

DIN EN 9721



ICS 35.240.99; 49.020

**Luft- und Raumfahrt –
Allgemeine Empfehlungen für die integrierte Prüfungs-(BIT)-Architektur
in einem integrierten System;
Deutsche und Englische Fassung EN 9721:2021**

Aerospace series –
General recommendation for the BIT Architecture in an integrated system;
German and English version EN 9721:2021

Série spatiale –
Recommandations générales pour l'architecture des BIT dans un système intégré;
Version allemande et anglaise EN 9721:2021

Gesamtumfang 180 Seiten

DIN-Normenausschuss Luft- und Raumfahrt (NL)

Nationales Vorwort

Der Verband der Europäischen Luft-, Raumfahrt- und Verteidigungsindustrie – Normung (ASD-STAN) ist vom Europäischen Komitee für Normung (CEN) für zuständig erklärt worden, Europäische Normen (EN) für das Gebiet der Luft- und Raumfahrt auszuarbeiten. Durch die Vereinbarung vom 3. Oktober 1986 wurde ASD Assoziierte Organisation (ASB) des CEN.

Das vorliegende Dokument (EN 9721:2021) wurde von ASD-STAN, Fachbereich Digitale Projekte, unter Mitwirkung deutscher Experten des DIN-Normenausschusses Luft- und Raumfahrt (NL) erarbeitet.

Das zuständige deutsche Normungsgremium ist der Arbeitsausschuss NA 131-04 FBR „Fachbereichsbeirat Elektrotechnik, Avionik“ im DIN-Normenausschuss Luft- und Raumfahrt (NL).

Entsprechend Beschluss 57/9 des Technischen Ausschusses des Beirats des DIN-Normenausschusses Luft- und Raumfahrt (NL) 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 dieses Dokuments die Englische Fassung hinzugefügt.

Für die in diesem Dokument zitierten Dokumente wird im Folgenden auf die entsprechenden deutschen Dokumente hingewiesen:

IEC 60812	siehe	DIN EN 60812
ISO 5577	siehe	DIN EN ISO 5577

Aktuelle Informationen zu diesem Dokument können über die Internetseiten von DIN (www.din.de) durch eine Suche nach der Dokumentennummer aufgerufen werden.

Nationaler Anhang NA (informativ)

Literaturhinweise

DIN EN 60812, *Analysetechniken für die Funktionsfähigkeit von Systemen — Verfahren für die Fehlzustandsart- und -auswirkungsanalyse (FMEA)*

DIN EN ISO 5577, *Zerstörungsfreie Prüfung — Ultraschallprüfung — Terminologie*

Deutsche Fassung

Luft- und Raumfahrt —
Allgemeine Empfehlungen für die integrierte
Prüfungs-(BIT)-Architektur in einem integrierten System

Aerospace series —
General recommendation for the BIT Architecture in an
integrated system

Série aérospatiale —
Recommandations générales pour l'architecture des
BIT dans un système intégré

Diese Europäische Norm wurde vom CEN am 2. November 2020 angenommen.

Diese Europäische Norm wurde korrigiert und vom CEN-CENELEC-Management-Zentrum am 23. Februar 2022 neu herausgegeben.

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 CEN-CENELEC-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, Bulgarien, Dänemark, Deutschland, Estland, Finnland, Frankreich, Griechenland, Irland, Island, Italien, Kroatien, Lettland, Litauen, Luxemburg, Malta, den Niederlanden, Norwegen, Österreich, Polen, Portugal, der Republik Nordmazedonien, Rumänien, Schweden, der Schweiz, Serbien, der Slowakei, Slowenien, Spanien, der Tschechischen Republik, der Türkei, Ungarn, dem Vereinigten Königreich und Zypern.



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

CEN-CENELEC Management-Zentrum: Rue de la Science 23, B-1040 Brüssel

Inhalt

	Seite
Europäisches Vorwort	4
Einleitung	5
1 Anwendungsbereich	6
2 Normative Verweisungen	6
3 Begriffe und Abkürzungen	6
3.1 Begriffe	6
3.2 Abkürzungen	12
4 <i>BIT</i> -Beteiligte	13
4.1 <i>BIT</i> -Spezifizierer	13
4.2 <i>BIT</i> -Entwerfer/Entwickler	13
4.3 Betrieblicher Nutzer	14
4.4 Instandhaltungstechniker	14
4.5 Technischer Leiter des Systems	14
4.6 Fachkraft	14
4.7 Einsatzdatentechniker	14
5 Systemvorgaben	15
5.1 Systementwurf	15
5.2 <i>BIT</i> -Schnittstellenfunktion	15
5.2.1 Die Alarmfunktion	15
5.2.2 Die Diagnosefunktion	16
5.2.3 Integrierte Rekonfiguration	17
5.2.4 Die Instandhaltungsfunktion	17
5.2.5 Die Datenaufzeichnung für die Nachanalysefunktion	17
5.3 Technische Systemzustände	18
5.4 Funktionsarten eines Systems	18
5.5 Systemkonfiguration	19
5.5.1 Betriebliche Systemkonfiguration	19
5.5.2 Technische Konfiguration	19
5.5.3 Parametrierung des <i>BIT</i>	19
6 Arten und Metriken des <i>BIT</i>	20
6.1 Allgemeines	20
6.2 Die verschiedenen Arten von <i>BIT</i>	20
6.2.1 Hochlauf- <i>BIT</i> oder Einschalt- <i>BIT</i> (<i>PBIT</i> — en: Power-on/up <i>BIT</i>)	20
6.2.2 Eingeleitetes <i>BIT</i> (<i>IBIT</i> — en: Initiated <i>BIT</i>) oder angefordertes <i>BIT</i> (<i>DBIT</i> — en: Demanded <i>BIT</i>)	20
6.2.3 Kontinuierliches <i>BIT</i> (<i>CBIT</i> — en: Continuous <i>BIT</i>)	21
6.2.4 Externes <i>BIT</i> (<i>EBIT</i> — en: External <i>BIT</i>)	21
6.2.5 Instandhaltungs- <i>BIT</i> (<i>MBIT</i> — en: Maintenance <i>BIT</i>)	21
6.2.6 Zusammenfassung der Eigenschaften der verschiedenen Arten von <i>BIT</i>	22
6.3 Die Metriken	22
6.3.1 Rolle der mathematischen Definitionen der Metriken	22
6.3.2 Erkennungsrate	23
6.3.3 Eingrenzungsrate	25
6.3.4 Durch das <i>BIT</i> verursachte Steigerungsrate der Unzuverlässigkeit	27
6.3.5 Fehlalarmraten, Fehlerraten bei Betriebsausfall	28

7	Anwendung von internen Prüfverfahren (<i>BIT</i>)	33
7.1	Während der Entwicklung	33
7.2	Während der Produktion	33
7.3	Während des Einsatzes	34
7.3.1	Im Betriebszustand.....	34
7.3.2	Im Instandhaltungsbetrieb	34
7.4	Während der Validierung bei der Reparatur.....	34
8	Architektur von internen Prüfverfahren (<i>BIT</i>)	34
8.1	Die allgemeingültigen Funktionen der <i>BIT</i>	34
8.1.1	Allgemeines	34
8.1.2	<i>BIT</i> -Erkennungsfunktion	37
8.1.3	<i>BIT</i> -Supervisor-Funktion.....	38
8.2	Die verschiedenen Architekturen der <i>BIT</i> -Funktion.....	40
8.2.1	Allgemeines	40
8.2.2	Verteilte <i>BIT</i> -Architektur	41
8.2.3	Zentralisierte <i>BIT</i> -Architektur	42
8.2.4	Wahl der <i>BIT</i> -Architektur.....	42
8.3	Topologie der ausgetauschten Daten.....	43
8.4	<i>BIT</i> -Spezifikation und Modellierung	44
8.4.1	Spezifikationsprozess.....	44
8.4.2	Entscheidungen des Systementwurfs: wesentliches Ziel und Aufwand.....	44
8.4.3	Der <i>BIT</i> -Spezifikationsprozess	46
8.5	Allgemeine Modellierungs- und Konfigurationssprache.....	47
8.5.1	Einleitung	47
8.5.2	Allgemeine Informationen	49
8.5.3	Beschreibung der Sprachtabellen	50
8.5.4	Funktionssprache.....	57
8.5.5	Instanziierungsprozess des Modells	58
8.6	Entwicklungsprozess und Validierung/Verifizierung eines <i>BIT</i> -Systems.....	58
9	Prognose	58
9.1	Ziel der Prognose.....	58
9.2	Organisation der Prognose	59
9.3	Für die Prognose genutzte Daten aus dem <i>BIT</i>	59
10	Schlussfolgerungen.....	60
Anhang A (informativ) Beispiele		61
A.1	Betriebseffizienz und Leistung.....	61
A.1.1	Allgemeines	61
A.1.2	Beispiel 1: Wie ist ein Baum schnell zu fällen?.....	61
A.1.3	Beispiel 2: Wie kann ein Stück Butter sauber abgeschnitten werden?	61
A.2	Beispiel für die Berechnung einiger Metriken	62
A.2.1	Allgemeines	62
A.2.2	Berechnung der Erkennungsraten.....	65
A.2.3	Berechnung der Eingrenzungsraten	67
A.3	Diagnose des einwandfreien Betriebs gegenüber der Diagnose des Ausfalls	72
A.4	Beispiel für die Fortpflanzung der Diagnosewerte im Fall einer einfachen Architektur	73
A.5	Hypothese der Ergodizität	80
A.6	Berechnungsbeispiel für die Bewertung der <i>NFF</i> -Rate – kein Fehler festgestellt.....	80
A.7	Ablaufdiagramm der Ereignisse	83
Anhang B (informativ) Liste der Empfehlungen		85
Literaturhinweise.....		89
Stichwortverzeichnis		90

Europäisches Vorwort

Dieses Dokument (EN 9721:2021) wurde vom Verband der Europäischen Luft-, Raumfahrt- und Verteidigungsindustrie — Normung (ASD-STAN) erstellt.

Nachdem Umfragen und Abstimmungen entsprechend den Regeln dieses Verbandes durchgeführt wurden, hat dieses Dokument die Zustimmung der nationalen Verbände und offiziellen Behörden der Mitgliedsländer von ASD-STAN erhalten, bevor es CEN vorgelegt wurde.

Dieses Dokument muss den Status einer nationalen Norm erhalten, entweder durch Veröffentlichung eines identischen Textes oder durch Anerkennung bis Juni 2022, und etwaige entgegenstehende nationale Normen müssen bis Juni 2022 zurückgezogen werden.

Es wird auf die Möglichkeit hingewiesen, dass einige Elemente dieses Dokuments Patentrechte berühren können. CEN [und/oder CENELEC] ist/sind nicht dafür verantwortlich, einige oder alle diesbezüglichen Patentrechte zu identifizieren.

Entsprechend der CEN-CENELEC-Geschäftsordnung sind die nationalen Normungsinstitute der folgenden Länder gehalten, diese Europäische Norm zu übernehmen: Belgien, Bulgarien, Dänemark, Deutschland, die Republik Nordmazedonien, Estland, Finnland, Frankreich, Griechenland, Irland, Island, Italien, Kroatien, Lettland, Litauen, Luxemburg, Malta, Niederlande, Norwegen, Österreich, Polen, Portugal, Rumänien, Schweden, Schweiz, Serbien, Slowakei, Slowenien, Spanien, Tschechische Republik, Türkei, Ungarn, Vereinigtes Königreich und Zypern.

Einleitung

Ein internes Prüfverfahren (*BIT*) ist eine Prüfung, die ausschließlich mit den für ein Ausrüstungselement bzw. für ein System spezifischen Hardware- und Software-Ressourcen durchgeführt wird, um diese und/oder deren Unterbaugruppen zu prüfen, um Ausfälle zu erkennen und diese einzugrenzen oder sogar zu bestimmen.

Systementwickler sind mit den folgenden Fragen konfrontiert:

- Wie ist eine Strategie oder ein Verfahren für eine Prüfung, die in ein System zu integrieren ist, festzulegen?
- Wie ist die Betriebseffizienz der *BIT*-Architektur eines Systems zu bewerten? (Fehlalarme, nicht reproduzierbare Alarme und falsche Außerbetriebnahmen)
- Wie ist eine schlüssige *BIT*-Architektur zwischen den verschiedenen Ebenen eines Systems (eines Systems von Systemen) zu erzielen?
- Wie sind die Bedürfnisse der verschiedenen Nutzer der *BIT*-Funktion zu berücksichtigen, wenn in Betracht gezogen wird, dass die Implementierung, die Zugriffe, die *BIT*-Berichte usw. für die Nutzer spezifisch sind?

1 Anwendungsbereich

Zweck dieses Dokuments ist es, den Dialog zwischen Herstellern, Hauptauftragnehmern, Eigentümern und dem Kunden zu harmonisieren, um die Erstellung von Spezifikationen, die gemeinsame Nutzung von *BIT*-Architekturmodellen und die technische *BIT*-Konfiguration von Systemen während des Betriebs zu erleichtern.

In dieser Empfehlung wird vorgeschlagen, Definitionen der *BIT*-Betriebseffizienz und der *BIT*-Leistung sowie Grundsätze für den Entwurf der Architektur und die *BIT*-Spezifikation oder -Validierung aufzustellen. Es werden keine Empfehlungen für die numerischen Werte der Betriebseffizienz oder der Leistung gegeben. Die Vielfalt der Situationen, die Entwicklung technologischer Lösungen und die sich ständig ändernden Betriebsanforderungen machen die Aufstellung allgemeiner Empfehlungen unmöglich.

Abschnitt 6 und Abschnitt 9 legen den allgemeinen Nutzungszusammenhang des *BIT* dar.

Abschnitt 7 führt Vorgaben auf, die beim Entwurf einer *BIT*-Architektur zu berücksichtigen sind.

Abschnitt 8 enthält ein Verzeichnis verschiedener gegenwärtig bekannter *BIT*-Arten und die Definitionen der Leistung und der Betriebseffizienz (Metriken).

Abschnitt 10 gibt Empfehlungen für die *BIT*-Architektur.

Abschnitt 11 empfiehlt eine Sprache für den Austausch von *BIT*-Architekturmodellen für die Aufstellung des Gesamtmodells eines Systems.

Abschnitt 12 ist eine Einführung in die Prognose.

Dieses Dokument ist in erster Linie für Systementwickler vorgesehen.

Obwohl sie auf Beispielen von Luftfahrtssystemen beruht, ist sie für jede Systemart anwendbar.

2 Normative Verweisungen

Die folgenden Dokumente werden im Text in solcher Weise in Bezug genommen, dass einige Teile davon oder ihr gesamter Inhalt Anforderungen des vorliegenden Dokuments darstellen. Bei datierten Verweisungen gilt nur die in Bezug genommene Ausgabe. Bei undatierten Verweisungen gilt die letzte Ausgabe des in Bezug genommenen Dokuments (einschließlich aller Änderungen).

ISO 5577, *Non-destructive testing — Ultrasonic testing — Vocabulary*

3 Begriffe und Abkürzungen

3.1 Begriffe

Für die Anwendung dieses Dokuments gelten die Begriffe nach ISO 5577 und die folgenden Begriffe.

ISO und IEC stellen terminologische Datenbanken für die Verwendung in der Normung unter den folgenden Adressen bereit:

- ISO Online Browsing Platform: verfügbar unter <https://www.iso.org/obp>
- IEC Electropedia: verfügbar unter <https://www.electropedia.org/>

3.1.1

Mehrdeutigkeitsgruppe

mit einer Signatur verbunden und besteht aus einer Reihe von austauschbaren Elementen des Systems, wobei mindestens einer der Ausfälle zu dieser Signatur beiträgt, der aber nicht eindeutig bestimmt werden kann; abhängig von der betrachteten Instandhaltungsebene können die austauschbaren Elemente *LRU*, *SRU*, Komponenten usw. sein; der Begriff der Mehrdeutigkeitsgruppe bezieht sich auf die Anforderung der Eingrenzung des ausgefallenen Elements in dem mit einer bestimmten Instandhaltungsebene geprüften System

3.1.2

Schnittmenge

Kombination von Ausfällen aus der Gesamtliste der möglichen Ausfälle (systemintern oder -extern), die zum Verlust eines Diensts führen; sie gilt als minimal, wenn der Dienst durch Beseitigung eines Ausfalls aus der Liste nicht mehr ausfällt; die Größe (oder der Umfang) der Schnittmenge entspricht der Anzahl der Elemente in der Liste

BEISPIEL

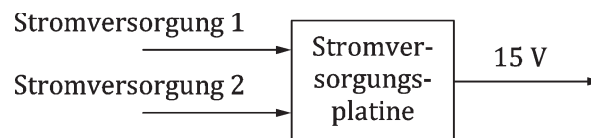


Bild 1 — Beispiel für eine Schnittmenge

In diesem Beispiel wird angenommen, dass die Stromversorgung 1 und die Stromversorgung 2 als doppelt redundante Teile arbeiten. Daher versagt der Dienst „Bereitstellen von 15 V“, falls beide Stromversorgungen ausfallen.

Es gibt zwei einzelne Minimalschnitte mit dem gleichen Dienstausfall (Verlust der 15-V-Versorgung):

- Schnittmenge 1: Verlust der „Stromversorgung 1“ UND Verlust der „Stromversorgung 2“ (vorgeschalteter Ausgang ausgefallen);
- Schnittmenge 2: Ausfall der „Stromversorgungsplatine“.

