

## Inhalt

	Seite
Vorwort.....	7
Einleitung .....	8
1 Anwendungsbereich .....	11
2 Normative Verweisungen .....	12
3 Begriffe und Abkürzungen .....	12
3.1 Begriffe .....	12
3.2 Abkürzungen .....	17
4 Ziele, Konformität und Software-Sicherheits-Integritätslevel .....	17
5 Softwaremanagement und -organisation .....	18
5.1 Organisation, Rollen und Verantwortlichkeiten .....	18
5.2 Kompetenz der Mitarbeiter .....	22
5.3 Fragen des Lebenszyklus und Dokumentation .....	23
6 Software-Sicherung .....	26
6.1 Softwaretests.....	26
6.2 Software-Verifikation .....	27
6.3 Software-Validierung .....	29
6.4 Software-Begutachtung.....	31
6.5 Software-Qualitätssicherung .....	33
6.6 Änderungen und Änderungsmanagement .....	35
6.7 Unterstützende Werkzeuge und Sprachen .....	36
7 Entwicklung generischer Software .....	40
7.1 Lebenszyklus und Dokumentation für generische Software .....	40
7.2 Software-Anforderungen .....	40
7.3 Architektur und Entwurf .....	43
7.4 Komponentenentwurf .....	49
7.5 Implementierung und Test der Komponenten .....	51
7.6 Integration.....	52
7.7 Test der Gesamtsoftware/Abschließende Validierung .....	54
8 Entwicklung der Anwendungsdaten oder -algorithmen – Systeme, die durch Anwendungsdaten oder -algorithmen konfiguriert werden.....	56
8.1 Ziele .....	56
8.2 Eingangsdokumente.....	56
8.3 Ausgangsdokumente.....	56
8.4 Anforderungen.....	57
9 Bereitstellung und Wartung der Software.....	61
9.1 Bereitstellung der Software .....	61
9.2 Wartung der Software.....	63

	Seite
Anhang A (normativ) Kriterien für die Auswahl der Techniken und Maßnahmen .....	67
A.1 Tabellen zu den Abschnitten .....	68
A.2 Detaillierte Tabellen.....	76
Anhang B (normativ) Software-Schlüsselrollen und Verantwortlichkeiten .....	82
Anhang C (informativ) Zusammenfassung der Dokumentenkontrolle .....	91
Anhang D (informativ) Verfahrensübersicht .....	93
D.1 KI(Künstliche-Intelligenz)-Fehlerkorrektur (en: AI Fault Correction).....	93
D.2 Analysierbare Programme .....	93
D.3 Avalanche-/Belastungstests (en: Avalanche/Stress Testing) .....	94
D.4 Grenzwertanalyse (en: Boundary Value Analysis).....	94
D.5 Rückwärts-Regeneration (en: Backward Recovery).....	95
D.6 Ursache-Wirkungsdiagramme (en: Cause Consequence Diagrams).....	95
D.7 Checklisten (en: Checklists).....	95
D.8 Steuerflussanalyse (en: Control Flow Analysis).....	96
D.9 Analyse gemeinsamer Fehler (en: Common Cause Failure Analysis) .....	96
D.10 Datenflussanalyse (en: Data Flow Analysis).....	97
D.11 Datenflussdiagramme (en: Data Flow Diagrams).....	97
D.12 Datenaufzeichnung und -analyse (en: Data Recording and Analysis) .....	98
D.13 Entscheidungstabellen (Wahrheitstabellen) (en: Decision Tables (Truth Tables)).....	98
D.14 Defensive Programmierung (en: Defensive Programming).....	99
D.15 Codierstandards und Anleitung zum Programmierstil (en: Coding Standards and Style Guide).....	100
D.16 Diversitäre Programmierung (en: Diverse Programming).....	100
D.17 Dynamische Rekonfiguration (en: Dynamic Reconfiguration) .....	101
D.18 Tests auf Basis von Äquivalenzklassen und Eingangsdaten-Unterteilung (en: Equivalence Classes and Input Partitioning Testing) .....	101
D.19 Fehlererkennende und -korrigierende Codes (en: Error Detecting and Correcting Codes) .....	102
D.20 Fehlererwartung (en: Error Guessing) .....	102
D.21 Fehlereinstreuung (en: Error Seeding) .....	102
D.22 Ereignisbaumanalyse (en: Event Tree Analysis) .....	103
D.23 Fagan-Inspektionen (en: Fagan Inspections) .....	103
D.24 „Failure Assertion“-Programmierung (en: Failure Assertion Programming) .....	103
D.25 SEEA – Softwarefehler-Auswirkungsanalyse (en: Software Error Effect Analysis).....	104
D.26 Fehlererkennung und Diagnose (en: Fault Detection and Diagnosis).....	104
D.27 Finite-Zustandsmaschinen (FSM)/Zustands-Übergangsdigramme (en: Finite State Machines/State Transition Diagrams).....	105
D.28 Formale Methoden (en: Formal Methods) .....	106
D.29 Formaler Beweis (en: Formal Proof).....	111
D.30 Vorwärts-Regeneration (en: Forward Recovery) .....	111
D.31 Abgestufte Funktionseinschränkungen (en: Graceful Degradation).....	112

