



Bundessprachenamt

Referat SMD 6 beim BAAINBw

Auftragsnummer 2023U-06508

- Übersetzung aus dem Englischen ins Deutsche**
- Textvergleich**
- Einsprachiges Lektorat**
- Inhaltsangabe**
- Maschinelle Übersetzung aus dem ins**

Originaltitel

AQAP-2210-SRD.1 - NATO GUIDANCE ON THE USE OF AQAP-2210 NATO SUPPLEMENTARY SOFTWARE QUALITY ASSURANCE REQUIREMENTS TO AQAP-2110 OR AQAP-2310 - Edition B Version 1

Zieltitel

AQAP-2210-SRD.1 - NATO-LEITLINIEN FÜR DIE ANWENDUNG VON AQAP-2210 "NATO-ZUSATZANFORDERUNGEN ZU AQAP-2110 ODER AQAP-2310 FÜR DIE QUALITÄTSSICHERUNG BEI SOFTWARE" - Ausgabe B, 1. Fassung

NORMENBEZOGENES DOKUMENT

AQAP-2210-SRD.1

**NATO-Leitlinien für die Anwendung von
AQAP-2210
„NATO-Zusatzanforderungen zu AQAP-2110
oder AQAP-2310
für die Qualitätssicherung bei Software“**

Ausgabe B, 1. Fassung

AUGUST 2022

[LOGO]

NORDATLANTIKVERTRAGSORGANISATION (NATO)

**Veröffentlicht durch
NATO-STANDARDISIERUNGSSAMT (NSO)
© NATO/OTAN**

LEERSEITE

NORDATLANTIKVERTRAGSORGANISATION (NATO)

NATO-STANDARDISIERUNGSAMT (NSO)

NATO-BEKANNTGABESCHREIBEN

29. August 2022

1. Das beigefügte normenbezogene Dokument AQAP-2210-SRD.1, Ausgabe B, 1. Fassung, NATO-LEITLINIEN FÜR DIE ANWENDUNG VON AQAP-2210 „NATO-ZUSATZANFORDERUNGEN ZU AQAP-2110 ODER AQAP-2310 FÜR DIE QUALITÄTSSICHERUNG BEI SOFTWARE“, wurde in Verbindung mit AQAP-2210 von den in der Lebenszyklusmanagementgruppe vertretenen Staaten genehmigt und wird hiermit bekanntgegeben.
2. Die Druckschrift AQAP-2210-SRD.1, Ausgabe B, 1. Fassung tritt bei Eingang in Kraft und ersetzt die Druckschrift AQAP-2210-SRD.1, Ausgabe A, 1. Fassung, die gemäß den vor Ort für die Vernichtung von Dokumenten geltenden Verfahren zu vernichten ist.
3. Dieses NATO-Standardisierungsdokument wird von der NATO herausgegeben. Über seine Vervielfältigung ist die NATO in Kenntnis zu setzen. Die NATO erhebt zu keinem Zeitpunkt Gebühren für ihre Standardisierungsdokumente, und diese sind nicht für den Verkauf bestimmt. Sie können aus der Datenbank für NATO-Standardisierungsdokumente (<https://nso.nato.int/nso/>) oder über die nationalen Standardisierungsstellen bezogen werden.
4. Diese Druckschrift ist gemäß den Bestimmungen von Dokument C-M(2002)60 zu behandeln.

[Unterschrift]

Dimitrios SIGOULAKIS
Generalmajor, GRC (A)
Direktor, NATO-Standardisierungsamt

LEERSEITE

INHALTSVERZEICHNIS

VORWORT

KAPITEL 1 EINLEITUNG

- 1.1 Allgemeines
- 1.2 Zweck
- 1.3 Anwendungsbereich

KAPITEL 2 ÜBEREINSTIMMUNG MIT DIESER DRUCKSCHRIFT

- 2.1 Erfüllung der Anforderungen
- 2.2 Anmerkungen und Leitlinien

KAPITEL 3 AUFBAU DER ANFORDERUNGEN IN DER AQAP-2210

- 3.1 Aufbau
- 3.2 Bezugsdokumente
 - 3.2.1 Normative Bezugsdokumente
 - 3.2.2 Informative Bezugsdokumente
- 3.3 Definitionen
- 3.4 Abkürzungen