Es gibt jedoch viele Schnittmengen, die kein Minimalschnitt sind. Die folgende Schnittmenge 3 ist z. B. kein Minimalschnitt:

- Schnittmenge 3: (Verlust der „Stromversorgung 1“ UND Ausfall der „Stromversorgungsplatine“) ODER (kein Verlust der „Stromversorgung 1“ UND Ausfall der „Stromversorgungsplatine“).

Schnittmenge 2 ist Schnittmenge 3 vorzuziehen.

3.1.3

Defekt

Nichterfüllung einer Anforderung im Zusammenhang mit einer festgelegten oder erwarteten Nutzung

3.1.4

Ursache der Funktionsbeeinträchtigung oder Ausfallursache

Umstände in Bezug auf den Entwurf, die Herstellung und die Nutzung, die zum Ausfall oder zu einem Vorfall führen

Anmerkung 1 zum Begriff: In diesem Dokument wird davon ausgegangen, dass es keine Systementwurfsfehler gibt.

[QUELLE: übernommen aus NF X60-500]

3.1.5

Funktionsbeeinträchtigung

allmähliche und teilweise Veränderung der Fähigkeit des Systems, bestimmte, jedoch nicht alle geforderten Funktionen auszuführen

3.1.6

ingeschränkter Funktionszustand

Zustand eines Systems nach einer Funktionsbeeinträchtigung (siehe Definition für die Funktionsbeeinträchtigung)

3.1.7

Erkennbarkeit

Fähigkeit der Ausfallerkennung eines Systems wird für jeden der Ausfälle, der auftreten kann, daraufhin beurteilt, ob ein Ausfall erkennbar ist oder nicht

Anmerkung 1 zum Begriff: Die Erkennbarkeit ist eine Metrik, die die Betriebseffizienz einer Architektur bewertet. Sie berücksichtigt die Betriebseffizienz der Prüfungen (oder setzt die gesamte Betriebseffizienz der Prüfungen voraus).

3.1.8

Diagnose

Bestimmung der wahrscheinlichen Ursache des Ausfalls (oder der Ausfälle) mit Hilfe einer logischen Beweisführung auf der Grundlage einer Informationsmenge, die aus einer Inspektion, einer Kontrolle oder einer Prüfung stammt

3.1.9

störende Prüfung

Prüfung, die wahrscheinlich das Betriebsverhalten des geprüften Elements verändert

3.1.10

Auswirkung

Ergebnis einer Ursache; diese Auswirkung im System kann kaskadierend (Dominoeffekt) sein; sie ist dann eine Ursache mit Auswirkungen, die sich bis in die oberste Ebene fortpflanzen können

3.1.11

Ausfall

Beenden der Fähigkeit eines Systems, die geforderte Funktion auszuführen; er ist durch seine Auswirkungen (mangelndes Verhalten) wie die Abweichung einer physikalischen Variablen über einen angegebenen Toleranzbereich hinaus erkennbar; in diesem Dokument wird er mit f bezeichnet

[QUELLE: übernommen aus NF X60-500]

3.1.12

Ausfalleingrenzung

<Fehlersuche> Verringerung der Größe der Mehrdeutigkeitsgruppe durch Beobachtungen oder zusätzliche Prüfungen; der Prozess der Ausfalleingrenzung (Fehlersuche) ist iterativ; er endet, wenn die Diagnose nicht mehr mehrdeutig ist und die Fehlersuche validiert ist

3.1.13

Ausfallmengen

in diesem Dokument werden mehrere Ausfallmengen (im mathematischen Sinn des Begriffs) betrachtet; das sind

- E : Menge aller Ausfälle: $E = HF \cup SDF = \{f_i\}$ für i von 1 bis $\text{Card}(E)$;
- HF : Menge von Ausfällen, die durch die Hardware verursacht werden. Diese Menge schließt Hardware-Entwurfsfehler aus;

- *DF*: Menge von Ausfällen, die durch die betreffende Prüfung erkennbar ist. *DF* ist in *E* enthalten;
- *FE*: Menge von Ausfällen mit einer Auswirkung, die als „zu berücksichtigen“ gilt (z. B. hinsichtlich der Kritikalität, eines angegebenen Einsatzszenarios usw.). *FE* ist in *E* enthalten. Der Umfang von *FE* hängt vom Zweck der Analyse und damit von der Art der Auswirkung ab, und zwar betrieblich, sicherheitsbezogen, wirtschaftlich usw.;
- *DFE*: Menge von erkennbaren Ausfällen mit einer Auswirkung, die als „zu berücksichtigen“ gilt; $DFE = DF \cap FE$; und
- *SDF*: Menge von Software-Entwurfsfehlern (unabhängig davon, ob sie von einem Mikroprozessor oder von einem programmierbaren Bauteil ausgeführt wird). Theoretisch sollte diese Menge leer sein. Folglich werden diese Softwareausfällen bei den *FMECA*-Analysen oder bei den Berechnungen der Erkennungsrate nicht berücksichtigt. Die Erfahrung zeigt jedoch, dass sie vorhanden sind und einige von ihnen durch Integritätsprüfungen erkannt werden können:

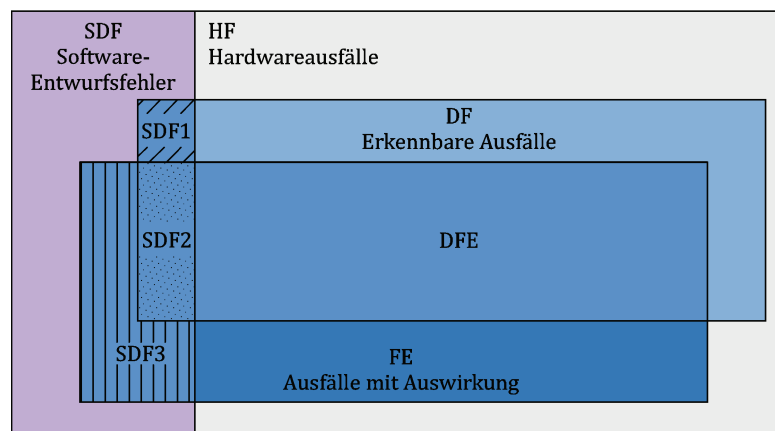


Bild 2 — Ausfallmengen

- *SDF1*: Erkennbare Software-Entwurfsfehler, die keine Auswirkung haben;
- *SDF2*: Erkennbare Software-Entwurfsfehler, die eine Auswirkung haben;
- *SDF3*: Software-Entwurfsfehler, die eine Auswirkung haben, die aber durch Integritätsprüfungen nicht erkannt werden

Anmerkung 1 zum Begriff: Software-Entwurfsfehler werden im weiteren Verlauf dieses Dokuments nicht berücksichtigt. Der Schwerpunkt dieses Dokuments ist die Menge *HF*. Folglich, und etwas ungenau verwendet, werden alle im Verlauf dieses Dokuments genannten Mengen so verstanden, dass sie sich auf die Überschneidung mit *HF* beschränken.

Anmerkung 2 zum Begriff: Mathematische Erinnerung: Der Kardinalzahl der Menge *E*, die mit „*Card(E)*“ bezeichnet wird, ist die Anzahl der Elemente, die diese Menge bilden.

3.1.14

Ausfallsignatur

erschöpfende (nicht minimale) Kombination von sichtbaren Anzeichen (mit dem Ergebnis OK oder NOK), die aus dem Ausfall eines Diensts resultieren; die Signatur besteht aus einem Kern und einer Peripherie; der Signaturkern ist die Kombination von Anzeichen, die immer sichtbar sind und ausreichen, um den Ausfall zu diagnostizieren; die Peripherie ist die Menge von Anzeichen, die den Kern begleiten können (Erscheinung des kaskadierenden Ausfalls)

Anmerkung 1 zum Begriff: **Empfehlung 5:** In Bezug auf Ausfallsignaturen ist es wichtig anzugeben, ob der Signaturkern (Entwurfsansatz) gemeint ist oder die Zusammensetzung aus Signatur-„Kern + Peripherie“ (Instandhaltungsansatz).

Anmerkung 2 zum Begriff: Mehrere Ausfälle dürfen die gleiche Signatur haben.

Anmerkung 3 zum Begriff: Wenn diese Signatur mit einer Mehrdeutigkeitsgruppe der Größe n verbunden ist, ist n der Grad der Signatur.

3.1.15

Fehler (Fehlzustand)

interne Ursache, die zu einem Ausfall führt; bei einem Fehler ist das Produkt im Fehlzustand

Anmerkung 1 zum Begriff: Das System kann seinen Dienst z. B. dann weiterhin ausführen, wenn die Architektur redundant ist.

[QUELLE: übernommen aus NF X60-500]

3.1.16

Fehlalarm

Ergebnis einer zwischen zwei möglichen Auswahlmöglichkeiten (positiv und negativ) getroffenen Entscheidung, das als positiv angegeben wird, wenn es tatsächlich negativ ist

3.1.17

Liste der Systemausfälle

Menge der Ausfälle, die während der Entwurfsphase ermittelt werden und die durch Erfahrungsrückfluss erweitert wird; sie kann formal in einem *FMECA*-Bericht [2] angegeben werden; für jeden Ausfall enthält sie mindestens:

- seine Auftretensrate (außer bei Ausfällen mit Ursachen, die nicht zum Anwendungsbereich des betrachteten Systems gehören);
- seine Auswirkung(en);
- die *LRU/SRU* oder die Ressource/Bedingung, die nicht zum betrachteten System gehört

Anmerkung 1 zum Begriff: **Empfehlung 4:** Die Gestaltung der Prüfbarkeit und die Prüfungen müssen auf der Liste der Systemausfälle beruhen.

Anmerkung 2 zum Begriff: Der Detaillierungsgrad der Liste der Ausfälle muss so genau sein, dass die Bedeutung der Werte aus den Metriken sichergestellt wird.

Anmerkung 3 zum Begriff: Dazu sollten die Ausfallarten der Funktionen, die durch austauschbare Elemente auf der gewählten Instandhaltungsebene bereitgestellt werden, definiert werden.

3.1.18

Betriebseffizienz

<Lösung> misst entweder

- das Niveau des erzielten Ergebnisses im Hinblick auf die angestrebte Auswirkung nach Zeit- oder Aufwandseinheiten durch den Bediener oder
- die erforderliche Zeit oder den Aufwand durch den Bediener, um das Niveau des erwarteten Ergebnisses im Hinblick auf die angestrebte Auswirkung zu erzielen

Anmerkung 1 zum Begriff: Die Betriebseffizienz ist eine betriebliche Metrik.

Anmerkung 2 zum Begriff: Die Betriebseffizienz ist das Ergebnis der Leistung und des Nutzungszusammenhangs:

Betriebseffizienz = Funktion (Leistung, Zusammenhang)

Anmerkung 3 zum Begriff: Dieser Begriff wird durch das in A.1 angegebene Beispiel veranschaulicht.

3.1.19

Leistung

eigentliche Qualität der Lösung unabhängig vom Nutzungszusammenhang; sie ist eine Entwurfsmetrik

Anmerkung 1 zum Begriff: Dieser Begriff wird durch das in A.1 angegebene Beispiel veranschaulicht.

3.1.20

Anzeichen

physische Manifestation eines Ausfalls; Anzeichen können durch Inspektion, durch Prüfungen oder aus den Einsatzinformationen des Systems beobachtet werden

BEISPIEL

Das Anzeichen für einen Ausfall (Fahrwerk nicht ausgefahren) ergibt z. B. für die Prüfung „Ist das Fahrwerk ausgefahren?“ das Ergebnis *NOK*, wenn die Information „Fahrwerk oben“ nach einem Befehl „Fahrwerk ausfahren“ codiert wird. Entsprechend wird das Ergebnis *OK* sein, wenn nach dem Senden des Befehls die Information „Fahrwerk unten“ codiert ausgegeben wird.

Unterscheidung zwischen Anzeichen und Prüfergebnis:

Ein Prüfergebnis ist der codierte Ausdruck des Beobachtungsergebnisses eines Anzeichens.

Es gibt zwei Arten von Prüfergebnissen:

- primäre Prüfergebnisse: solche Prüfergebnisse, die sich aus der direkten Beobachtung eines Anzeichens ergeben;
- zusammenfassende Prüfergebnisse: solche Prüfergebnisse, die sich aus einer Gleichung ergeben, die andere Prüfergebnisse miteinander kombiniert (primäre oder zusammenfassende Prüfergebnisse).

Unterscheidung zwischen Anzeichen und codierter Information:

Im Beispiel des Fahrwerks stammt das Anzeichen „Fahrwerk nicht ausgefahren“ aus einer Kombination von codierten Einsatzinformationen: aus dem Befehl „Fahrwerk ausfahren“ und der Information „Fahrwerk oben“, die nicht aus dem *BIT* stammt.

3.1.21

Systemausfallrate

λ_f

Auftretenshäufigkeit des Ausfalls f , ausgedrückt als Anzahl des Auftretens je Stunde

Anmerkung 1 zum Begriff: Im weiteren Verlauf dieses Dokuments wird davon ausgegangen, dass die Ausfallraten über die Zeit konstant sind. (Diese Hypothese ist für elektronische Systeme allgemein anerkannt).

Anmerkung 2 zum Begriff: Die Ausfallrate λ eines Systems wird auf der Grundlage der Ausfallraten der Menge von Ausfällen, die für dieses System ermittelt wird, bewertet. Die Systemausfallrate ist gleich der Summe der Ausfallraten für die Ausfälle, die für dieses System ermittelt werden (mit der Hypothese der konstanten Ausfallrate).

Anmerkung 3 zum Begriff: Die Systemausfallrate gilt nur für die immanente Ausfälle des betrachteten Systems.

3.1.22

technische Effizienz

Messung der Betriebseffizienz bezogen auf die für die Lösung notwendigen technischen Ressourcen; sie ist eine Entwurfsmetrik

3.1.23

Prüfergebnis

Abbild des Vorhandenseins oder Fehlens eines Anzeichens; das Ergebnis kann den Wert OK oder NOK annehmen

3.1.24

Fehlersuche

Prozess der Ausfalleingrenzung

3.2 Abkürzungen

Am Ende des Dokuments (Anhang C) ist eine Indextabelle angegeben. Mit dieser können die Definitionen der wichtigsten in diesem Dokument verwendeten Begriffe gefunden werden.

Tabelle 1 erläutert die Abkürzungen.

Tabelle 1 — Erläuterung der Abkürzungen

Abkürzung	Erläuterung
<i>BIT</i>	internes Prüfverfahren (en: built-in test)
<i>C</i>	Vorgabe (en: constraint)
<i>CAS</i>	Besatzungswarnsystem (en: crew alert system)
<i>CBIT</i>	kontinuierliches <i>BIT</i> (en: continuous <i>BIT</i>)
<i>CO</i>	einwandfreier Betrieb (en: correct operation)
<i>COR</i>	Rate des einwandfreien Betriebs (en: correct operation rate)
<i>COTS</i>	handelsübliches Produkt (en: commercial off-the-shelf)
<i>DBIT</i>	angefordertes <i>BIT</i> (en: demanded <i>BIT</i>)
<i>DF</i>	erkennbare Ausfälle (en: detectable failures)
<i>DFE</i>	erkennbare Ausfälle mit Auswirkung (en: detectable failures with effect)
<i>DV</i>	Diagnosewert (en: diagnostic value)
<i>E</i>	Menge von Ausfällen (en: set of failures)
<i>EBIT</i>	externes <i>BIT</i> (en: external <i>BIT</i>)
<i>F</i>	Ausfall (en: failure)
<i>FAR</i>	Fehlalarmrate (en: false alarm rate)
<i>FCOR</i>	Fehlerrate bei Betriebsausfall (en: false correct operation rate)
<i>FDC</i>	Fähigkeit der Ausfallerkennung (en: failure detection capability)
<i>FDP</i>	Wahrscheinlichkeit der Ausfallerkennung (en: failure detection probability)
<i>FE</i>	Ausfälle mit Auswirkung (en: failures with effect)
<i>FIP</i>	Wahrscheinlichkeit der Ausfalleingrenzung (en: failure isolation probability)
<i>FM</i>	Ausfallart (en: failure mode)
<i>FMECA</i>	Fehlermöglichkeits-, Einfluss- und Kritikalitätsanalyse (en: failure modes, effects and criticality analysis)
<i>FNOK</i>	falsches NOK (en: false NOK)

Abkürzung	Erläuterung
<i>FOK</i>	falsches OK (en: false OK)
<i>FRP</i>	Wahrscheinlichkeit der Ausfallauflösung (en: failure resolution probability)
<i>HF</i>	Hardwareausfälle (en: hardware failures)
<i>IBIT</i>	eingeleitetes <i>BIT</i> (en: initiated <i>BIT</i>)
<i>LRU</i>	am Produkt direkt wechselbare Einheit (en: line replaceable unit)
<i>MBIT</i>	Instandhaltungs- <i>BIT</i> (en: maintenance <i>BIT</i>)
<i>MC</i>	Herstellercode (en: manufacturer code)
<i>NFF</i>	kein Fehler festgestellt (en: no fault found)
<i>OBIT</i>	betriebliches <i>BIT</i> (en: operational <i>BIT</i>)
<i>PBIT</i>	Hochlauf- <i>BIT</i> (en: power-up <i>BIT</i>)
<i>SBIT</i>	Anlauf- <i>BIT</i> (en: start-up <i>BIT</i>)
<i>SDF</i>	Software-Entwurfsfehler (en: software design fault)
<i>SE</i>	Systemengineering
<i>Sn</i>	Empfindlichkeit (en: sensitivity)
<i>Sp</i>	Spezifität (en: specificity)
<i>SRU</i>	am Produkt nicht direkt wechselbare Einheit (en: shop replaceable unit)
<i>TNOK</i>	wahres NOK (en: true NOK)
<i>TOK</i>	wahres OK (en: true OK)
<i>UR</i>	Steigerungsrate der Unzuverlässigkeit (en: unreliabilisation rate)

4 *BIT*-Beteiligte

4.1 *BIT*-Spezifizierer

Der *BIT*-Spezifizierer ist verantwortlich für die Definition der Systemanforderungen in Bezug auf die Fähigkeit, Ausfälle zu erkennen und einzugrenzen, unter Berücksichtigung von

- Kundenanforderungen an ein Projekt,
- Anforderungen von anderen Beteiligten als dem Kunden und
- technischen Anforderungen, die aus internen Vorschriften, aus dem Entwurf und/oder dem Erfahrungsrückfluss resultieren.