	Seite
D.32	Auswirkungsanalyse (en: Impact Analysis)..... 112
D.33	Information-Hiding/Einkapselung (en: Information Hiding/Encapsulation)..... 112
D.34	Schnittstellentests (en: Interface Testing) ..... 113
D.35	Untermenge der Programmiersprache (en: Language Subset)..... 113
D.36	Aufzeichnung ausgeführter Fälle (en: Memorising Executed Cases) ..... 113
D.37	Metriken (en: Metrics)..... 114
D.38	Modularer Ansatz (en: Modular Approach) ..... 114
D.39	Leistungs-Modellierung (en: Performance Modelling)..... 115
D.40	Leistungsanforderungen (en: Performance Requirements)..... 115
D.41	Wahrscheinlichkeits-Tests (en: Probabilistic Testing)..... 116
D.42	Prozesssimulation (en: Process Simulation)..... 117
D.43	Prototyping/Animation ..... 117
D.44	Recovery Block..... 117
D.45	Antwortzeiten und Speichergrenzen (en: Response Timing and Memory Constraints) ..... 118
D.46	„Re-Try Fault Recovery“-Mechanismen (en: Re-Try Fault Recovery Mechanisms)..... 118
D.47	Externe Überwachungseinrichtung (en: Safety Bag) ..... 118
D.48	Software-Konfigurationsmanagement (en: Software Configuration Management)..... 119
D.49	Streng typisierte Programmiersprache (en: Strongly Typed Programming Languages) ..... 119
D.50	Strukturabhängige Tests (en: Structure Based Testing) ..... 119
D.51	Strukturdiagramme (en: Structure Diagrams) ..... 120
D.52	Strukturierte Methodik (en: Structured Methodology) ..... 120
D.53	Strukturierte Programmierung (en: Structured Programming) ..... 121
D.54	Geeignete Programmiersprachen (en: Suitable Programming Languages)..... 121
D.55	Zeit-Petri-Netze (en: Time Petri Nets)..... 122
D.56	Walkthroughs/Entwurfsüberprüfungen (en: Walkthroughs/Design Reviews) ..... 123
D.57	Objektorientierte Programmierung (en: Object Oriented Programming)..... 123
D.58	Rückverfolgbarkeit (en: Traceability)..... 124
D.59	Metaprogrammierung (en: Metaprogramming) ..... 124
D.60	Prozedurale Programmierung (en: Procedural programming)..... 125
D.61	Sequentielle Funktionslisten (en: Sequential Function Charts – SFC) ..... 125
D.62	Kontaktplan (en: Ladder Diagram) ..... 125
D.63	Funktionsblockdiagramm (en: Functional Block Diagram)..... 126
D.64	Zustandsliste oder Zustandsdiagramm (en: State Chart or State Diagram)..... 126
D.65	Datenmodellierung (en: Data modelling)..... 126
D.66	Kontrollflussdiagramm/Kontrollflussgraph (en: Control Flow Diagram/Control Flow Graph) ..... 126
D.67	Ablaufdiagramm (en: Sequence diagram)..... 128
D.68	Tabellarische Spezifikationsverfahren (en: Tabular Specification Methods) ..... 128
D.69	Anwendungsspezifische Sprache (en: Application specific language) ..... 128
D.70	UML (Unified Modelling Language)..... 129