KAPITEL 4 ALLGEMEINE QUALITÄTSSICHERUNGSANFORDERUNGEN FÜR SOFTWARE

- 4.1 Zusatzanforderungen zu AQAP-2110 für die Qualitätssicherung bei Software
- 4.2 Zusatzanforderungen zu AQAP-2310 für die Qualitätssicherung bei Software
- 4.3 Anpassung
- 4.4 Nicht anpassbare Absätze

KAPITEL 5 NATO-SPEZIFISCHE QUALITÄTSSICHERUNGSANFORDERUNGEN FÜR SOFTWARE

- 5.1 Allgemeines
- 5.2 Software-Qualitätssicherungsplan des Projekts (PSQP)
- 5.3 Untersuchung der Softwarekritikalität
- 5.4 Softwarequalitätsmodell
- 5.5 Identifizierung und Prüfung der Softwareanforderungen
- 5.6 Softwareentwurfs- und -entwicklungsprozess
- 5.7 Management
 - 5.7.1 Rollen, Verantwortlichkeiten und Befugnisse in der Organisation
 - 5.7.2 Unterauftragnehmermanagement
 - 5.7.3 Software-Konfigurationsmanagement (SCM)
 - 5.7.4 Fremdentwickelte/Off-The-Shelf-Software
 - 5.7.5 Infrastruktur
 - 5.7.6 Managementbewertung
- 5.8 Bewertung, Verifizierung und Validierung (EVV)
 - 5.8.1 Allgemeines
 - 5.8.2 Test-Durchführung
 - 5.8.3 Prüfungen (Reviews)
- 5.9 Wartung/Pflege

**ANHANG A Vorlage für die Begründung von Anpassungen im Rahmen der
AQAP-2210**

VORWORT

Dieses Dokument wurde als Informationsquelle und Leitlinie für die Anwendung der AQAP-2210, Ausgabe B, 1. Fassung, „NATO-Zusatzanforderungen zu AQAP-2110 oder AQAP-2310 für die Qualitätssicherung bei Software“ erstellt und herausgegeben.

Es soll zur Einheitlichkeit der Auslegung dieser Anforderungen beim Auftragnehmer und beim Auftraggeber beitragen. Es ist nicht als Beschaffungsdokument vorgesehen. Sein Inhalt ist weder rechtlich noch vertraglich bindend, noch ergänzt oder ersetzt es Anforderungen der AQAP-2210 oder setzt diese außer Kraft. Ausfertigungen dieser Druckschrift können der Industrie zur Verfügung gestellt werden, um ihr die Anwendung und das Verständnis der AQAP-2210 zu erleichtern.

Jeder Abschnitt (und Unterabschnitt) der AQAP-2210 ist in dieser Druckschrift mit dem jeweiligen Titel wiedergegeben, gefolgt von der zugehörigen Leitlinie bzw. dem Satz „Keine Leitlinien erforderlich“.

Aufgrund der Vielfalt möglicher Gegebenheiten sollten diese Leitlinien weder als umfassend angesehen noch als Vorgabe für bestimmte Mittel und Methoden zur Erfüllung der vertraglichen Forderungen aufgefasst werden. Den Führungsverantwortlichen muss bewusst sein, dass auch andere Mittel oder Methoden angewandt werden könnten, um diese Anforderungen zu erfüllen.

KAPITEL 1 EINLEITUNG

1.1 Allgemeines

Keine Leitlinien erforderlich.

1.2 Zweck

Zusätzlich zu den Anforderungen a bis e, die sich nur auf den Software-Entwicklungsprozess beziehen, behandelt diese Druckschrift auch das Verhältnis von System zu Software. Die zusätzlichen Anforderungen f und g regeln die sinnvolle Beteiligung des Software Engineering an der Systementwicklung und die Behandlung von system-/ softwarekritischen Vorgaben wie Fragen der Sicherheit.

1.3 Anwendungsbereich

Keine Leitlinien erforderlich.

KAPITEL 2 ÜBEREINSTIMMUNG MIT DIESER DRUCKSCHRIFT
--

2.1 Erfüllung der Anforderungen

Keine Leitlinien erforderlich.

2.2 Anmerkungen und Leitlinien

Keine Leitlinien erforderlich.