Er sendet diese Anforderungen mittels technischer Spezifikationen an Entwickler (einschließlich Unterauftragnehmer und Vertragspartner).

4.2 *BIT*-Entwerfer/Entwickler

Das Entwerfer-/Entwicklerteam ist für den Entwurf und die Entwicklung des *BIT* für das System (für die Hardware- und Softwareelemente) nach den Festlegungen verantwortlich.

Es ist auch für die Begründung des Inhalts der *BIT*-Anforderungen verantwortlich.

4.3 Betrieblicher Nutzer

Der betriebliche Nutzer der obersten Ebene des Systems möchte sicherstellen, dass die Betriebsfunktionen verfügbar sind (Nennzustand), und er möchte vor dem Beginn, aber auch während einer Mission über eine Störung (verschlechterte Situation) informiert werden, um entsprechend zu reagieren.

Er ist über die Alarmfunktion in das *BIT* eingebunden.

4.4 Instandhaltungstechniker

Der Instandhaltungstechniker ist für die Betreuung des Systems verantwortlich.

Unter anderem führt er die aus den *BIT*-Ergebnissen resultierende Instandsetzung durch, um die Leistungsfähigkeit des Systems wiederherzustellen.

Er muss über die notwendigen Informationen verfügen, um die ausgefallenen Elemente einzugrenzen, die er dann austauschen/reparieren muss.

In einigen Fällen dürfen ihm zusätzliche Prognoseinformationen (die aus *BIT*-Ergebnissen stammen dürfen) zur Vorbereitung/Durchführung der vorausschauenden Instandhaltung zur Verfügung gestellt werden, um die Systemverfügbarkeit zu verbessern.

4.5 Technischer Leiter des Systems

Die beiden Hauptaufgaben des technischen Leiters des Systems sind

- sicherzustellen, dass Prüfbarkeitsanforderungen erfüllt werden (Erkennung, Eingrenzung und Rate der falschen Außerbetriebnahme, Eingrenzungs- und Prüfdauer usw.) und
- sicherzustellen, dass diese Anforderungen berücksichtigt werden (Erkennung/Eingrenzung von Ausfällen auf den verschiedenen Instandhaltungsebenen).

Zu diesem Zweck muss er Vorhersageanalysen und Prüfbarkeitsnachweise durchführen, um sicherzustellen, dass die Anforderungen erfüllt werden.

4.6 Fachkraft

Die Fachkraft ist für die Erkennung und Eingrenzung der Ausfälle eines Systems/Ausrüstungselements bis zur untersten Ebene des Instandhaltungskonzepts verantwortlich, falls dies mit den internen Prüfverfahren nicht möglich war.

Dafür nutzt sie das Ergebnis des internen Prüfverfahrens (Spuren, Ereignisse, Zustände usw.).

Aus diesem Grund ist sie an der Leistung des internen Prüfverfahrens sehr interessiert und darf daher einer der Spezifizierer sein.

4.7 Einsatzdatentechniker

Der Einsatzdatentechniker ist verantwortlich für die Analyse aller Vorfälle auf der Grundlage von Informationen, die während der Betriebsphase über das System/Ausrüstungselement erfasst werden. Dies geschieht zur Messung der Kennwerte (Zuverlässigkeit, Verfügbarkeit usw.) oder im Rahmen eines kontinuierlichen Verbesserungsprozesses. Dazu muss der Einsatzdatentechniker die ausgefallenen Elemente, den Zusammenhang und die vom Instandhaltungstechniker durchgeführten Maßnahmen kennen.

Daher ist er an den Informationen interessiert, die in den Prüfergebnissen enthalten sind (Prüfung OK, Prüfung NOK, Zusammenhang, Konfiguration, Datierung usw.).

5 Systemvorgaben

5.1 Systementwurf

In diesem Dokument wird das „System“ als ein „technisches System“ betrachtet, in dem Menschen und Organisationen nicht berücksichtigt werden.

Ein System ist eine Menge von Komponenten, die miteinander und mit der Umgebung in Wechselwirkung treten, um einen Dienst zu erbringen [7].

Ein System wird innerhalb eines als Systemengineering (SE) bezeichneten Prozesses entworfen, der die Komponenten und ihre Schnittstellen sowie den Lebenszyklus des Systems von der anfänglichen Definition des Problems bis zur endgültigen Außerbetriebnahme berücksichtigt [8].

Die Prozesse des Systementwurfs umfassen Definition, Verifizierung, Validierung, Produktion, Bereitstellung, Betrieb, Instandhaltung und Außerbetriebnahme. Dazu gehören auch die für diese Prozesse notwendigen unterstützenden Elemente (Prüfeinrichtungen, Simulatoren usw.) [1].

Ein System wird mit Grundkomponenten entworfen, die oft von verschiedenen Lieferanten entwickelt werden, die am globalen SE-Prozess beteiligt sind.

Eine neue Grundkomponente muss so entwickelt werden, dass sie in das System, für das sie ursprünglich entworfen wurde, integriert werden kann. Sie muss aber auch bestimmte Vorgaben für die Anpassungsfähigkeit (Wiederverwendbarkeit usw.) erfüllen, damit sie in anderen Systemen (Portfolio des Lieferanten) verwendet werden kann.

Darüber hinaus darf die Akquisitions-Strategie des Hauptauftragnehmers, des Eigentümers oder des Kunden für die Einrichtung, die während des gesamten Lebenszyklus des Systems gilt, die Integration von handelsüblichen Produktkomponenten (COTS) umfassen, die zur Senkung der Entwicklungs- und Instandhaltungskosten angepasst wurden.

Der Top-down-Prozess (vom System zu den Grundkomponenten) wird mit dem Bottom-up-Prozess kombiniert, um bereits vorhandene Komponenten (COTS) oder Technologien während des Systemengineerings in den Entwurf eines Systems zu integrieren [9].

Der gesamte Entwurfsprozess des Systemengineerings bezieht sich auf die Prüfbarkeit eines Systems und damit auf den Entwurf des *BIT* des Systems.

5.2 *BIT*-Schnittstellenfunktion

ANMERKUNG In diesem Abschnitt werden die Funktionen beschrieben, die die *BIT*-Ergebnisse nutzen.

5.2.1 Die Alarmfunktion

Ziel der Alarmfunktion eines Systems ist es, den Systembenutzer über Ereignisse, die den Betrieb des Systems stören, und über den Zusammenhang ihres Auftretens zu informieren.

Informationen, die für die Alarmfunktion nützlich sind, vermitteln entweder den Zustand des Systems (Defekt) oder die Überwachung der Einsatzbedingungen (z. B. für ein Luftfahrzeug: Fahrwerk in der Anflugphase oben, Ausklappen bei zu hoher Geschwindigkeit, Unvereinbarkeit der erhaltenen Befehle, externes Ereignis usw.).

Die Alarme im Zusammenhang mit Ausfällen von Systemkomponenten stammen aus der Verarbeitung von *BIT*-Ergebnissen.

Die Alarmfunktion filtert Daten aus den Überwachungssystemen und verarbeitet die Dringlichkeitsstufe für die Darstellung von visuellen (Piktogramme, Parameterwerte usw.) und akustischen Hinweisen für den Nutzer sowie Empfehlungen für die zu ergreifenden Maßnahmen (Meldungen, Prüfliste usw.).

In der zivilen Luftfahrt werden z. B. Informationen von Computern zur Überwachung von Motor- und Systemparametern (*CAS*: Besatzungswarnsystem) verwendet, um mindestens zwei Ebenen der Verarbeitungsdringlichkeit (siehe [4]) auf den elektronischen Anzeigesystemen anzuzeigen:

- Meldungen über kleinere Fehlfunktionen, die während des restlichen Flugs zu berücksichtigen sind;
- größere Defekte oder Alarme, die von der Besatzung dringend zu verarbeiten sind.

Die Dringlichkeitsstufen der Verarbeitung vermitteln eine Stufe der Kritikalität des Nebenalarms oder Hauptalarms, die jeweils Folgendes verursacht:

- eine vernachlässigbare Störung der ausgeführten Funktion, des ausgeführten Diensts oder der durchgeführten Mission, die Ursache von Behinderungen für den Nutzer oder von Redundanzverlusten eines nicht lebenswichtigen Systems sein könnte;
- eine größere Störung der ausgeführten Funktion, des ausgeführten Diensts oder der durchgeführten Mission, die Ursache für den Verlust einer kritischen Funktion oder der Redundanz eines lebenswichtigen Systems sein wird.

In bestimmten Systemen kann ein zusätzlicher Alarm einer niedrigeren Dringlichkeitsstufe bestehen, der keinen Einfluss auf den Betrieb hat. Dieser Fall wird als „Information“ bezeichnet.

Die Alarmfunktion darf nicht die Betriebsfunktionen des Systems stören. Sie muss die Aufmerksamkeit des Nutzers zum richtigen Zeitpunkt und nach der entsprechenden Dringlichkeitsstufe auf sich ziehen. Die Zeit für die Anzeige von Informationen kann auch auf der Grundlage der Alarmstufe geregelt werden [5].

Die Alarmbestätigungs- und -löschungsmechanismen werden auch von der Alarmfunktion verarbeitet.

Während des Betriebs ist die Alarmfunktion eine Eingabe für die Systemfunktion „Verwaltung des eingeschränkten Betriebs“.

Die Ereignisse (mit den Umständen ihres Auftretens) werden außerdem in „Aktivitätslisten“ aufgezeichnet. Sie stellen dann die Funktionen der Vorfallsanalyse mit „praktischen Erfahrungen“ bereit, die nach der betrieblichen Mission schrittweise verarbeitet werden.

5.2.2 Die Diagnosefunktion

Das Ziel der Diagnosefunktion eines Systems ist es, den Systemausfallstatus und im Fall eines Ausfallzustands die Ursache (oder die Ursachen) des Ausfalls zu bestimmen.

Die Auswertung eines Datensatzes und von wahrnehmbaren Hinweisen anhand einer logischen Beweisführung ermöglicht die Ausführung einer Diagnose (siehe Definition in Abschnitt 3).

Mit Diagnoseverfahren (Erkennungsalgorithmen, Fehlerbäume, Prüfungen durch den Nutzer) kann der Ausfall erkannt, bestimmt und eingegrenzt werden.

BIT-Ergebnisse sind Eingaben in den Prozess zur Durchführung der Diagnose.

Die Diagnosefunktion erfolgt während des Einsatzes (Überwachung des Systemzustands), der Instandhaltung (für die Reparatur) und der Fertigung (Kontrolle an den Fertigungslinien).

Die Diagnosefunktion nutzt Daten von den Überwachungssystemen, um einen vertretbaren Kompromiss zwischen der Fähigkeit der Fehlererkennung und der Begrenzung von Fehlalarmen zu finden.

5.2.3 Integrierte Rekonfiguration

Das Ziel der integrierten Rekonfigurationsfunktion eines Systems ist es, das System in die Lage zu versetzen, seine betriebliche Mission auch bei einem Ausfall fortzusetzen.

Während des Betriebs des Systems nutzt sie vor allem *BIT*-Informationen, um die „eingeschränkten“ Betriebszustände des Systems unter Berücksichtigung der Umstände ihres Auftretens zu verwalten.

Die integrierte Rekonfigurationsfunktion konfiguriert dann die Hardware und/oder Software des Systems neu und hebt dabei die Redundanzen und Toleranzmechanismen für die während des Entwurfs vorhergesagten Ausfälle hervor.

Sämtliche Systemfunktionen werden nicht mehr „nominell“ ausgeführt, sondern es stehen nur noch diejenigen zur Verfügung, die für die Fortführung der grundlegendsten Ziele der Mission wesentlich sind.

5.2.4 Die Instandhaltungsfunktion

Das Ziel der Instandhaltungsfunktion eines Systems ist die Wiederherstellung der Betriebsverfügbarkeit des Systems.

Die Instandhaltungsfunktion umfasst die Instandsetzung und die vorbeugende Instandhaltung auf verschiedenen Ebenen des Instandhaltungseingriffs.

Auf der ersten Eingriffsebene ermöglicht die Instandhaltungsfunktion dem Instandhaltungstechniker den Austausch von „am Produkt direkt wechselbaren“ Elementen (*LRU* — en: *Line replaceable unit*) im System, die von den integrierten Überwachungssystemen als ausgefallen erklärt wurden.

Auf der zweiten Ebene ermöglicht die Instandhaltungsfunktion dem Instandhaltungstechniker den Austausch von „am Produkt nicht direkt wechselbaren“ Elementen (*SRU* — en: *Shop replaceable unit*) im System.

Die Instandhaltungsfunktion verwendet die Ergebnisse der *BIT* und anderer Bedienerprüfungen, die in der Regel mit Einrichtungen außerhalb des Systems (Prüfeinrichtungen usw.) durchgeführt werden. Die technische Instandhaltungsdokumentation (vor allem Fehlerbäume) wird unter anderem zur Auswertung der Prüfergebnisse und für die durchzuführenden Korrekturmaßnahmen (Fehlersuche) erstellt.

5.2.5 Die Datenaufzeichnung für die Nachanalysefunktion

Das Ziel der Datenaufzeichnung eines Systems für eine Nachanalysefunktion ist es, während des Systembetriebs Informationen in einen Speicher zu schreiben, die zu einem späteren Zeitpunkt verwendet werden können.

Diese Informationen können sich auf die während der Mission aufgetretenen Ausfälle beziehen und auf den Nutzungszusammenhang des Systems, in dem die Ausfälle aufgetreten sind. In dieser Art von Informationen sind die *BIT*-Ergebnisse enthalten.

Die Datenaufzeichnung für die Nachanalysefunktion kann auch Informationen über den Einsatz oder den Zustand des Systems und seiner Einrichtungen in den Speicher schreiben.

Alle diese Informationen können offline von verschiedenen Nutzerkategorien angewendet werden, wie von

- Instandhaltungspersonal am Boden für die Reparatur des Systems oder Ausrüstungselements,
- Teams für die Verarbeitung technischer Ereignisse zur Beurteilung spezifischer oder wiederkehrender Ausfälle und Vorfälle,
- Flottenmanagern zur Verbesserung der Instandhaltungspläne und
- industriellen Entwicklungsteams zur Verbesserung des zukünftigen Entwurfs oder für Instandhaltungsarbeiten, die am Systemdurchzuführen sind.

5.3 Technische Systemzustände

Die technischen Zustände eines Systems beschreiben dessen Fähigkeit, die für die Mission geforderten Funktionen zu erfüllen oder nicht zu erfüllen. Sie beeinflussen die mögliche Nutzung eines Systems.

Wenn es in Betrieb ist, kann ein System

- verfügbar (nicht ausgefallen);
- betriebsbereit (verfügbar und für die Nutzung konfiguriert);
- eingeschränkt nutzbar (bestimmte Funktionen nicht verfügbar); oder
- nicht verfügbar (ausgefallen)

sein.

Die Konfiguration eines Militärflugzeugs für den Einsatz umfasst z. B. (unter anderem) die Versorgung mit Treibstoff und den Einbau von Waffensystemen.

5.4 Funktionsarten eines Systems

Die Funktionsarten eines Systems beschreiben die Nutzungsphasen des Systems, die umgesetzt werden, um den geforderten Dienst zu erhalten. Jede Funktionsart kann mehrere Unterfunktionsarten oder Nutzungsphasen haben [10].

Die folgenden Hauptfunktionsarten werden angegeben:

- Stillsetzung;
- Abschaltung/Initialisierung (Anlauf);
- Betrieb;
- Instandhaltung;
- Simulation/Schulung.

Die Verwendung der verschiedenen Arten von *BIT* in den verschiedenen Funktionsarten wird in Abschnitt 6.2.6 beschrieben.

5.5 Systemkonfiguration

5.5.1 Betriebliche Systemkonfiguration

Die betriebliche Konfiguration besteht darin, das System mit den für die kommende Mission erforderlichen Ressourcen auszustatten. Sie wird durch den Nutzer des Systems durchgeführt.

Zur Veranschaulichung sind die Waffensysteme, der Flugplan und die Vorschriften für Störsendungen für den Selbstschutz Entscheidungen der betrieblichen Konfiguration.

ANMERKUNG Das konfigurierte System stellt mitunter selbst ein System dar, wenn die externe Ressourcen mit dem Basissystem in Wechselwirkung treten. Zum Beispiel:

- das Auftanken ist Teil der Konfiguration, schafft aber keine Systeme höherer Ebenen, da es sich dabei um das Zusammenfügen von Komponenten handelt, die Dienste nur als Ganzes erbringen können;
- der Einbau einer Waffe in ein Luftfahrzeug schafft ein System einer höheren Ebene, weil mindestens eine der Komponenten, und zwar das Luftfahrzeug selbst, ein System ist, da es allein bereits einen Betriebsdienst (den Flug) erbringt.

Die Änderung von Konfigurationen erfolgt dynamisch, wenn sie den Betriebsdienst nicht unterbricht.

