. Diese können sein:

- a) Unerkannte Fehler in für das System oder die Software geltenden Anforderungen, die die Verletzung oder den Tod von Menschen, ein Scheitern der Mission oder einen finanziellen Verlust bzw. eine Beschädigung oder einen katastrophalen Verlust von Geräten und Ausrüstung zur Folge hat;

- b) Grad der Reife der anzuwendenden Softwaretechnologie und damit verbundene Risiken;
- c) Verfügbarkeit von Finanzmitteln und Ressourcen.

ERWARTETES ERGEBNIS: Software-Verifizierungsplan — Risiken und Aufwand.

5.6.4.3 Festlegen des Verifizierungsprozesses

Zum Verifizieren der Software ist ein Verifizierungsprozess zu entwickeln.

5.6.4.4 Auswahl der verantwortlichen Organisation

Ermöglicht das Projekt eine unabhängige Verifizierung, so muss dazu eine kompetente Organisation ausgewählt werden. Diese muss die Unabhängigkeit wie auch die Berechtigung zur Durchführung von Verifizierungstätigkeiten zusichern. Siehe auch ECSS-Q-80A, 3.2.5.p (unabhängige Software-Verifizierung und -Validierung), EN 13290-1:1999, 6.2.3, und EN 13290-3:2001 (Projektorganisation) zu finden.

ZIEL: Entwickeln eines schlüssigen Konzepts für die Projektorganisation für jedes Projekt.

ERWARTETES ERGEBNIS: Dokumente mit Anforderungen an Projektelemente, die die Projektorganisation zum Gegenstand haben.

5.6.4.5 Festlegen der mit dem Lebenszyklus zusammenhängenden Tätigkeiten sowie der Softwareprodukte, die eine Verifizierung erfordern

Unter Berücksichtigung von Umfang, Größe und Komplexität des Projekts und der oben erwähnten Risikoanalyse müssen die auf den Lebenszyklus bezogenen Tätigkeiten sowie die Softwareprodukte, die zu verifizieren sind, festgelegt werden. Dafür müssen die in 5.6.6 beschriebenen Tätigkeiten und Aufgaben einschließlich zugehöriger Verfahren, Technik und Werkzeuge ausgewählt werden.

ERWARTETES ERGEBNIS: Verifizierungsplan — Verfahren und Werkzeuge.

5.6.4.6 Entwicklung und Dokumentation eines Plans zur Verifizierung

Entsprechend den festgelegten Verifizierungsaufgaben ist ein Verifizierungsplan zu entwickeln und dokumentieren. Der Plan muss sich auf die mit dem Lebenszyklus zusammenhängenden Tätigkeiten sowie die Softwareprodukte, die zu verifizieren sind, die dafür erforderliche Verifizierungstätigkeit, die zugehörigen Mittel, Zuständigkeiten und den Zeitplan beziehen. Er muss die Verfahren zur Übermittlung des Verifizierungsberichts an den Kunden und andere betroffene Organisationen regeln.

ERWARTETES ERGEBNIS: Verifizierungsplan — Organisation.

5.6.5 Vorbereitende Arbeiten zur Validierung

5.6.5.1 Allgemeines

Die Arbeiten umfassen:

- Bestimmung des für das Projekt zu leistenden Validierungsaufwands;
- Festlegen des Validierungsprozesses;
- Auswahl der für die Validierung verantwortlichen Organisation;
- Entwicklung und Dokumentation des Validierungsplans.

5.6.5.2 Bestimmung des Validierungsaufwands

Auf der Basis von ECSS-Q-80A, 3.2.5.p, sind der Validierungsaufwand und der Grad der organisatorischen Unabhängigkeit der damit zusammenhängenden Tätigkeiten zu bestimmen.

ERWARTETES ERGEBNIS: Validierungsplan — Aufwand und Grad der Unabhängigkeit.

5.6.5.3 Festlegen des Validierungsprozesses

Zur Validierung des Softwareprodukts ist ein Validierungsprozess festzulegen. Dabei ist unter den in 5.6.7 festgelegten Validierungsaufgaben einschließlich zugehöriger Verfahren, Technik und Werkzeugen eine Auswahl zu treffen.

ERWARTETES ERGEBNIS: Validierungsplan — Verfahren und Werkzeuge.

5.6.5.4 Wahl einer Validierungsorganisation

Ermöglicht das Projekt eine unabhängige Validierung, muss dazu eine kompetente Organisation ausgewählt werden. Diese muss die Unabhängigkeit wie auch die Berechtigung zur Durchführung von Validierungstätigkeiten zusichern. 6.2.3 von EN 13290-1:1999 ist in diesem Zusammenhang zu beachten.

ERWARTETES ERGEBNIS:

- a) Dokumente mit Anforderungen an Projektdaten, die die Projektorganisation zum Gegenstand haben;
- b) Plan über eine unabhängige Validierung.

5.6.5.5 Entwicklung und Dokumentation eines Plans zur Software-Validierung

Es ist ein Validierungsplan zu entwickeln und dokumentieren. Er muss mindestens Folgendes regeln:

- a) Elemente, die zu validieren sind;
- b) auszuführende Validierungstätigkeiten;
- c) Mittel, Zuständigkeiten und Zeitplan für die Validierung;
- d) Verfahren der Übermittlung der Validierungsberichte an den Kunden und andere Beteiligte.

ERWARTETES ERGEBNIS: Validierungsplan — Organisation.

5.6.6 Durchführung der Verifizierung

5.6.6.1 Allgemeines

Die Arbeiten umfassen:

- Verifizierung der Softwareanforderungen;
- Verifizierung des Softwaredesigns;
- Verifizierung des Software-Codes;
- Verifizierung der Softwareintegration;
- Verifizierung der Softwaredokumentation;
- Bewertung der Testspezifikationen;
- Behandlung von Problemen und Fehlern.

5.6.6.2 Verifizierung der Anforderungen an die Software

Die Anforderungen an die Software müssen anhand nachfolgender Kriterien verifiziert werden.

- a) Die Softwareanforderungen sind miteinander kompatibel (was nicht Kompatibilität in formaler Hinsicht bedeutet), realisierbar, verifizierbar und entsprechen genau den Systemanforderungen.
- b) Die Softwareanforderungen, die sich auf die persönliche Sicherheit, Datensicherheit und Risiken beziehen, sind insoweit richtig, als sich dies durch geeignete genaue Verfahren nachweisen lässt.