KAPITEL 3 AUFBAU DER ANFORDERUNGEN IN DER AQAP-2210

3.1 Aufbau

Keine Leitlinien erforderlich.

3.2 Bezugsdokumente

3.2.1 Normative Bezugsdokumente

Keine Leitlinien erforderlich.

3.2.2 Informative Bezugsdokumente

Keine Leitlinien erforderlich.

3.3 Definitionen

Keine Leitlinien erforderlich.

3.4 Abkürzungen

Keine Leitlinien erforderlich.

KAPITEL 4 ALLGEMEINE QUALITÄTSSICHERUNGSANFORDERUNGEN FÜR SOFTWARE
--

4.1 Zusatzanforderungen zu AQAP-2110 für die Qualitätssicherung bei Software

Keine Leitlinien erforderlich.

4.2 Zusatzanforderungen zu AQAP-2310 für die Qualitätssicherung bei Software

Keine Leitlinien erforderlich.

4.3 Anpassung

1. Die Ersetzung des Modalverbs „should“ (sollen/sollte) durch das Modalverb „shall“ (als verbindliche Forderung, im Deutschen etwa: müssen, ist zu/sind zu, oder Präsens) im Anwendungsbereich der Norm AS9115 hat zu der Notwendigkeit geführt, die Forderung nach „Anpassung“ in die AQAP-2210 aufzunehmen.

2. Ziel des Anpassungsprozesses sollte die Streichung (vollständig oder teilweise), Herabstufung, Änderung oder Ergänzung von Prozessen (Aktivitäten und Aufgaben) der AQAP-2210 sein, um die Effektivität und Effizienz der technischen und Managementprozesse für das spezifische Softwareprojekt zu optimieren.

3. In ähnlicher Weise sollte die Anpassung auch auf die Lebenszyklusdaten angewendet werden, die von jedem Prozess (Aktivität/Aufgabe) erzeugt werden. Die erstellten Informationen sollten nur dann in ein Dokument aufgenommen werden, wenn Informationen in ein zu lieferndes Dokument umgesetzt werden müssen, wobei zu berücksichtigen ist, dass diese Umsetzung einen zusätzlichen Zeit- und Arbeitsaufwand erfordert. Bei einigen Projekten kann es wünschenswert sein, Liefergegenstände zusammenzufassen, wodurch dann weniger zu liefernde Dokumente zu kontrollieren, nachzuverfolgen und zu bearbeiten sind; insofern ist es wichtig, auch den Zeitpunkt der Erstellung, die beabsichtigten Zielgruppen und die Häufigkeit der Aktualisierung der zu liefernden Dokumente zu berücksichtigen.

4. In dieser Leitlinie wird nicht erörtert, zu welchem Zeitpunkt (z. B. vorvertragliche Anpassung, Anpassung durch Verhandlungen, nachvertragliche Anpassung usw.) oder von wem (Auftragnehmer, Auftraggeber oder andere) die Anpassung von AQAP-2210 durchgeführt wird.

5. Die Anpassungsziele sollten Qualitätsverbesserungen umfassen und Nutzen und Effizienz berücksichtigen. Insofern sollten alle an einem Projekt beteiligten Parteien an der Anpassung beteiligt sein, und die Entscheidungen sollten zwischen Kunde und Auftragnehmer abgestimmt und in den Vertragsunterlagen abgebildet werden; ein Teamansatz unter Einbeziehung aller am Lebenszyklus beteiligten Parteien könnte zu erheblichen Vorteilen führen.

6. Für die Durchführung des Anpassungsprozesses sollte die Software kategorisiert werden, um den zu betreibenden Aufwand an Tests, Prüfungen oder Kontrollen zu bestimmen. Diese Kategorisierung sollte auch auf Grundlage möglicher Bedrohungen für die Verfügbarkeit,

Integrität und Vertraulichkeit erfolgen, die Auswirkungen auf die Geschäftsabläufe, die Sicherheit von Personen oder Sachschäden haben können.