5.5.2 Technische Konfiguration

Die technische Konfiguration besteht darin, das System mit den für die globale Funktion erforderlichen Ressourcen auszustatten. Sie wird vom Hersteller ausgeführt, der das System liefert.

5.5.3 Parametrierung des *BIT*

An dieser Stelle wird ein weiterer Aspekt der Systemkonfiguration eingeführt, und zwar die Parametrierung des *BIT*: dabei handelt es sich um eine technische Konfiguration, die vom Entwickler entschieden und vom Bediener auf Anweisung des Entwicklers umgesetzt wird. Die Konfiguration der *BIT*-Funktion ist eine vom Entwickler validierte Fähigkeit.

Die *BIT*-Funktion darf aus verschiedenen Gründen aktualisiert werden, wie

- unter Berücksichtigung von Felddaten (praktische Erfahrungen) zur Verbesserung oder Korrektur der *BIT*-Funktion;
- Ersatz einer Systemkomponente durch eine andere, die auf der funktionalen Ebene gleichwertig ist. Es gibt zwei Prozesse zur Aktualisierung der *BIT*-Funktion:
 - Parametrierung oder allgemeiner die technische Konfiguration, wenn diese Parametrierung möglich ist;
 - andernfalls die Notwendigkeit, einen neuen Standard des Systems nach einer möglichen weiteren Entwicklung und Integration zu validieren.

Die Parametrierung ist ein Element der technischen Konfiguration.

Empfehlung 1: Der Entwickler sollte das System mit der Parametrierfähigkeit für die *BIT*-Funktion mit den folgenden Vorgaben ausrüsten:

- die parametrierbaren Elemente des *BIT* dürfen weder eine Auswirkung auf das Betriebsverhalten des Systems noch auf dessen Sicherheit haben;
- die Änderung der Parameterwerte darf keine erneute Validierung der Software erfordern.

Lediglich eine Überprüfung, ob die Parameterwerte einwandfrei aktualisiert wurden, kann erforderlich sein.

Im Fall eines Luftfahrzeugs geht es vor allem darum, jegliche Konsequenzen bezüglich der Zertifizierung zu vermeiden.

Empfehlung 2: Der Systementwickler muss die sicherheitsbezogenen und die nicht sicherheitsbezogenen Einstellungen der *BIT*-Funktion unabhängig voneinander verwalten (Trennungsprinzip).

6 Arten und Metriken des *BIT*

6.1 Allgemeines

Ein internes Prüfverfahren (*BIT*) ist eine Prüfung, die ausschließlich mit den für ein Ausrüstungselement bzw. System spezifischen Hardware- und Software-Ressourcen durchgeführt wird, um diese und/oder deren Unterbaugruppen zu prüfen, um Ausfälle zu erkennen und diese einzugrenzen oder sogar zu bestimmen.

6.2 Die verschiedenen Arten von *BIT*

ANMERKUNG Dieser Abschnitt enthält eine Definition für die verschiedenen Arten von *BIT*.

6.2.1 Hochlauf-*BIT* oder Einschalt-*BIT* (*PBIT* — en: Power-on/up *BIT*)

Das *PBIT* wird während der Initialisierung oder Einschaltung ausgeführt. Im Allgemeinen ist es eine störende Prüfung beim Einschalten.

Sie wird mitunter als Inbetriebnahme-*BIT* (*SBIT* — en: start-up *BIT*) bezeichnet.

Bei dieser Art der Prüfung wird überprüft, ob die Systeme betriebsbereit sind.

ANMERKUNG Ein „Kaltstart-*PBIT*“ entspricht einem vollständigen Prüfverfahren, das beim Einschalten des Systems während der Systemhochlaufphase durchgeführt wird. Tritt während einer Mission an einem Ausrüstungselement ein Ausfall oder ein anomales Verhalten auf, wird dieses Ausrüstungselement während der betrieblichen Mission neu gestartet. Dann wird ein verkürztes *PBIT*, das als „Warmstart-*PBIT*“ bezeichnet wird, ausgeführt. Dieses Verfahren wird verwendet, um die Hauptfunktionen für eine möglichst schnelle Rückkehr zu den Betriebsbedingungen schnell zu prüfen.

6.2.2 Eingeleitetes *BIT* (*IBIT* — en: Initiated *BIT*) oder angefordertes *BIT* (*DBIT* — en: Demanded *BIT*)

Diese Prüfung wird in das System integriert. Sie kann „manuell“ auf Veranlassung des Bedieners zur Verbesserung der Ausfalleingrenzung oder durch die *BIT*-Überwachungsfunktion des Systems ausgelöst werden (siehe Bild 3).

Sie darf die Prüfungen anderer *BIT*-Arten wiederholen.

Das *IBIT* ist störend. Die geprüfte Funktion ist während der gesamten Prüfdauer nicht betriebsbereit, liefert aber ein genaueres Ergebnis als das *CBIT* (siehe folgender Abschnitt).

Obwohl sie während bestimmter Betriebsperioden vom System ausgelöst werden kann, darf sie für die laufende Mission nicht störend sein.

Empfehlung 3: Die Bezeichnung „Angeordnetes *BIT*“ (en: commanded *BIT*), die mitunter in der Literatur zu finden ist, ist unzulässig, um Verwechslungen mit dem Begriff „*CBIT*“ (en: Continuous *BIT*) zu vermeiden, der für „Kontinuierliches *BIT*“ reserviert werden muss.

6.2.3 Kontinuierliches *BIT* (*CBIT* — en: Continuous *BIT*)

Das *CBIT* überwacht das System kontinuierlich im Betriebszustand. Es darf folglich nicht störend sein.

Das *CBIT* darf Informationen zur Systemintegrität oder zum Leistungsniveau zur Verfügung stellen. Seine Ergebnisse können in Echtzeit (siehe Abschnitt 5.2.1 „Die Alarmfunktion“) oder nach Speicherung im Speicher zeitversetzt verwendet werden.

In der Regel handelt es sich dabei um eine periodische oder zyklische Prüfung.

Diese Prüfung wird mitunter als betriebliches *BIT* (*OBIT* — en: Operational *BIT*) oder kontinuierliche Überwachung (*CM* — en: continuous monitoring) bezeichnet.

6.2.4 Externes *BIT* (*EBIT* — en: External *BIT*)

Der *EBIT*-Auslösemechanismus erfolgt außerhalb des Systems, aber die Prüfung selbst erfordert keine weiteren äußeren Stimuli. Der Speichermechanismus ist in das System integriert.

Der Unterschied zwischen dem *IBIT* und dem *EBIT* ist die Eingrenzung der Ressource, die die Prüfung auslöst. Im Fall des *IBIT* ist diese Ressource Teil des zu prüfenden Systems (z. B. Drucktaster im Luftfahrzeug). Im Fall des *EBIT* befindet sich die Ressource, die die Prüfung auslöst, jedoch außerhalb des Systems (z. B. ein mit dem Luftfahrzeug verbundener Computer).

Der Vorteil des *EBIT* gegenüber des *IBIT* ist die Tatsache, dass die Prüfergebnisse in einigen Fällen von einem Gerät außerhalb des Systems abgerufen werden können.

Ein *EBIT* wird während der Phasen zur Eingrenzung der ausgefallenen Elemente auf verschiedenen hierarchischen Strukturebenen des zu prüfenden Systems durchgeführt.

Das *EBIT* ist besonders für die ferngesteuerte Instandhaltung geeignet.

6.2.5 Instandhaltungs-*BIT* (*MBIT* — en: Maintenance *BIT*)

Das *MBIT* wurde vor kurzem als Werkzeug zur Unterstützung der Bediener während der Instandhaltungsphasen eingeführt.

Es wird während der Phasen der Ausfalleingrenzung angewendet. Es leitet das Instandhaltungspersonal durch die Reparaturfolge. Außerdem kann es zur Überprüfung der Integrität eines Systems nach einer Reparatur verwendet werden.

Es besteht oft aus einer Kombination mehrerer *BIT*-Arten (*IBIT* oder *EBIT*). Das *MBIT* wird im Allgemeinen mit externen Ressourcen durchgeführt, die das zu prüfende System stimulieren.

Diese Prüfung ist im Allgemeinen störend.

6.2.6 Zusammenfassung der Eigenschaften der verschiedenen Arten von BIT

Tabelle 2 — Zusammenfassung der Eigenschaften der verschiedenen Arten von BIT

BIT-Art	Auslösung	Betriebszustand	Einfluss auf den Betrieb	Ressourcen der Realisierung	Bemerkungen
PBIT	Intern	Einschalten (Kaltstart) Neuinitialisierung (Warmstart)	Störend	Interne Ressourcen	Interne Auslösung während der Einschaltung und externe Auslösung während der Neuinitialisierung
IBIT (DBIT)	Bediener oder internes (durch die BIT-System-Supervisor-Funktion) System	Betrieb oder Instandhaltung	Störend	Interne Ressourcen	Sie darf nicht störend sein, wenn sie in bestimmten Betriebsphasen durch ein Supervisor-BIT ausgelöst wird.
CBIT	Intern	Betrieb	Nicht störend	Interne Ressourcen	Kontinuierliche Prüfung (periodisch/zyklisch)
EBIT	Außerhalb des Systems	Instandhaltung	Störend	Interne Ressourcen, aber Auslösung durch externe Ressourcen	Externe Auslösung Wird für die ferngesteuerte Instandhaltung verwendet
MBIT	Außerhalb oder innerhalb des Systems	Instandhaltung	Störend	Interne oder externe Ressourcen	Wird für Eingrenzung und Reparatur verwendet Wird zur Bestätigung der Integrität nach Reparatur verwendet

ANMERKUNG In Tabelle 2 wird die Funktionsart „Simulation/Schulung“ in den Betriebszustand „Betrieb“ eingeordnet.

6.3 Die Metriken

6.3.1 Rolle der mathematischen Definitionen der Metriken

Nachstehend werden die Definitionen der Metriken für die Prüfleistung aufgeführt. Diese Definitionen werden durch eine mathematische Gleichung ergänzt, um sicherzustellen, dass sie richtig nachvollzogen werden können. Unter keinen Umständen dürfen diese Gleichungen als Empfehlung für die Nachweisverfahren oder die Begründung der Metriken interpretiert werden. Die Vielfalt der Lösungen, ihre Anwendbarkeit in den verschiedenen Zusammenhängen oder ganz einfach die Tatsache, dass sich die Technologien und Verfahren ständig verändern, machen solche Empfehlungen unmöglich.

Es unterliegt den Herstellern, sich unter Berücksichtigung der Besonderheiten ihrer Probleme auf geeignete Verfahren zu einigen, sei es durch Analyse, Begutachtung, Vergleich, Laborprüfungen usw. oder durch Kombinationen dieser Verfahren.

Der wichtigste Aspekt für die Arbeit ist, dass eine eindeutige Semantik für die Metriken zur Verfügung steht. Nur diese Semantik wird als ausreichend tragfähig für die Aufnahme in dieses Dokument angesehen.

6.3.2 Erkennungsrate

6.3.2.1 Allgemeines

Die Erkennungsraten geben das Verhältnis zwischen der Anzahl der Ausfälle, die durch eine Reihe von Prüfungen erkennbar sind, und der Gesamtzahl der ermittelten Systemausfälle an (Liste der Systemausfälle: siehe Definition in Abschnitt 3). Bei der Berechnung darf eine Gewichtung erfolgen, z. B. nach Ausfallrate und/oder nach Kritikalität.

Empfehlung 6: Je höher die Kritikalität der Ausfälle ist, umso mehr sollten diese Ausfälle geprüft werden.

Es wird empfohlen, für eine angegebene Reihe von Prüfungen die Erkennungsraten für verschiedene Anwendungen und/oder für verschiedene Ausfallauswirkungen zu berechnen. Tatsächlich spiegeln die Erkennungsraten insbesondere die Abdeckung der Ausfälle wider, die abhängig davon als wichtig angesehen werden, ob die Reihe von Prüfungen vom Alarmsystem (eine Rate je Kritikalität des Ausfalls, die zu folgenden Arten von Alarmen führt: Hauptalarm, Nebalarm, Information), vom Instandhaltungstechniker oder von anderen Nutzern verwendet wird.

Abhängig davon, ob ein Interesse an den Ausfällen besteht, die sich auf den Betrieb des Systems auswirken oder nicht, wird die Berechnung auf die jeweilige Referenzausfallmenge gestützt.

Empfehlung 7: Die Hypothese, die zur Berechnung der Erkennungsrate verwendet wird (welche ist die jeweilige Ausfallmenge?), muss eindeutig dargelegt werden.

6.3.2.2 Fähigkeit der Ausfallerkennung (*FDC*, en: failure detection capability)

Die Fähigkeit der Ausfallerkennung für eine Reihe von Prüfungen ist das Verhältnis zwischen der Anzahl der Ausfälle, die durch diese Reihe von Prüfungen erkannt werden, und der Gesamtzahl der ermittelten Systemausfälle.

Allgemeiner Fall:

$$FDC = \frac{\text{Card}(DF)}{\text{Card}(HF)} \quad (1)$$

Fall, der auf Ausfälle mit einer Auswirkung begrenzt ist, die als „zu berücksichtigen“ gilt:

$$FDC_{FE} = \frac{\text{Card}(DF \cap FE)}{\text{Card}(FE)} \quad (2)$$

Ein Beispiel für die Berechnung dieser Metrik ist in A.2 angegeben.

6.3.2.3 Wahrscheinlichkeit der Ausfallerkennung (*FDP*, en: failure detection probability)

Die Wahrscheinlichkeit der Ausfallerkennung für eine Reihe von Prüfungen ist das Verhältnis zwischen der Summe der Ausfallraten von Ausfällen, die durch diese Reihe von Prüfungen erkannt werden, und der Summe der Ausfallraten der Ausfallmenge, die durch das System festgestellt werden.

Allgemeiner Fall:

$$FDP = \frac{\sum_{f_i \in DF} \lambda_i}{\sum_{f_i \in HF} \lambda_i} \quad (3)$$

Fall, der auf Ausfälle mit einer Auswirkung begrenzt ist, die als „zu berücksichtigen“ gilt;

$$FDP_{FE} = \frac{\sum_{f_i \in DF \cap FE} \lambda_i}{\sum_{f_i \in FE} \lambda_i} \quad (4)$$

Ein Beispiel für die Berechnung dieser Metrik ist in Anhang A angegeben.

6.3.2.4 Bemerkungen

Bei unterschiedlichen Ausfallraten sichert eine hohe *FDP* nicht eine gute Erkennungsleistung, d. h. die Fähigkeit, eine große Anzahl möglicher Ausfälle zu erkennen.

Empfehlung 8: Zur Berechnung der *FDP* sollten bei unterschiedlichen Ausfallraten die folgenden Bereiche untersucht werden, um die Metriken an die Anforderung/das System anzupassen:

- Vergleich von *FDP* und *FDC* für eine angegebene Reihe von Prüfungen; wenn die *FDP*- und *FDC*-Werte voneinander abweichen, müssen die Ausfallraten, die zu einer hohen *FDP* geführt haben, analysiert werden;
- Einteilung der Ausfälle bezüglich ihrer Ausfallrate und Berechnung der Erkennungsrate für jede der erhaltenen Kategorien;
- Berechnung einer neuen Erkennungsrate, bei der die Gewichtung eher die Kritikalität als die Ausfallrate sein könnte;
- usw.

BEISPIEL

Es wird ein System mit 2 Ausfällen A und B (Elemente der Menge *FE*) mit einer Auswirkung, die als „zu berücksichtigen“ gilt, betrachtet.

Die Ausfallraten sind

- $\lambda_A = 50$ Vorkommen je Stunde für Ausfall A und
- $\lambda_B = 2$ Vorkommen je Stunde für Ausfall B.

Außerdem wird eine Prüfung betrachtet, die Ausfall A abdeckt aber nicht Ausfall B. Das heißt, die Menge *DF* besteht nur aus Ausfall A.

Damit ergibt sich

- $\text{Card}(FE) = 2$ und
- $\text{Card}(DF \cap FE) = 1$.

Dabei ist

$$FDC = \frac{\text{Card}(DF \cap FE)}{\text{Card}(FE)} = \frac{1}{2} = 50 \% \quad (5)$$

und

$$FDP = \frac{\lambda_A}{\lambda_A + \lambda_B} = \frac{50}{50 + 2} = 96,2 \% \quad (6)$$

Bei der gleichen Prüfung weicht die *FDP* erheblich von der *FDC* ab.

6.3.3 Eingrenzungsrate

6.3.3.1 Diagnose und Eingrenzung

Während des Betriebs ist die Diagnose die Entsprechung zwischen den Ergebnissen einer angegebenen Reihe von Prüfungen und einer bekannten Signatur, die ihrerseits einer Mehrdeutigkeitsgruppe zugeordnet ist.

Die Berechnung der Eingrenzungsrate erfolgt unter korrekter Anwendung von Empfehlung 7.

Empfehlung 9: Die Datenbank der Ausfallsignaturen sollte bereits bei der Konzeption der Prüfungen aufgestellt werden.

Empfehlung 10: Die Datenbank der bekannten Signaturen für die Diagnose wird erstellt auf der Grundlage

- der *FMECA* (Fortpflanzung von Auswirkungen);
- der Systemarchitektur;
- der Definition der Prüfungen und
- des Erfahrungsrückflusses.

Die Diagnose ist mehrdeutig, wenn die Mehrdeutigkeitsgruppe größer als 1 ist. Die Fehlereingrenzung (Fehlersuche) beinhaltet die Verringerung der Größe der Mehrdeutigkeitsgruppe durch Beobachtungen oder zusätzliche Prüfungen.