	Seite
D.71 Domänenspezifische Sprachen (en: Domain specific languages).....	130
Literaturhinweise .....	131
<b>Bilder</b>	
Bild 1 – Software, Übersicht über das Vorgehen .....	10
Bild 2 – Darstellung der bevorzugten Organisationsstruktur.....	20
Bild 3 – Beispielhafter Entwicklungs-Lebenszyklus 1 .....	25
Bild 4 – Beispielhafter Entwicklungs-Lebenszyklus 2 .....	26
<b>Tabellen</b>	
Tabelle 1 – Beziehung zwischen Werkzeugklasse und anwendbarem Abschnitt .....	39
Tabelle A.1 – Fragen des Lebenszyklus und der Dokumentation (5.3) .....	68
Tabelle A.2 – Software-Anforderungsspezifikation (7.2).....	70
Tabelle A.3 – Software-Architektur (7.3) .....	71
Tabelle A.4 – Software-Entwurf und -Implementierung (7.4).....	72
Tabelle A.5 – Verifikation und Testen (6.2 und 7.3).....	73
Tabelle A.6 – Integration (7.6).....	73
Tabelle A.7 – Testen der Gesamtsoftware (6.2 und 7.7) .....	74
Tabelle A.8 – Software-Analysetechniken (6.3) .....	74
Tabelle A.9 – Software-Qualitätssicherung (6.5).....	74
Tabelle A.10 – Software-Wartung (9.2).....	75
Tabelle A.11 – Techniken für die Datengenerierung (8.4) .....	75
Tabelle A.12 – Codierstandards .....	76
Tabelle A.13 – Dynamische Analyse und Testen.....	76
Tabelle A.14 – Funktions-/Black-Box-Tests .....	77
Tabelle A.15 – Text-Programmiersprachen .....	77
Tabelle A.16 – Diagrammartige Sprachen für Anwendungsalgorithmen .....	78
Tabelle A.17 – Modellierung.....	78
Tabelle A.18 – Leistungstests .....	78
Tabelle A.19 – Statische Analyse.....	79
Tabelle A.20 – Komponenten .....	79
Tabelle A.21 – Testabdeckung für Code.....	80
Tabelle A.22 – Objektorientierte Software-Architektur .....	81
Tabelle A.23 – Objektorientierter detaillierter Entwurf.....	81
Tabelle B.1 – Spezifikation der Rolle des Anforderungsmanagers.....	82
Tabelle B.2 – Spezifikation der Rolle des Entwerfers .....	83
Tabelle B.3 – Spezifikation der Rolle des Implementierers.....	84
Tabelle B.4 – Spezifikation der Rolle des Testers.....	85
Tabelle B.5 – Spezifikation der Rolle des Verifizierers .....	86
Tabelle B.6 – Spezifikation der Rolle des Integrators .....	87
Tabelle B.7 – Spezifikation der Rolle des Validierers .....	88
Tabelle B.8 – Spezifikation der Rolle des Gutachters.....	89

	Seite
Tabelle B.9 – Spezifikation der Rolle des Projektmanagers .....	90
Tabelle B.10 – Spezifikation der Rolle des Konfigurationsmanagers.....	90
Tabelle C.1 – Zusammenfassung der Dokumentenkontrolle .....	91