7. Im Hinblick auf die Anpassung sollten vier wesentliche Bereiche berücksichtigt werden:
 - a. Sowohl die Prozesse als auch die erzeugten Daten können und sollten angepasst werden.
 - b. Es muss denjenigen, die die Anpassungsprozesse durchführen, klar sein, welche der ausgewählten Anforderungen sich für die Anpassung eignen und welche nicht.
 - c. Anpassung ist kein Selbstzweck, sondern es muss einen logischen Grund für die Durchführung eines Anpassungsprozesses geben. Die Notwendigkeit der Anpassung der Anforderungen an ein bestimmtes Projekt wird im Allgemeinen durch Überlegungen wie Kritikalitätsstufe der Softwaresicherheit, Anforderungen an die Wartbarkeit, spezifische Entwicklungsverfahren wie agile Methoden oder formale Methoden, spezifische Randbedingungen des Projekts (multinational, Festpreis, Lieferintervalle, nationale Sicherheitsüberlegungen usw.), Produkteigenschaften usw. bestimmt.
 - d. Anpassungen sind potenziell gefährlich, wenn sie nicht ordnungsgemäß kontrolliert werden, und können zu einer Auswahl von Anforderungen führen, die zwar kurzfristig ein schnelles und kostengünstiges Projekt unterstützen, aber letztendlich zu einem Produkt führen können, das den Vertrag nicht erfüllt. Daher ist es während des gesamten Anpassungsprozesses von entscheidender Bedeutung, stets die Integrität und die Sicherheitsaspekte des Endprodukts zu achten.

TREIBENDE FAKTOREN FÜR DIE ANPASSUNG

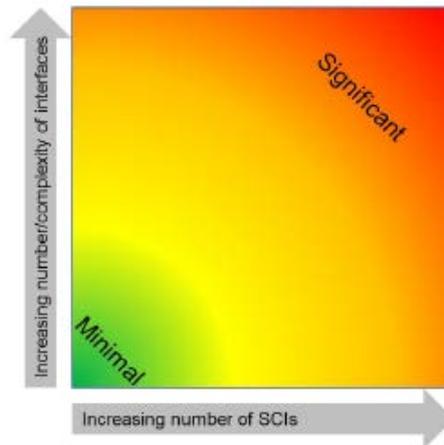
0. Der Anpassungsprozess wird durch Projekt- und Produktmerkmale (treibende Faktoren für die Anpassung) bestimmt und sollte stets ohne Beeinträchtigung der Qualität, Sicherheit und Integrität des Softwareprodukts durchgeführt werden.

1. Bei der Entscheidungsfindung zur Anpassung ist es wichtig, eine Risikoanalyse durchzuführen, um kurzfristige Einsparungen bei Kosten und Zeit gegen langfristige Risiken abzuwägen. So kann beispielsweise die Streichung der für die Softwareunterstützung benötigten Produkte im Rahmen der Anpassung Zeit und Geld bei der Erstentwicklung sparen, sich aber erheblich negativ auf die langfristigen Kosten der Softwareunterstützung auswirken. In ähnlicher Weise kann die Streichung von Softwareproduktbewertungen, Softwaretests und damit zusammenhängenden Aktivitäten kurzfristig Zeit und Geld sparen, aber zu einer Qualitätsminderung und kostspieliger und zeitaufwändiger Nacharbeit führen, sobald die Software geliefert und in Betrieb genommen wurde.

2. Es gibt verschiedene Arten von treibenden Faktoren für die Anpassung, die sich auf verschiedene Aspekte beziehen, wie z. B. Zuverlässigkeit und Sicherheit, Randbedingungen und Methoden der Softwareentwicklung, Produktqualität und Geschäftsziele:

- a. Anpassungen aufgrund von Zuverlässigkeits- und Sicherheitsaspekten sollten auf der Auswahl der Anforderungen im Zusammenhang mit der Verifizierung, Validierung und dem durch die Kritikalität der Software erforderlichen Nachweisniveau basieren.
- b. Bei Anpassungen aufgrund von Randbedingungen und Methoden der Softwareentwicklung sollten die besonderen Merkmale der zu entwickelnden Software und der Entwicklungsumgebung berücksichtigt werden. Die Art der Softwareentwicklung und das Zielsystem sollten ebenfalls berücksichtigt werden; spezifische Anforderungen an Verifizierung, Review und Prüfung (Inspektion) sollten beispielsweise erhoben werden, wenn eine vollständige Validierung am Zielgerät nicht möglich ist oder wenn Leistungsziele schwer zu erreichen sind.