Der Prozess der Ausfalleingrenzung (Fehlersuche) ist iterativ. Er endet, wenn die Diagnose nicht mehr mehrdeutig ist und die Fehlersuche validiert ist.

BEISPIEL

Ein einfaches System setzt sich aus 2 Komponenten zusammen: aus einem Schalter und einer Glühlampe.

Der Ausfall des Schalters wird durch falsches Verhalten der Glühlampe wahrgenommen. Bleibt die Glühlampe eingeschaltet, schließt die Beweisführung auf den Ausfall des Schalters.

Bleibt die Glühlampe ausgeschaltet, dann wird die Beweisführung der Komponente mit der höchsten Ausfallrate den Vorzug geben, und da die Schalter selten ausfallen, belastet die Beweisführung die Glühlampe (vielleicht zu Unrecht).

Nach dem Auswechseln der Glühlampe ist der Ausfall immer noch vorhanden. Die Beweisführung wird auf den Schalter gerichtet. Die Überprüfung des Verhaltens der Glühlampen mit einem anderen Schalter führt zu dem Schluss, dass die Glühlampen einwandfrei funktionieren. Die Beweisführung schließt auf den Ausfall des Schalters als Ausfallursache des Systems.

Dieses Beispiel veranschaulicht,

- dass ein Ausfall durch ein mangelhaftes Verhalten vermittelt wird, das am Ort des Ausfalls möglicherweise nicht wahrnehmbar ist (dies ist einer der Gründe für die Fehlersuche) und
- dass die Sequenzierung oder der Kreuzvergleich von Prüfungen dazu beiträgt, den Ausfall einzugrenzen.

Die Grundsätze der Fehlersuche hängen von den Umständen ab. Hier sind einige Darstellungen (die keine Empfehlung darstellen):

- während des Betriebs muss die Fehlersuche versuchen, vorrangig die Ausfälle zu behandeln, die wahrscheinlich schwerwiegende Auswirkungen haben;
- während einer Nicht-Betriebsphase, die einen Neustart an einem bestimmten Datum erfordert, wird die Fehlersuche vorrangig die Behandlung der Ausfälle für eine möglichst schnelle, mit dem Termin zu vereinbarende Wiederaufnahme des Diensts umfassen;
- während einer Nicht-Betriebsphase aber ohne erforderlichen Termin für die Wiederaufnahme des Diensts wird die Fehlersuche nach einer wirtschaftlichen Logik durchgeführt (Kosten für das Ersatzteil, Kosten für die Außerbetriebnahme usw.).

Diese Darstellungen zeigen, dass die Richtlinien auf unterschiedliche Betriebseffizienzen abzielen dürfen: Die ersten beiden Darstellungen berücksichtigen nicht die Verringerung der direkten Instandhaltungskosten.

Empfehlung 11: Der Entwickler oder Bediener muss das Hauptkriterium für die Strategie der Fehlersuche (*LRU*-Zuverlässigkeit, *LRU*-Zugänglichkeit, Schweregrad, Kosten, Dauer der Außerbetriebnahme usw.) im Hinblick auf die Umstände (z. B. für ein Luftfahrzeug: Flugphase, Phase zwischen Flügen, Instandhaltungsphase) festlegen.

6.3.3.2 Wahrscheinlichkeit der Ausfalleingrenzung (*FIP*, en: Failure isolation probability)

FIP_n entspricht der Wahrscheinlichkeit, dass eine Prüffolge den Ausfall auf eine Mehrdeutigkeitsgruppe mit n *LRU* eingrenzt.

Aspekte:

- S_n ist eine Menge von Signaturen mit einem Rang von kleiner oder gleich n ;
- F_s ist die Menge der Ausfälle f_i mit der Signatur s .

Der Ausdruck FIP_n wird deshalb wie folgt formuliert:

$$FIP_n = \frac{\sum_{s \in S_n} \{(\sum_{f_i \in F_s \cap FE} \lambda_i)\}_s}{\sum_{f_i \in DF \cap FE} \lambda_i} \quad (7)$$

Diese Metrik wird im Allgemeinen für $n = 1$, $n = 2$ und $n = 3$ berechnet.

Ein Beispiel für die Berechnung dieser Metrik ist in Abschnitt A.2 angegeben.

6.3.3.3 Wahrscheinlichkeit der Ausfallauflösung (*FRP*, en: Failure resolution probability)

FRP_n entspricht der Wahrscheinlichkeit, den Ausfall durch schrittweisen Austausch von n *LRU* zu beheben. Die Idee dabei ist, unter den belasteten *LRU* zuerst die *LRU* zu ersetzen, deren erkannte Ausfälle die höchste kumulierte Ausfallrate der Mehrdeutigkeitsgruppe aufweist.

Aspekte:

- ein System umfasst K *LRU*;
- T ist die Reihe der Prüfungen t ;
- S_T ist die Menge der Signaturen s , die mit T verknüpft sind;

— $DF_{s,k}$ ist die Menge der Ausfälle f_i der LRU_k mit der Signatur s .

$x_{\text{sup}}(E)$ ist ein in diesem Dokument definierter Operator. Er gibt das x -größte Element der Menge E an.

Der Ausdruck FRP_n wird deshalb wie folgt formuliert:

$$FRP_n = \frac{\sum_{s \in S_T} \left\{ \sum_{x=1}^{x=n} x_{\text{sup}} \left(\left\{ \sum_{f_i \in DF_{s,k} \cap FE} \lambda_i \right\}_k \right) \right\}_s}{\sum_{f_i \in DF \cap FE} \lambda_i} \quad (8)$$

Diese Metrik wird im Allgemeinen für $n = 1$, $n = 2$ und $n = 3$ berechnet.

$\left\{ \sum_{f_i \in DF_{t,k} \cap FE} \lambda_i \right\}_k$ entspricht der List der „Summen der Ausfallraten λ_i der Ausfälle f_i der LRU_k mit der Signatur s “.

Ein Beispiel für die Berechnung dieser Metrik ist in Abschnitt A.2.2.2 angegeben.

6.3.3.4 Wann ist es sinnvoller, *FIP* oder *FRP* zu verwenden?

FIP und *FRP* sind Leistungsmerkmale der Systemarchitektur. Die Betriebseffizienz der Architektur hängt vom Zusammenhang ab. Eine leichte Zugänglichkeit zur *LRU* beeinflusst z. B. die Betriebseffizienz,

- wenn alle *LRU* gleich leicht zugänglich sind: Es ist wichtig, die am wenigsten zuverlässige *LRU* zuerst zu entfernen, da sie am wahrscheinlichsten ausgefallen ist. In diesem Fall ist das Leistungsmerkmal der *FRP* in der Entwurfsphase am wichtigsten für den Vergleich der Architekturen miteinander und
- wenn alle *LRU* verschieden leicht zugänglich sind. In diesem Fall wird diese leichte Zugänglichkeit ein entscheidender Faktor für die Fehlersuche sein, wenn das Instandhaltungskonzept der Außerbetriebnahme der am leichtesten zugänglichen *LRU* den Vorzug gibt. Das Leistungsmerkmal der *FIP* ist daher in der Entwurfsphase am wichtigsten für den Vergleich der Architekturen miteinander, da die Zuverlässigkeit der *LRU* kein urteilsfähiger Faktor für deren Berechnung ist.

Wenn andererseits die Ausfallraten der *LRU* im System gleichwertig sind, ist das Leistungsmerkmal der *FRP* weniger maßgeblich als die *FIP*.

Das Kriterium der Zugänglichkeit stellt selbst nicht das Kriterium der Instandhaltbarkeit dar. Wenn also andere Erwägungen zu berücksichtigen sind (Logistikunterstützung, Verfügbarkeit von Ersatzteilen, Werkzeuge, Kosten usw.) und diese vorherrschen, müssen sie in die Entscheidung zwischen *FIP* und *FRP* einbezogen werden.

Empfehlung 12: Die Art der Eingrenzungswahrscheinlichkeit (z. B. *FRP* oder *FIP*) muss eindeutig dargelegt werden.

6.3.4 Durch das *BIT* verursachte Steigerungsrate der Unzuverlässigkeit

Das Hinzufügen von Komponenten, die nur für die Prüfung in einem System bestimmt sind, führt dazu, dass das System weniger zuverlässig wird (Steigerung der Unzuverlässigkeit des Systems).

Die Steigerungsrate der Unzuverlässigkeit (*UR*, en: Unreliabilisation rate), die durch das *BIT* verursacht wird, ist das Verhältnis zwischen der Summe der Ausfallraten der Komponenten, die in einem System speziell für dessen Prüfung implementiert werden, und der Summe der Ausfallraten aller ermittelten Ausfälle. Die Menge der Ausfälle dieser Komponenten, die speziell in einem System für dessen Prüfung implementiert werden, bildet die Menge C .

Allgemeiner Fall:

$$UR = \frac{\sum_{f_i \in C} \lambda_i}{\sum_{f_i \in FE} \lambda_i} \quad (9)$$

Fall, der auf Ausfälle mit einer Auswirkung, die als „zu berücksichtigen“ gilt, begrenzt ist:

$$UR = \frac{\sum_{f_i \in C \cap FE} \lambda_i}{\sum_{f_i \in FE} \lambda_i} \quad (10)$$

Die Rate darf als ein Schätzwert berechnet werden, indem eine *FMECA* an der *BIT*-Hardware durchgeführt wird.

Empfehlung 13: Der Entwickler muss bei der Dimensionierung der *BIT*-Funktion im System zwischen den Vorteilen und den Nachteilen entscheiden. Die Steigerungsrate der Unzuverlässigkeit sollte für diese Entscheidung genauso berücksichtigt werden wie der Stromverbrauch, das Gewicht, das Volumen usw.

6.3.5 Fehlalarmraten, Fehlerraten bei Betriebsausfall

6.3.5.1 Allgemeines

Die Fehlalarmrate wird definiert durch:

$$FAR = \frac{\text{Anzahl der Fehlalarme}}{\text{Gesamtanzahl der Alarme (wahr oder falsch)}} \quad (11)$$

Dieser Wert kann in der Phase des Systementwurfs noch nicht berechnet werden, da zu diesem Zeitpunkt die Terme der Gleichung nicht bekannt sind.

Es gibt jedoch Metriken, die die Prüfleistungen grundsätzlich beschreiben, das sind die Empfindlichkeit und die Spezifität. Dieses Dokument wird zeigen, dass die Fehlalarmrate von der Empfindlichkeit und der Spezifität der Prüfungen und den Ausfallraten der zu prüfenden Ausrüstungselemente abhängt.

Die Bedeutung dieses Prozesses besteht darin, ein Verfahren für Berechnung der *FAR* in der Entwurfsphase zu empfehlen, die auf der Prüfleistung und der Ausfallrate der zu prüfenden Ausrüstungselemente beruht.

Die Vorhersage der Ausfallraten erfolgt in der Regel in der Entwurfsphase und die Prüfleistungen können durch den Vergleich mit bestehenden Systemen eingeschätzt werden.

Daher müssen zunächst die Empfindlichkeit und die Spezifität zusammen mit ihrer Beziehung zum Diagnosewert definiert werden, um die Fehlalarmrate zu ermitteln.

6.3.5.2 Empfindlichkeit und Spezifität

6.3.5.2.1 Allgemeines

Empfindlichkeit und Spezifität sind immanente Größen der Einheitsprüfung.

Statistisch gesehen misst die Empfindlichkeit einer Prüfung deren Fähigkeit, ein positives Ergebnis auszugeben, wenn eine Hypothese bestätigt wird.

Das ist das Gegenteil der Spezifität, die die Fähigkeit einer Prüfung misst, ein negatives Ergebnis auszugeben, wenn die Hypothese nicht bestätigt wird.

Empfindlichkeit und Spezifität sind weitere Kennzeichen für die Sachdienlichkeit einer Prüfung.

Im weiteren Verlauf dieses Dokuments kann die Hypothese entweder „Ausgefallenes Element“ oder „Einwandfrei funktionierendes Element“ lauten.

Die Begriffe „Einwandfreier Betrieb“ und „Ausfall“ bezeichnen nicht das Prüfergebnis, sondern den tatsächlichen Zustand des Systems. Die Begriffe OK und NOK bezeichnen die Prüfergebnisse.

Bei einer N -fach wiederholten Prüfung werden die Ergebnisse wie folgt aufgezeichnet:

- $TNOK$ (Wahres NOK): Anzahl der Ausfälle, die mit dem Prüfergebnis NOK beobachtet werden;
- $FNOK$ (Falsches NOK): Anzahl der einwandfreien Betriebsweisen, die mit dem Prüfergebnis NOK beobachtet werden;
- FOK (Falsches OK): Anzahl der Ausfälle, die mit dem Prüfergebnis OK beobachtet werden;
- TOK (Wahres OK): Anzahl der einwandfreien Betriebsweisen, die mit dem Prüfergebnis OK beobachtet werden.

Folglich müssen 4 Prüffälle betrachtet werden:

Tabelle 3 — Prüffälle

	Tatsächlicher Systemzustand	
	Ausgefallenes Element	Einwandfrei funktionierendes Element
Prüfergebnis NOK	$TNOK$	$FNOK$
Prüfergebnis OK	FOK	TOK

Es sollte Folgendes beachtet werden: $TOK + FOK + TNOK + FNOK = N$ (Gesamtanzahl der Prüfungen).

6.3.5.2.2 Empfindlichkeit und Spezifität der Prüfung für einen Ausfall

Eine Prüfung kann 2 Ziele haben: entweder wird versucht, den einwandfreien Betrieb des Systems zu bestätigen, oder es wird versucht, den Ausfall des Systems zu bestätigen.

Die Empfindlichkeit für einen Ausfall ist die Wahrscheinlichkeit, bei einem Ausfall das Ergebnis NOK zu erhalten. Sie wird nur im Fall eines Ausfalls gemessen. Sie wird wie folgt angegeben:

$$Sn_{\text{Ausfall}} = \frac{TNOK}{TNOK + FOK} \quad (12)$$

Die Empfindlichkeit einer Ausfallmessung geht immer mit einer Spezifität der Ausfallmessung einher. Letztere wird nur bei einwandfreiem Betrieb gemessen.

Die Spezifität für einen Ausfall ist also die Wahrscheinlichkeit, bei einwandfreiem Betrieb das Prüfergebnis OK zu erhalten, und sie wird wie folgt angegeben:

$$Sp_{\text{Ausfall}} = \frac{TOK}{TOK + FNOK} \quad (13)$$

Das Konzept der Empfindlichkeit und der Spezifität wird für zweigliedrige Prüfungen (ja/nein, positiv/negativ usw.) verwendet, während viele Signale einen kontinuierlichen Wert ergeben. Der Schwellenwert der Prüfung

(der Wert, bei dem entschieden wird, dass die Prüfung positiv ausgehen wird) beeinflusst deren Empfindlichkeit und Spezifität.

So kann durch die Anpassung von Schwellenwerten oder Toleranzen eine empfindlichere, aber weniger spezifische Prüfung (oder umgekehrt) erzeugt werden. Dieser Schwellen- oder Toleranzwert hängt stark vom Zweck der Prüfung ab. Sehr empfindliche Prüfungen sind vor allem sinnvoll, um sicherzustellen, dass kein Ausfall vorliegt (wenige falsche OK), während sehr spezifische Prüfungen sinnvoll sind, um sicherzustellen, dass ein Ausfall tatsächlich vorliegt (wenige falsche NOK).

6.3.5.2.3 Empfindlichkeit und Spezifität der Prüfung für einen einwandfreien Betrieb

Die Empfindlichkeit für einen einwandfreien Betrieb ist die Wahrscheinlichkeit, bei einwandfreiem Betrieb das Prüfergebnis OK zu erhalten. Sie wird nur im Fall eines einwandfreien Betriebs gemessen. Sie wird wie folgt angegeben:

$$Sn_{CO} = \frac{ TOK }{ TOK + FNOK } \quad (14)$$

Die Spezifität für einen einwandfreien Betrieb ist die Wahrscheinlichkeit, bei einem Ausfall das Prüfergebnis NOK zu erhalten. Sie wird wie folgt angegeben:

$$Sp_{CO} = \frac{ TNOK }{ TNOK + FOK } \quad (15)$$

Es wird angemerkt, dass für die Empfindlichkeit für einen Ausfall (bzw. für einen einwandfreien Betrieb) und für die Spezifität für einen einwandfreien Betrieb (bzw. für einen Ausfall) die gleiche Gleichung gilt.

Empfehlung 14: Daher sollte die Art der Empfindlichkeits- und der Spezifitätsmetriken eindeutig angegeben werden.

Es ist z. B. verbindlich vorgeschrieben, den Ausdruck „Empfindlichkeit für einen Ausfall“ zu benutzen. Der Begriff „Empfindlichkeit“ ohne Präzisierung bedeutet nichts und kann nur zu Missverständnissen führen.

ANMERKUNG Diese Metriken sind für den *BIT*-Entwickler vorgesehen. Sie sind für den Systementwickler von geringem Nutzen, der Diagnosewerte bevorzugt, die aus Sicht des Systems Messungen der Betriebseffizienz darstellen.

6.3.5.3 Diagnosewert

Der Diagnosewert für den Ausfall ist die Wahrscheinlichkeit, dass der Ausfall vorliegt, wenn das Prüfergebnis NOK ist (bei vorliegenden Gültigkeitsbedingungen).

Der Diagnosewert für den einwandfreien Betrieb ist die Wahrscheinlichkeit, dass der Ausfall nicht vorliegt, wenn das Prüfergebnis OK ist (bei vorliegenden Gültigkeitsbedingungen).

Nach Tabelle 3 ergibt Gleichung (16) den Diagnosewert für den Ausfall:

$$DV_{\text{Ausfall}} = \frac{ TNOK }{ TNOK + FNOK } \quad (16)$$

und der Diagnosewert für den einwandfreien Betrieb wird in Gleichung (17) abgeleitet:

$$DV_{CO} = \frac{ TOK }{ TOK + FOK } \quad (17)$$

Das Konzept des Diagnosewerts ist sehr wichtig, da während der Diagnose das Prüfergebnis bekannt ist. Und auf dieser Grundlage muss der Techniker beurteilen, ob der Ausfall vorliegt oder nicht. Die Diagnosewerte hängen von der Prävalenz (Auftretensrate) des Ausfalls ab (wichtig: Die Ausfallrate berücksichtigt nur interne

Ausfallursachen.). Je seltener (nicht sehr häufig) der Ausfall ist, desto besser ist also bei gleicher Empfindlichkeit und Spezifität der Diagnosewert einer bestimmten Prüfung für den einwandfreien Betrieb, und je häufiger der Ausfall ist, umso besser wird der Diagnosewert der gleichen Prüfung für den Ausfall sein.

ANMERKUNG 1 Eine Prüfung mit einem guten Diagnosewert für den Ausfall bedeutet, dass ihr Ergebnis NOK zuverlässig ist.

ANMERKUNG 2 Eine Prüfung mit einem guten Diagnosewert für den einwandfreien Betrieb bedeutet, dass ihr Ergebnis OK zuverlässig ist.

Eine Prüfung mit einem guten Diagnosewert für den einwandfreien Betrieb und mit einem schlechten Diagnosewert für den Ausfall gibt z. B. gültige Informationen, wenn das Prüfergebnis OK ist, sie ist aber schwer zu interpretieren, wenn das Prüfergebnis NOK ist.

Siehe Beispiel in Abschnitt A.3.

Darüber hinaus zeigt das Beispiel in Abschnitt A.5, wie die Rate der falschen Außerbetriebnahme oder die *NFF*-Rate (*NFF*, en: No fault found, kein Fehler festgestellt) in Bezug auf die Systemarchitektur und die Fehlalarmrate zu bewerten ist.

6.3.5.4 Berechnung der Fehlalarmrate

Die Fehlalarmrate wird aus dem Diagnosewert für den Ausfall abgeleitet und wie folgt angegeben:

$$FAR = \frac{FNOK}{TNOK + FNOK} = 1 - DV_{\text{Ausfall}} \quad (18)$$

Genauso wird die Definition der Fehlerrate bei Betriebsausfall (*FCOR*, en: False correct operation rate) aus dem Diagnosewert für den einwandfreien Betrieb abgeleitet und wie folgt angegeben:

$$FCOR = \frac{FOK}{TOK + FOK} = 1 - DV_{\text{Ausfall}} \quad (19)$$

Diese Raten entsprechen daher den Werten im vorstehenden Abschnitt.

Aus diesen Definitionen kann die Beziehung zwischen der Empfindlichkeit für einen Ausfall (Sn_{Ausfall}), der Spezifität für einen Ausfall (Sp_{Ausfall}), der Ausfallrate λ_F des Systems und der Fehlalarmrate abgeleitet werden.

$$(1) Sn_{\text{Ausfall}} \approx \frac{TNOK}{TNOK + FOK} \quad (20)$$

$$(2) Sp_{\text{Ausfall}} \approx \frac{TOK}{TOK + FNOK} \quad (21)$$

Die Hypothese der Ergodizität (siehe A.5) ermöglicht es, λ_F für die Anforderungen dieser Berechnung als Anzahl der Ausfälle je Prüfung anzugeben und nicht mehr als Anzahl der Ausfälle je Stunde. Die Anzahl der Ausfälle je Stunde wird durch Multiplikation dieser Zahl mit der Anzahl der Prüfungen je Stunde ermittelt.

Damit ergibt sich:

$$(3) \lambda_F \approx \frac{TNOK + FOK}{TNOK + FNOK + TOK + TOK} = \frac{TNOK + FOK}{N} \quad (22)$$

$$(4a) 1 - \lambda_F \approx \frac{TOK + FNOK}{TNOK + FNOK + TOK + TOK} = \frac{VOK + FNOK}{N} \quad (23)$$

$$(4b) FNOK \approx N(1 - \lambda_F) \quad (24)$$

$$(5) \leftarrow (2) \text{ und } (4a) Sp_{\text{Ausfall}} \approx \frac{TNOK}{N(1 - \lambda_F)} \leftrightarrow TNOK \approx N(1 - \lambda_F) \cdot Sp_{\text{Ausfall}} \quad (25)$$

$$(6) \leftarrow (5) \text{ und } (4b) FNOK \approx N(1 - \lambda_F) \cdot (1 - Sp_{\text{Ausfall}}) \quad (26)$$

$$(7) \leftarrow (1) \text{ und } (3) TNOK \approx N \cdot \lambda_F \cdot Sn_{\text{Ausfall}} \quad (27)$$

$$(8) FAR = \frac{FNOK}{TNOK + FNOK} \quad (28)$$

$$(9) \leftarrow (8) \text{ und } (7) \text{ und } (6) FAR \approx \frac{(1 - \lambda_F) \cdot (1 - Sp_{\text{Ausfall}})}{\lambda_F \cdot Sn_{\text{Ausfall}} + (1 - \lambda_F) \cdot (1 - Sp_{\text{Ausfall}})} \quad (29)$$

Es folgt ein Beispiel für den Einfluss der Ausfallrate auf die *FAR* für einen Entwurf und dessen Prüfungen. Die wichtigste Auswirkung, der für diese Prüfungen angestrebt wird, ist: *FOK* = 0 (um sicher zu sein, dass das Ergebnis der Prüfung nicht OK ist, wenn ein Ausfall vorliegt), d. h. eine Empfindlichkeit für einen Ausfall von 100 %.

Tabelle 4 zeigt die Betriebseffizienz der Prüfung hinsichtlich der Ausfallrate des Systems.

Tabelle 4 — Betriebseffizienz der Prüfung hinsichtlich der Ausfallrate des Systems

Prüfleistung		Nutzungs- zusammenhang	Betriebseffizienz
Sn_{Ausfall}	Sp_{Ausfall}	λ_F	FAR
Empfindlichkeit für einen Ausfall (Wahrscheinlichkeit, dass das Ergebnis der Prüfung beim Vorliegen eines Ausfalls NOK ist)	Spezifität für einen Ausfall (Wahrscheinlichkeit, dass das Ergebnis der Prüfung beim Vorliegen eines einwandfreien Betriebs OK ist)	Ausfallrate je Prüfung	Fehlalarmrate
100 %	99 %	0,1	8 %
100 %	99 %	0,01	50 %
100 %	99 %	0,001	91 %
100 %	99 %	0,000 1	99 %
100 %	99 %	0,000 01	> 99,99 %

Empfehlung 15: Da es sich bei der Empfindlichkeit und der Spezifität um Prüfleistungen handelt und die *FAR* mit Sp_{Ausfall} , Sn_{Ausfall} und der Ausfallrate in Zusammenhang steht, ist es sinnvoll, die Prüfungen miteinander zu vergleichen, indem die für diese Prüfungen vorhergesagten *FAR* verglichen werden.

Empfehlung 16: Der Hersteller sollte Informationen über die Spezifitäten und Empfindlichkeiten seiner Prüfungen sammeln, um die Empfindlichkeiten und Spezifitäten der später entworfenen Prüfungen vorhersagen bzw. vorwegnehmen zu können.

Die Datenbank der Empfindlichkeiten und Spezifitäten kann unter Verwendung einer bekannten *FAR* gefüllt werden.

Auf die gleiche Weise ergibt das:

$$(10) \leftarrow (3) FOK \approx N \cdot \lambda_F \cdot TNOK \quad (30)$$

$$(11) \leftarrow (10) \text{ und } (1) Sn_{\text{Ausfall}} \approx \frac{TNOK}{N \cdot \lambda_F} \leftrightarrow TNOK \approx N \cdot \lambda_F \cdot Sn_{\text{Ausfall}} \quad (31)$$

$$(12) \leftarrow (10) \text{ und } (11) FOK \approx N \cdot \lambda_F - N \cdot \lambda_F \cdot Sn_{\text{Ausfall}} \quad (32)$$

$$(13) FCOR = \frac{FOK}{TOK + FOK} \quad (33)$$

$$(14) \leftarrow (13), (12) \text{ und } (5) FCOR = \frac{\lambda_F \cdot (1 - Sn_{\text{Ausfall}})}{(1 - \lambda_F) \cdot Sp_{\text{Ausfall}} + \lambda_F \cdot (1 - Sn_{\text{Ausfall}})} \quad (34)$$

ANMERKUNG 1 Gleichung (14) verwendet die Empfindlichkeit und die Spezifität für einen Ausfall. Auf der Grundlage der gleichen Beweisführung können die *FCOR* und die *FAR* aufgestellt werden, indem die Empfindlichkeit für einen Ausfall durch die Spezifität für einen einwandfreien Betrieb bzw. die Spezifität für einen Ausfall durch die Empfindlichkeit für einen einwandfreien Betrieb ersetzt wird (siehe nachstehende Ausdrücke).

$$FCOR = \frac{\lambda_F \cdot (1 - Sp_{CO})}{(1 - \lambda_F) \cdot Sn_{CO} + \lambda_F \cdot (1 - Sp_{CO})} \quad (35)$$

$$FAR = \frac{(1 - \lambda_F) \cdot (1 - Sn_{CO})}{\lambda_F \cdot Sp_{CO} + (1 - \lambda_F) \cdot (1 - Sn_{CO})} \quad (36)$$

ANMERKUNG 2 Gleichung (14) zeigt, dass die *FCOR* fast null ist, wenn die Empfindlichkeit für einen Ausfall annähernd 100 % ist. Deshalb ist das wesentliche Ziel „Fehlerrate bei Betriebsausfall annähernd null“ (siehe 9.1.1) leicht zu erreichen.

7 Anwendung von internen Prüfverfahren (*BIT*)

7.1 Während der Entwicklung

In der Entwicklungsphase sind zwei der Ziele, Systemfehler zu beseitigen und Modelle oder Prototypen zu validieren. *BIT* können zu diesen Zielen beitragen.

7.2 Während der Produktion

In der Produktionsphase ist das Ziel, Produkte herzustellen, die trotz möglicher Abweichungen der Komponenten und Herstellungsprozesse mit dem qualifizierten Produkt übereinstimmen.

Die Eingrenzung der Grundursachen ist das Endziel der Prüfbarkeitstätigkeiten während der Produktion. Diese Eingrenzung hilft, das ausgefallene Element zu ermitteln, um die Übereinstimmung des ausgefallenen Produkts nach einer Nichtübereinstimmung wiederherzustellen.

Während der Produktion umfasst die Prüfbarkeitstätigkeit die Sicherstellung der Übereinstimmung der verkauften Funktionen/Leistungen, d. h.:

- Sicherstellung der Akzeptanz der Einrichtungen/Systeme;
- Beitrag zur Verringerung von Reparaturen während der Herstellung, indem die Eingrenzung von Ausfällen verbessert wird (Ersatz nur des ausgefallenen Teils);
- Beitrag zur Erkennung/Eingrenzung von flüchtigen Ausfällen während der Fehlerbehebung.

Folglich darf ein *BIT* in den folgenden beiden Fällen angewendet werden:

- am Ende der Herstellung: Dies sind die vom Hersteller durchgeführten Prüfungen zum Nachweis der Übereinstimmung der Spezifikation eines Produkts (System, Teilsystem, *LRU*, *SRU*, Produkt) mit einer qualifizierten Spezifikation (funktionale und physische Konfiguration);
- während des Environmental Stress Screenings: Hierbei handelt es sich um die Prüfungen, die vom Hersteller durchgeführt werden, um Ausfälle während der frühen Lebensphase eines Produkts in Bezug auf eine an einen Kunden verkaufte Spezifikation (funktionale und physische Konfiguration) zu erkennen, indem die Einheit unter Belastung (thermisch, mechanisch ...) geprüft wird.

7.3 Während des Einsatzes

7.3.1 Im Betriebszustand

Dies sind die Prüfungen von Produkten im Zustand der Betriebsfunktion, die vom Kunden zur Bestimmung von deren Betriebsverfügbarkeit verwendet werden.

Diese Prüfungen können automatisch (*PBIT* oder *CBIT*) erfolgen oder ausgelöst werden, wie z. B. das *IBIT*.

7.3.2 Im Instandhaltungsbetrieb

Im Instandhaltungsbetrieb kann dies die Prüfungen betreffen, die vom Hersteller (Lieferant oder Hersteller) oder vom Kunden durchgeführt werden für

- die Bestätigung des einwandfreien Betriebs des Systems (vorbeugende Instandhaltung),
- die Eingrenzung des ausgefallenen Elements, das zu ersetzen oder zu reparieren ist, und
- die Sicherstellung, dass das Produkt nach der Reparatur keine Fehler mehr aufweist, die mit der Spezifikation in Zusammenhang stehen.

Es kann unterschieden werden zwischen

- den Prüfungen in der Instandhaltungsphase, die direkt am Produkt durchgeführt werden (O-Ebene). In diesem Fall wird die Aufgabe von einem vom Kunden beauftragten Instandhaltungstechniker durchgeführt und
- den Prüfungen in der Instandhaltungsphase, die nicht direkt am Produkt durchgeführt werden (I-Ebene und D-Ebene). In diesem Fall wird die Aufgabe im Zusammenhang mit der industriellen Instandhaltung in einem Ausbesserungswerk durchgeführt.

7.4 Während der Validierung bei der Reparatur

Dieser Abschnitt wird in einer nachfolgenden Version erstellt.

8 Architektur von internen Prüfverfahren (BIT)

8.1 Die allgemeingültigen Funktionen der BIT

8.1.1 Allgemeines

In diesem Abschnitt werden die Grundfunktionen beschrieben, die das *BIT* eines Systems bietet. Diese Funktionen werden in zwei Gruppen unterteilt:

- *BIT*-Erkennung: diese Funktionen beziehen sich auf die Ausfallerkennung;

- *BIT*-Supervisor: Diese Funktionen beziehen sich auf die Zusammenfassung von *BIT*-Daten, -Diagnose, -Backup und die Schnittstelle mit den *BIT* anderer Systeme einer niedrigeren oder höheren Ebene.

Die verschiedenen Architekturen der *BIT*-Funktionen sind in 8.2 angegeben.

Bild 3 zeigt ein Beispiel für die logische Architektur eines *BIT*-Systems.

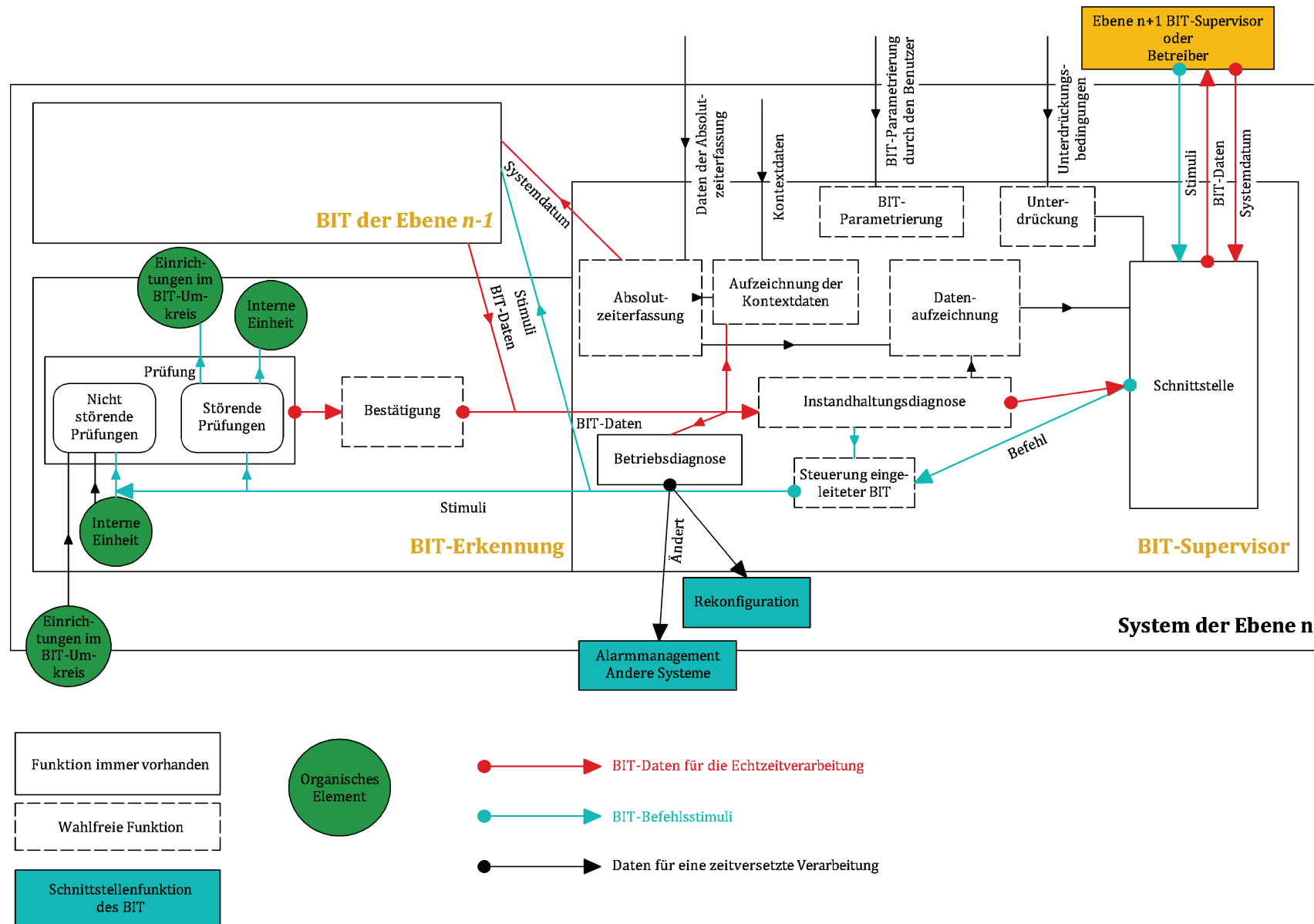


Bild 3 — Beispiel für die logische Architektur eines BIT-Systems

8.1.2 *BIT*-Erkennungsfunktion

8.1.2.1 Allgemeines

Die *BIT*-Erkennungsfunktion ist in allen Systemen vorhanden, die *BIT* ausführen.