Einflussfaktoren auf die Bedeutung der Integration

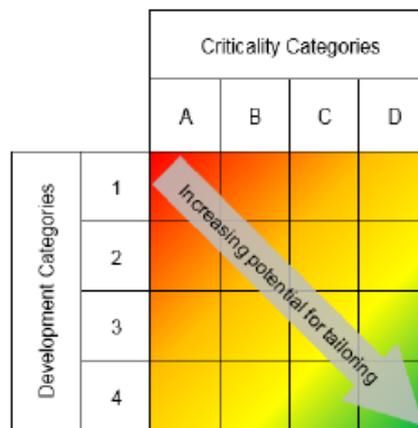


Glossar:

Significant	Sehr bedeutsam
Minimal	Kaum bedeutsam
Increasing number/complexity of interfaces	Zunehmende Anzahl/Komplexität der Schnittstellen
Increasing number of SCIs	Zunehmende SCI-Anzahl (Software-Konfigurationseinheiten)

- c. Anpassungen aufgrund von Produktqualität und Geschäftszielen sollten durch Auswahl der Qualitätsanforderungen an das Produkt auf der Grundlage der vom Kunden festgelegten Qualitätsziele für das Produkt erfolgen.

3. Das nachstehende Diagramm zeigt das steigende Potenzial für Anpassungen, wenn Kritikalitätsstufe und Entwicklungsumfang abnehmen.



Glossar:

Criticality Categories	Kritikalitätskategorien
Development Categories	Entwicklungskategorien
Increasing potential for tailoring	Steigendes Potenzial für Anpassungen

4. Weitere treibende Faktoren für den Anpassungsprozess existieren im Zusammenhang mit technischen, operativen und Managementaspekten:

a) Technische Faktoren:

- Neuheit des Anwendungsbereichs
- Komplexität der Software und des Systems
- Kritikalitätsstufe
- Größe der Software
- erforderliche Wiederverwendbarkeit der zu entwickelnden Software
- Schnittstelle zu Systementwicklungsprojekten
- Grad der Nutzung von handelsüblicher oder vorhandener Software
- Reife der handelsüblichen Software und Vollständigkeit oder Konstanz der Nutzeranforderungen

b) Operative Faktoren:

- Art der Anwendung
- Anzahl der potenziellen Nutzer der Software
- Kritikalität der Software gemessen an den Folgen ihres Ausfalls
- erwartete Lebensdauer der Software
- Anzahl der Standorte, an denen die Software verwendet wird
- Randbedingungen von Betrieb, Wartung/Pflege, Migration und Außerbetriebnahme

c) Managementfaktoren:

- Für die Softwareentwicklung erforderlicher Zeit- und Arbeitsaufwand
- Haushalterische Anforderungen für Implementierung und Betrieb der Software
- Akzeptables Risikoniveau für das Projekt
- Art des Lebenszyklus
- Anforderungen an die Liefertermine der Software
- erforderliche Personenzahl für die Entwicklung, den Betrieb und die Pflege der Software
- Komplexität der Organisation
- Erfahrung des Auftragnehmers
- Finanzielle Mittel

5. Die nachstehende Tabelle zeigt (nur als einfache Veranschaulichung der Wechselbeziehung zwischen Faktoren und Prozessen) eine nicht erschöpfende Korrelation zwischen den „Faktoren/Merkmalen“ und den hauptsächlich betroffenen „Prozessen“.

	Faktor/ Merkmal	Beschaffung	Nachschub/ Versorgung	Entwicklung	Betrieb	Wartung/ Pflege	Dokumentation	Konfigurations- management	Qualitäts- sicherung	Verifizierung	Validierung	Gemeinsame Prüfung	Audit	Problemlösung	Management	Infrastruktur	Ausbildung/ Schulung
Produkt	Komplexität des Systems			X	X	X	X	X	X		X						
	Komplexität der Software			X	X	X	X	X	X		X						
	Größe der Software			X					X						X		
	Randbedingungen der Hardware- /Softwareressourcen			X					X	X							
	Sicherheit im Sinne der technisch-/ betrieblichen Sicherheit (Safety)			X	X		X	X	X	X	X	X		X			
	Sicherheit im Sinne des Schutzes vor äußeren Angriffen (Security)			X	X		X	X									
	Zuverlässigkeit			X	X			X	X	X	X	X					
	Wartbarkeit			X		X	X	X	X								
	Interoperabilität			X	X		X	X				X					
	Verfügbarkeit				X							X					
	Gebrauchstauglichkeit			X	X		X					X					
	Wiederverwendungsgrad			X						X	X						
Nutzung neuer Technologien			X						X							X	X
Projekt	Randbedingungen des Projektplans	X	X									X	X		X		
	Randbedingungen der Projektressourcen	X	X									X			X		
	Vertragliche Komplexität	X	X	X				X	X			X			X		