Der Zweck dieser Funktion ist die Erkennung von

- internen Ausfällen des Systems, das diese *BIT*-Funktion integriert, und
- zugehörigen Systemausfällen, die mit dem festgelegten Anwendungsbereich dieser *BIT*-Funktion im Zusammenhang stehen.

Empfehlung 17: Es ist unerlässlich, den internen und externen Anwendungsbereich jeder *BIT*-Funktion des Systems festzulegen.

Der Anwendungsbereich einer *BIT*-Funktion ist die Liste der Elemente, die von der *BIT*-Funktion zu prüfen sind (die Elemente, die zu dem Element gehören können oder nicht, das die *BIT*-Funktion unterstützt).

Es werden 2 Teilfunktionen der *BIT*-Erkennungsfunktion bestimmt:

- Prüfung;
- Bestätigung.

8.1.2.2 Prüfung

Ein Ausfall oder eine Funktionsbeeinträchtigung wird mit einer Prüfung erkannt.

Eine Prüfung darf störend sein (diese Art von Prüfungen wird mitunter als intrusive Prüfung bezeichnet) oder nicht, sie darf ausgelöst werden oder nicht und sie darf funktionspezifisch sein oder nicht.

Das Prüfergebnis kann folgend Zustände annehmen:

- Prüfung OK (bestanden);
- Prüfung NOK (nicht bestanden).

Das *BIT* eines Systems integriert eine oder mehrere Prüfungen, die in einer Prüffolge auszuführen sind oder nicht.

8.1.2.3 Bestätigung (wahlfrei)

Vor dem Senden eines Ergebnisses NOK zum System der höheren Ebene darf das *BIT* folgende Funktionen ausführen:

- eine Filterfunktion für ein falsches NOK; das NOK muss z. B. mehrmals festgestellt werden, bevor es bestätigt wird;
- eine Bestätigungsfunktion durch automatische Wiederholung der Prüfung im Falle einer NOK-Prüfung; die Anzahl der Wiederholungen kann als Parameter festgelegt werden;
- eine Validierungsfunktion für das Prüfergebnis auf der Grundlage der Kriterien des Zusammenhangs (Hochlaufen, Übergangsphase usw.).

Die Bestätigung kann auch durch die *BIT*-Supervisor-Funktion durchgeführt werden.

8.1.3 *BIT*-Supervisor-Funktion

8.1.3.1 Allgemeines

Ein System der Ebene $n-1$ darf Supervisor für ein System der Ebene n sein (siehe Bild 3).

Dessen Aufgabe besteht darin, die Ergebnisse der *BIT*-Funktionen der *LRU* der Ebene $n-1$ zu erfassen und die Daten zusammenzufassen, um sie an die obere Ebene weiterzugeben.

Die folgenden Funktionen sind mit Ausnahme der Funktionen „Betriebsdiagnose“ und „Schnittstelle“ wahlfrei.

Die im Supervisor vorhandenen Funktionen hängen von der gewünschten Architektur auf der Ebene $n+1$ oder auf der Ebene des Gesamtsystems ab.

8.1.3.2 Betriebsdiagnose

Anhand des Ergebnisses einer Prüfung (oder einer Reihe von Prüfungen) stellt der *BIT*-Supervisor den Status einer Funktion (ihren technischen Zustand) fest.

Der Status einer Funktion darf mit den folgenden Zuständen belegt werden:

- einwandfreier Betrieb (verfügbar oder betriebsbereit);
- funktionsbeeinträchtigt (mit eingeschränkter Nutzung, teilweise erfüllte Funktion);
- ausgefallen (nicht verfügbar).

(Siehe 5.3 „Technische Systemzustände“)

ANMERKUNG Je nach Status der Funktionen kann das System z. B. neu konfiguriert werden, bestimmte Funktionen sperren, Redundanzen verwenden oder nichts tun.

Das System kann auch andere Systeme über den ausgefallenen Zustand der Funktionen informieren, um den Nutzer oder andere Peripheriesysteme zu alarmieren.

Siehe 5.2 für die Beschreibung der Funktionen im Zusammenhang mit dem *BIT*, wie z. B. die Alarm- und Rekonfigurationsfunktionen.

8.1.3.3 Aufzeichnung des Zusammenhangs (wahlfrei)

Bei einem bestätigten Ausfalls kann der Supervisor eine Zusammenhangsaufzeichnung für die anschließende Verarbeitung durchführen. Diese Aufzeichnung darf die folgenden Daten enthalten:

- Name der Funktion;
- ausgefallene *LRU*;
- Datierung;
- Fehlercode;
- unterstützter Wert;
- Daten über den Zusammenhang für eine eingehendere Analyse im Instandhaltungsbetrieb oder bei der Instandhaltung am Boden.

8.1.3.4 Systemdatum (wahlfrei)

Der Supervisor erhält von der oberen Ebene die Weltzeit, um sich mit dem übrigen System zu synchronisieren und die Verarbeitungsinformationen für seine Ebene zu datieren.

Diese Weltzeit wird vom Supervisor an die untere Ebene weitergegeben. Die *BIT*-Erkennungsfunktion wird daher für Ereignisse mit Absolutzeiterfassung die Weltzeit vergeben.

8.1.3.5 Instandhaltungsdiagnose (wahlfrei)

Das Ziel dieser Supervisor-Funktion ist die möglichst enge Eingrenzung des Fehlers auf ein austauschbares Element oder eine Mehrdeutigkeitsgruppe mit Hilfe einer Eingrenzungsmatrix (aus der Bibliothek der Signaturen für die Mehrdeutigkeitsgruppen).

Diese Funktion darf als Betriebsfunktion verfügbar sein oder nur während bestimmter Phasen der Mission (z. B. am Boden, während der Instandhaltungsphasen).

Wenn diese Funktion implementiert ist, erlaubt sie dem Supervisor, eine Nachricht an das System der Ebene $n+1$ zu senden, um dieses über die Eingrenzung eines Ausfalls zu informieren.

8.1.3.6 Datenaufzeichnung (wahlfrei)

Der Supervisor darf eine Funktion zur Aufzeichnung von Daten während der gesamten Dauer der Mission für einen aufgeschobenen Einsatz am Boden haben.

8.1.3.7 Schnittstelle

Die Schnittstelle dient der Übermittlung der *BIT*-Ergebnisse des Systems der Ebene n an den *BIT*-Supervisor der oberen Ebene. Die Ergebnisse werden auf einem fest zugeordneten oder nicht fest zugeordneten Datenbus nach einem mit dem Supervisor der Ebene $n+1$ festgelegten Protokoll übermittelt.

Diese Funktion verwaltet auch die *BIT*-Befehle, die von einem Bediener oder einem anderen System (*IBIT*, *MBIT*, *EBIT*) ausgelöst und an die Erkennungsfunktionen ausgegeben werden.

8.1.3.8 Steuerung des ausgelösten *BIT* (wahlfrei)

Der Supervisor darf Prüfungen unter den folgenden Situationen neu starten:

- bei einer vorübergehenden Anomalie kann der Supervisor das *BIT* eigenverantwortlich erneut starten, um einen Defekt zu bestätigen;
- im Zusammenhang mit einem Betriebszustand des Systems (z. B. Ein- oder Ausschalten des Motors) kann der Supervisor über Befehle an die Erkennungsfunktion eine Prüfung eigenverantwortlich einleiten;
- auf Befehl über die Schnittstellenfunktion generiert die Funktion (je nach systembezogenen Unterdrückungen) Befehle an die Erkennungsfunktionen oder nicht, um die ausgelösten Prüfungen einzuleiten.

8.1.3.9 Unterdrückung (wahlfrei)

Im Betriebszustand und unter bestimmten Bedingungen oder auf Befehl des Bedieners darf das Supervisor-*BIT* einige seiner Funktionen unterdrücken (vollständige oder teilweise Unterdrückung).

Beispiele:

- Die eingeleiteten Prüfungen werden während des Flugbetriebs des Luftfahrzeugs unterdrückt.

- Die Übertragung von Informationen an das System der Ebene $n+1$ wird während der Abschaltsequenz unterdrückt.

8.1.3.10 BIT-Parametrierung (wahlfrei)

Diese Funktion wird verwendet, um das Verhalten der *BIT*-Funktion durch technische Konfiguration zu modifizieren.

Verschiedene *BIT*-Parameter dürfen veränderbar sein.

Sie dürfen z. B. die folgenden Funktionen beeinflussen:

- Schnittstelle: Modifikation der zu sendenden Informationen;
- Eingrenzungsdiagnose: die Eingrenzungsmatrix darf wie gefordert konfiguriert werden;
- Bestätigung: Modifikation des Schwellenwerts bei Bestätigung durch Zählung;
- Erkennung: Modifikation der Parameter der Prüfungen für die Ausfallerkennung.

Siehe 5.5.3 „Parametrierung der *BIT*“.

8.2 Die verschiedenen Architekturen der *BIT*-Funktion

8.2.1 Allgemeines

In diesem Abschnitt wird ein System der Ebene n betrachtet, das als System A bezeichnet wird. System A umfasst Teilsysteme der Ebene $n-1$ und muss *BIT*-Daten an das System der Ebene $n+1$ liefern.

In diesem Abschnitt werden die verschiedenen möglichen Architekturen für die *BIT*-Funktion im System angegeben und ihre verschiedenen Merkmale definiert.

Der Entwickler des Systems der Ebene n kann dann die am besten geeignete *BIT*-Architektur empfehlen, um die Anforderungen seines Kunden (System der Ebene $n+1$) zu erfüllen.

In komplexen Systemen muss der *BIT*-Spezifizierer unter Berücksichtigung des Systemzusammenhangs eine Funktionsanalyse der *BIT*-Architektur mit einem Top-down-Ansatz zur Spezifikation der *BIT*-Funktionen der unteren Ebene durchführen.

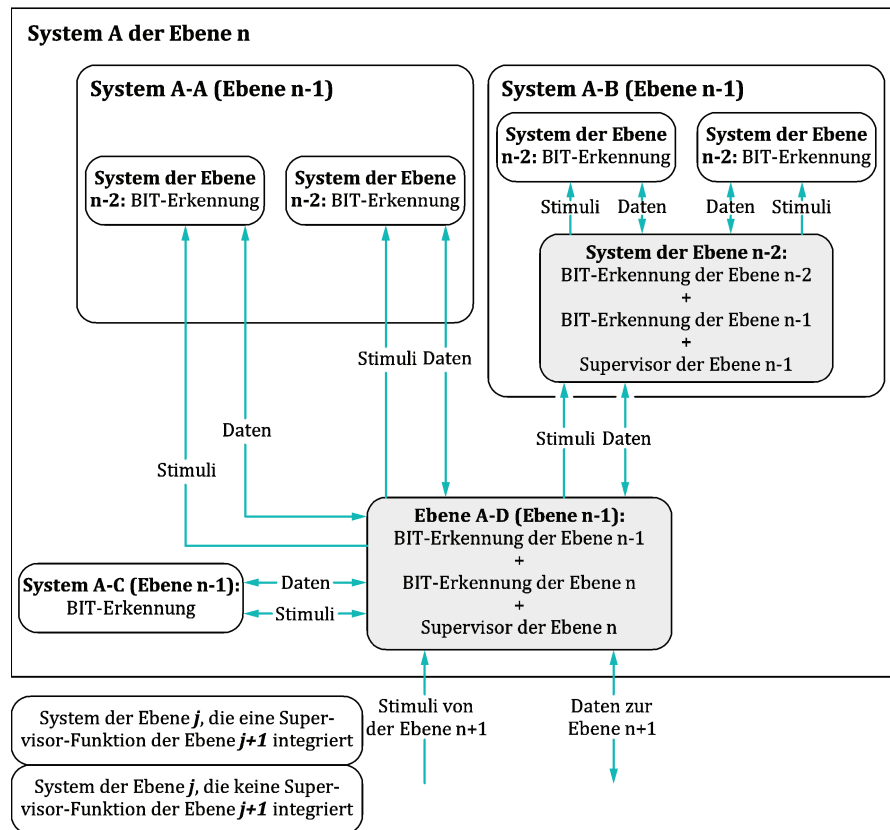


Bild 4 — BIT-Architektur

In diesem Beispiel

- weist System A eine zentralisierte *BIT*-Architektur bezüglich des Systems der Ebene $n+1$ auf,
- weist System A-A eine verteilte *BIT*-Architektur bezüglich des Systems A auf, und
- was System A-B betrifft, weist dieses eine zentralisierte *BIT*-Architektur bezüglich des Systems A auf.

ANMERKUNG „Stimuli“ sind numerische oder analoge Informationseinheiten, die zur Einleitung einer Prüfung verwendet werden (*IBIT*-Auslöser der Ebene $n-1$, Aktivierung einer Ausgabe für eine gemeinsame Prüfung usw.).

8.2.2 Verteilte *BIT*-Architektur

In der verteilten Architektur binden die Systeme unterer Ebenen keinen Supervisor ein (oder zumindest einen sehr begrenzten Supervisor).

Der *BIT*-Supervisor der Ebene n ist für die in 8.1 beschriebenen Funktionen zuständig. In dem Beispiel in Bild 4 „*BIT*-Architektur“ kontrolliert System A-D direkt die gesamte Einrichtung, die das System A-A bildet.

Der Supervisor der Ebene n benötigt Rohinformationen von den *BIT*-Erkennungsfunktionen der Ebenen $n-1$. Es ist die Eingrenzungsfunktion des Supervisors der Ebene n , die den Ausfall anhand der Liste der Ausfallsignaturen so gut wie möglich eingrenzt (die Größe der Mehrdeutigkeitsgruppe verringert).

8.2.3 Zentralisierte *BIT*-Architektur

In der zentralisierten Architektur wird für ein gegebenes System der Ebene n ein Element der Ebene $n-1$ eine Supervisor-*BIT*-Funktion der Ebene n integrieren, die die *BIT*-Informationen der Systeme der Ebene $n-1$ präzisiert, um den Ausfall so gut wie möglich einzugrenzen (die Größe der Mehrdeutigkeitsgruppe zu verringern). Im Beispiel in Bild 4 „*BIT*-Architektur“ wirkt eines der Ausrüstungsteile des Systems A-B als Supervisor für dieses System (siehe Definition in Abschnitt 3). Dabei handelt es sich um das einzige Ausrüstungsteil, das mit dem Supervisor der Ebene n (System A-D) kommuniziert.

8.2.4 Wahl der *BIT*-Architektur

Zunächst kann festgestellt werden, dass die *BIT*-Architektur von der Systemarchitektur maßgeblich wie folgt beeinflusst wird:

- funktionale Ebene der Systemautonomie;
- bordeigener oder nicht bordeigener Teil des Systems;
- zentralisiertes oder verteiltes System;
- im System verwendeter Datenbus:
 - Determinismus;
 - Verwaltung der Prioritäten;
 - Auslastung des Busses.

Ob eine Eingrenzungsfunktion in ein Supervisor-*BIT* der Ebene n integriert wird oder nicht, hängt hauptsächlich vom Grad der Unabhängigkeit des Systems vom System der Ebene $n+1$ ab. Wenn ein System der Ebene n eine große Anzahl von Schnittstellen zu anderen Systemen derselben Ebene hat, wird es schwierig sein, einen Ausfall einzugrenzen, da er von einem der anderen Systeme, zu denen es eine Schnittstelle hat, stammen kann. Daher wird eine verteilte Architektur in Bezug auf ein System der Ebene $n+1$ empfohlen. Dies gilt umso mehr, als der Supervisor der Ebene $n+1$ besser in der Lage sein wird, die Signatur des Ausfalls zu ermitteln (tatsächlich verfügt er über alle *BIT*-Daten der Systeme der Ebene n).

Empfehlung 18: Wenn ein *BIT* der Ebene $n-1$ vom verteilten Typ ist, dann muss das *BIT* der Ebene n die Schnittstellen zum Ausrüstungsteil der Ebene $n-1$ prüfen, um die Diagnose auszuführen, indem der Zustand der externen Ressourcen, die für jedes Ausrüstungsteil der Ebene $n-1$ erforderlich sind, nachvollzogen wird.

Wenn dieses System jedoch sehr unabhängig ist (wenige Schnittstellen mit anderen Systemen hat), dann wird es für das System „einfach“ sein, bei einem Ausfall eine seiner *LRUs* automatisch einzugrenzen. Daher wird für das System der höheren Ebene eine zentralisierte Architektur empfohlen.

Andererseits erfordert die verteilte Architektur eine erhebliche Datenübertragungsgeschwindigkeit zum Supervisor. In einigen Fällen können diese Datenübertragungsgeschwindigkeiten begrenzt sein (bei einem ARINC-Bus, 1553 usw.). Daher wird eine zentralisierte Architektur empfohlen.

Die *BIT*-Architektur wird auch von der Rolle (oder den Rollen) abhängen, die der Diagnosefunktion zugewiesen wird: Betrieb oder Instandhaltung.

Empfehlung 19: Die Ansprechzeit des *BIT* und die Ausbreitungszeit für dessen Ergebnisse müssen bekannt, deterministisch und mit der getroffenen Entscheidung vereinbar sein.

ANMERKUNG Die Zeit, die mit der getroffenen Entscheidung vereinbar ist, ist im Betriebszustand restriktiver als im Instandhaltungsbetrieb.

Empfehlung 20: Die Betriebsdiagnose muss auf der höchsten Ebene der Zentralisierung in Bezug auf die *BIT*-Ansprechzeiten durchgeführt werden.

Tabelle 5 fasst die verschiedenen Kriterien mit den Vor- und Nachteilen in Gruppen zusammen, die bei der Wahl einer *BIT*-Architektur zu berücksichtigen sind.

Tabelle 5 — Kriterien für die Wahl einer *BIT*-Architektur

Systemparameter	Wahl der <i>BIT</i> -Architektur		Einflussgrad des Parameters	Bemerkungen
	Zentralisiert	Verteilt		
Maßgebliche funktionale Autonomie des Systems	Empfohlene Wahl	–	Verfahren	Eingrenzungsdiagnosefunktion, die in das System- <i>BIT</i> zu integrieren ist
Verteilte Systemarchitektur	–	Empfohlene Wahl	Hoch	–
Zentralisierte Systemarchitektur	Empfohlene Wahl	–	Hoch	–
Deterministischer Datenbus	Empfohlene Wahl	–	–	Mögliche zentralisierte Betriebsdiagnosefunktion. Der deterministische Datenbus ist vorteilhaft für eine kurze Ausbreitungszeit.
Datenbus mit Verwaltung der Prioritäten	Empfohlene Wahl	–	–	Zentralisierte Betriebsdiagnosefunktion möglich, wie oben
Schnelle Ansprechzeit der Prüfung	Empfohlene Wahl	–	–	Zentralisierte Betriebsdiagnosefunktion möglich, wie oben
Maßgebliche Bandbreite auf dem Datenbus	Möglich (aber „verteilt“ ist die bessere Wahl)	Empfohlene Wahl	Niedrig	–

8.3 Topologie der ausgetauschten Daten

Die Daten, die von einem System der Ebene $n-1$ zu einem System der Ebene n weitergegeben werden müssen, sind

- die *BIT*-Daten;
- die Validierungsdaten;
- die Absolutzeiterfassung der Ereignisse und
- die Kontextdaten.

Empfehlung 21: Die Kontextdaten für die Ausfälle sollten erfasst werden, da sie sehr nützlich für den Erfahrungsrückfluss und die kontinuierliche Verbesserung der Diagnose sind.

Empfehlung 22: Alle *BIT*-Daten und Kontextdaten der Ausfälle müssen die gleiche Zeitreferenz verwenden und synchron zur Verfügung gestellt werden.

Unter den Normen für den *BIT*-Datenaustausch kann die Norm ARINC 624 zitiert werden [6].

8.4 *BIT*-Spezifikation und Modellierung

8.4.1 Spezifikationsprozess

Empfehlung 23: Die Entwicklungsgeschwindigkeit eines Systems erfordert die gleichzeitige Entwicklung von *BIT*.

8.4.2 Entscheidungen des Systementwurfs: wesentliches Ziel und Aufwand

Es wird das Beispiel eines Luftfahrzeugs betrachtet.

Der Hersteller des Systems muss „ein funktionsfähiges System herstellen“. Aus Sicht dieses Dokuments bedeutet ein funktionsfähiges System, dass

- das Luftfahrzeug zuverlässige Informationen über seinen einwandfreien Betrieb und damit seine Betriebsfähigkeit hervorbringt;
- das Luftfahrzeug nicht übermäßig wegen „falscher Defekte“ stillgelegt wird und/oder
- die Stilllegungsdauer und die Ressourcen zur Eingrenzung des ausgefallenen Elements so gering wie möglich sind.

Für die ersten beiden Punkte muss der Entwickler im Idealfall eine *BIT*-Lösung entwickeln, die eine globale Auswirkung hat, indem der Zustand des Systems, d. h. entweder der einwandfreie Betrieb oder der Ausfall mit seiner sofortigen Eingrenzung des ausgefallenen Elements eindeutig angegeben wird.

Unabhängig von den Gründen (physisch, technologisch, wirtschaftlich usw.) kann der Systemlieferant jedoch der Ansicht sein, dass diese globale eindeutige Auswirkung zu viel Aufwand erfordert. Aus Gründen der Betriebseffizienz wird der Systemlieferant verschiedene Betriebszusammenhänge in Betracht ziehen und für jeden eine Entscheidung treffen, indem das zu erreichende wesentliche Ziel definiert wird.

Im Zusammenhang mit dem Flug besteht ein wesentliches Ziel z. B. darin, „zuverlässige Informationen über den einwandfreien Betrieb hervorzubringen“.

Bei der Betrachtung einer Prüfung, bei der die Abweichung einer physikalischen Größe über einen vorgegebenen Toleranzbereich hinaus überprüft wird, kann der Diagnosewert für den einwandfreien Betrieb von 100 % ($FCOR \approx 0$) durch eine Einschränkung des Toleranzbereichs erzielt werden.

Das wesentliche Ziel ist erreicht, aber der Systemlieferant ist sich bewusst, dass dadurch die Fehlalarmrate verschlechtert (erhöht) wird.

In Anbetracht der Ziele des Lieferanten („ein betriebsfähiges System herstellen“) kann ein zusätzlicher Aufwand (der erheblich sein kann) erforderlich sein, um die Fehlalarmrate zu verringern.

In diesem Beispiel ist die Tätigkeit, die für den Entwickler ein Problem darstellt, die Lenkung der Fehlalarmrate.

In diesem Dokument entspricht das Wort „Aufwand“ der Tätigkeit in der (besonders entscheidenden) Entwicklungsphase, in der der Entwickler hauptsächlich einen erheblichen Aufwand unternehmen muss, um die Entwurfsanforderungen zu erfüllen. Der Erfolg dieser Tätigkeit (oder Phase) wird durch die Metrik der Betriebseffizienz verzeichnet.

Die folgende Tabelle 6 ist eine (nicht erschöpfende) Darstellung der verschiedenen Entscheidungssituationen zwischen Zielen und Aufwand. Es sollte beachtet werden, dass die Wahl des wesentlichen Ziels und des Aufwands nur Beispiele für ein System im Zusammenhang mit einem Luftfahrzeug sind.

Tabelle 6 — Beispiel für die Entscheidungen zwischen Ziel und Aufwand beim Systementwurf

Zusammenhang	Zu erreichende Ziele	Wesentliches Ziel (Wahl aus den Zielen)	Betriebseffizienz des wesentlichen Ziels	Zusätzlicher Aufwand	Betriebseffizienz des zusätzlichen Aufwands
Der Techniker grenzt die ausgefallene Einrichtung ein.	Eingrenzung der ausgefallenen Einrichtung ohne Mehrdeutigkeit Kein spezifisches Verfahren der Fehlersuche	Kein spezifisches Verfahren der Fehlersuche	Umfang der Ressourcen der Fehlersuche mit maximaler Wertemenge Erkennungsrate (Ausfälle, die zu Haupt- und Nebenalarmen führen) mit minimaler Wertemenge	Begrenzung der Mehrdeutigkeitsgruppen des Ausfalls	Maximale Eingrenzungsrate
Nach der Fehlersuche erklärt der Techniker die Genehmigung zur Wiederinbetriebnahme anhand der <i>BIT</i> -Information. Der Pilot überprüft vor dem Flug, ob das Luftfahrzeug einwandfrei funktioniert. Der Pilot überprüft während des Flugs, ob das System einwandfrei funktioniert. Ist das nicht der Fall, stoppt er die betriebliche Mission.	<i>BIT</i> mit einem Diagnosewert für den einwandfreien Betrieb (siehe Definition in Abschnitt 3) <i>BIT</i> mit einem Diagnosewert für den Ausfall (siehe Definition in Abschnitt 3)	<i>BIT</i> mit einem Diagnosewert für den einwandfreien Betrieb	Diagnosewert für den einwandfreien Betrieb mit minimaler Wertemenge Erkennungsrate (Ausfälle, die zu Hauptalarmen führen) mit minimaler Wertemenge	Erhaltung der Verfügbarkeit	Minimale Fehlalarmrate Minimale Steigerungsrate der Unzuverlässigkeit (siehe Definition in Abschnitt 3)

Empfehlung 24: Der Systemlieferant muss für den Entwurf des *BIT* die Hierarchie seiner Anforderungen an seine Lieferanten (wesentliche Ziele und zusätzlicher Aufwand) klar darlegen.

8.4.3 Der *BIT*-Spezifikationsprozess

8.4.3.1 Allgemeines

Empfehlung 25: Vorzugsweise führt der Hersteller der Ebene n eine Funktionsanalyse durch und leitet die Leistungszuweisungen und Ressourcen für den Hersteller der Ebene $n-1$ ab. Wenn der Hersteller der Ebene n diese Leistungszuweisungen nicht zur Verfügung stellt, müssen diese Informationen vom Hersteller der Ebene $n-1$ zur Validierung durch den Hersteller der Ebene n bereitgestellt werden.

ANMERKUNG Dieser Abschnitt schlägt einen Spezifikationsprozess vor. Die Vielfalt der Situationen, sei es in Bezug auf die Betriebsanforderungen oder die Architektur, macht es unmöglich, einen allgemeingültigen Wert für die verschiedenen *BIT*-Leistungen und Betriebseffizienzen zu empfehlen.

Unabhängig von den Zusammenhängen oder den wesentlichen Zielen definiert der Hersteller (unabhängig von der Hierarchieebene) eine Prüfstrategie (Nutzung von *BIT*?, Verwendung von externen Prüfeinrichtungen usw.), die sich auf die Hardware-Architektur des zu prüfenden Systems auswirkt, aber auch auf

- die *BIT*-Architektur und
- die Definition der Prüfungen.

Für diese 2 Punkte muss der Spezifizierer die in den folgenden Abschnitten entwickelten Fragen stellen.

8.4.3.2 Für die *BIT*-Architektur

Der Spezifizierer muss die folgenden Fragen stellen:

- 1) Welche sind die möglichen Ausfallarten? Welche werden geprüft und welche nicht und warum? (Siehe 6.3.2)
- 2) Wie lautet das Ziel für die Ausfalleingrenzungsrate? (Siehe 6.3.3)
- 3) Welche Ausrüstungselemente müssen grundsätzlich gemeinsam mit anderen Systemelementen geprüft werden?

8.4.3.3 Für die Prüfungen

Der Spezifizierer muss die folgenden Fragen stellen:

- 1) Wie wäre bei einem Vergleich mit einem vorhandenen System die Betriebseffizienz der Prüfung?
 - Wie viel Vertrauen kann in dem gegebenen Zusammenhang in das *BIT*-Ergebnis gelegt werden? (Siehe 6.3.5.2.)
 - Wie hoch ist die Fehlalarmrate? Wie hoch ist die Fehlerrate bei Betriebsausfall? (Siehe 6.3.5.)
- 2) Wie wäre bei einem Vergleich mit einem vorhandenen System die Prüfleistung?
 - Wie hoch ist die Steigerungsrate der Unzuverlässigkeit? (Siehe 6.3.4.)
 - Wie sind die Leistungen der Empfindlichkeit und der Spezifität? (Siehe 6.3.5.)
 - Wie lange ist die Ansprechzeit der Prüfung?
 - Welche Gültigkeitsbedingungen gibt es für die Prüfung? Welcher Zeitrahmen wird bei der Prüfung für die Ausgabe gültiger Ergebnisse gefordert, sobald die Gültigkeitsbedingungen erfüllt sind?
 - Wie ist die Betriebseffizienz der Prüfung?

Der Entwickler muss die anzuwendenden Ressourcen kennen (oder die Zuordnungen festlegen) hinsichtlich von:

- Masse;
- Kosten;
- Energien;
- usw.

Nach dieser ersten Analyse wird die Liste der erforderlichen Prüfungen ermittelt. Die Art ist noch festzulegen.

Empfehlung 26: Die folgende Tabelle 7 legt die Verteilungsstrategie der Prüfungen auf die verschiedenen *BIT*-Arten für ein rekonfigurierbares und kontrollierbares System fest.

Z. B. muss dem *IBIT* der Vorzug gegeben werden, wenn die Prüfung zu einer Störung führt, die nur in bestimmten, dem System nicht bekannten Fällen zulässig ist.

Tabelle 7 — Verteilungsstrategie der Prüfungen auf die verschiedenen *BIT*-Arten

Auslösung der Prüfung, die vom System verwaltet werden kann	Störung des Betriebszustands durch die Prüfung	Dauer der Prüfung	Art des für diese Prüfung empfohlenen <i>BIT</i>
Ja	Zulässig	Nicht anwendbar	<i>CBIT</i>
Ja	Nicht zulässig	Zulässig	Warmstart- <i>PBIT</i>
Ja	Nicht zulässig	Nicht zulässig	Kaltstart- <i>PBIT</i>
Nein	In einigen Fällen nicht zulässig	In einigen Fällen nicht zulässig	<i>IBIT</i>

ANMERKUNG Definition der Kaltstart- und der Warmstart-*PBIT* siehe 6.2.1.

8.5 Allgemeine Modellierungs- und Konfigurationssprache

8.5.1 Einleitung

Der Zweck dieses Abschnitts ist die Empfehlung einer Sprache zur Beschreibung eines *BIT*, die es Herstellern ermöglicht, *BIT*-Funktionsmodelle (und deren Parametrierung) unabhängig von den Entwurfswerkzeugen und Betriebsarchitekturen auszutauschen.

Die *BIT*-Sprache wird zur Kommunikation zwischen Herstellern verwendet, um

- fundierte *BIT*-Modelle zu übertragen und
- die Systeme in Betrieb zu konfigurieren.

Während der Systementwurfsphase ist die zentrale Information der *BIT*-Funktion die Schnittmenge (siehe Definition in Abschnitt 3), weil diese aus der *FMECA*-Analyse (die eine Entwurfstätigkeit ist) weitergegeben wird.

In der Betriebsphase ist die zentrale Information der *BIT*-Funktion jedoch die Signatur (siehe Definition in Abschnitt 3), da sie die einzige Information ist, die beobachtet werden kann.

Diese 2 Ansätze werden nachfolgend veranschaulicht.

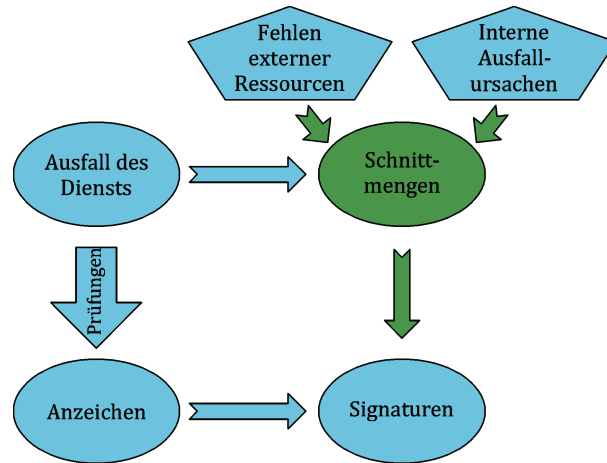


Bild 5 — Zentrale Information der *BIT*-Funktion in der Entwurfsphase – die Schnittmengen

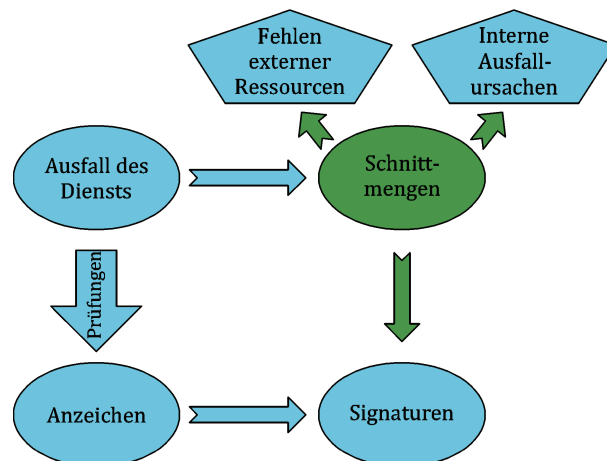


Bild 6 — Zentrale Information der *BIT*-Funktion in der Betriebsphase – die Signaturen

Deshalb basiert die *BIT*-Sprache auf der Schnittmenge und auf der Signatur.

Die Sprache basiert auf 5 Tabellen, die für jedes Gerät bis zur erforderlichen Tiefe (Einrichtung, Platine, Komponente usw.) auszufüllen sind; sie werden nachstehend kurz beschrieben und ihr Inhalt wird in 9.2.2 ausführlich beschrieben:

- Die Tabelle der ausgefallenen Dienste enthält die Liste der ausgefallenen Dienste. Diese Liste kann aus der Spalte „Auswirkungen“ der *FMECA* abgeleitet werden.
- Die Tabelle der Schnittmengen enthält alle Ausfälle, die die jeweilige Schnittmenge bilden.
- Die Tabelle der Signaturen führt die Prüfungen und ihre Ergebnisse auf (Entstehung der Ausfallsignaturen).
- Die Tabelle der Schnittmengen/Signaturen ordnet die Schnittmengen den Signaturen zu.