ANPASSUNGSSCHRITTE

1. Die wichtigsten Schritte zur Anpassung der AQAP-2210:

a) Identifizierung des Umfelds und der Merkmale des Projekts

Die Merkmale des Projektumfelds und die Umstände, die die Anpassung beeinflussen werden, sollten identifiziert und festgehalten werden.

Darunter fällt ohne Anspruch auf Vollständigkeit Folgendes:

- Konstanz und Vielfalt der Einsatzanforderungen
- Risiken (kommerzielle oder leistungsmäßige) für die Belange der interessierten Parteien
- Neuartigkeit, Größe, Komplexität und Kritikalität
- Bereitstellung von Hardware-Ressourcen
- Verwendung einer bestimmten Sprache
- Einsatz eines unabhängigen Verifikations- und Validierungsverantwortlichen
- Fragen der Integrität wie Sicherheit (Safety und Security), Datenschutz, Nutzbarkeit, Verfügbarkeit
- neue Technologiemöglichkeiten
- Profil der verfügbaren Haushalts- und Organisationsressourcen
- Beginn und Dauer der Nutzung
- Anzahl der beteiligten Personen und Parteien
- Aufgaben, Zuständigkeiten, Verantwortlichkeiten und Befugnisse im gesamten Lebenszyklus
- einschlägige Gesetze und Vorschriften
- Notwendigkeit der Übereinstimmung mit anderen Normen

b) Anforderung von Beiträgen

Beiträge der von den Anpassungsentscheidungen betroffenen Organisationen sollten angefordert werden.

Jede Organisation sollte die Möglichkeit haben, ihre Erfordernisse darzulegen und das gemeinsame Fachwissen aller Teilnehmer zu nutzen; Nutzer und Unterstützungspersonal sollten ebenfalls an der Anpassung beteiligt sein.

Diese umfassen u.a.:

- die am System beteiligten Interessengruppen
- die an der von der Organisation getroffenen Vereinbarung interessierten Parteien
- die beitragenden Funktionsbereiche der Organisation.

c) Auswahl der Entwicklungsstrategie

Die Entwicklungsstrategie mit der größten Relevanz und Eignung (z. B. Wasserfallmodell, Spiralmodell, evolutionäres oder inkrementelles Vorgehensmodell) sollte festgelegt werden. Jede derartige Strategie schreibt bestimmte Prozesse (Aktivitäten und Aufgaben) vor, die nacheinander, wiederholt und gemeinsam durchgeführt werden können.

d) Auswahl der Phasen und Prozesse

Die relevanten Phasen des Lebenszyklusmodells sowie die Eingangs- und Ausgangsbedingungen des Phasenübergangs und ihre Beziehung zueinander (seriell, parallel, ganz oder teilweise kombiniert) sollten identifiziert werden.

Die Lebenszyklusprozesse (Aktivitäten und Aufgaben) und die zugehörigen anzupassenden Lebenszyklusdaten sollten ausgewählt werden.

e) Dokumentieren der Anpassungsentscheidungen und der Begründung

Alle Anpassungsentscheidungen sollten zusammen mit der Begründung für Entscheidungen dokumentiert werden.

Der Auftragnehmer muss das Grundprinzip seiner Anpassung begründen und die Ergebnisse der Anpassung sollten im PSQP dokumentiert werden.

Ein vorgeschlagenes Muster für die Dokumentation der Anpassungsentscheidungen ist **Anhang A** zu entnehmen.

4.4 Nicht anpassbare Absätze

Keine Leitlinien erforderlich.

KAPITEL 5	NATO-SPEZIFISCHE QUALITÄTSSICHERUNGSANFORDERUNGEN FÜR SOFTWARE
------------------	---

5.1 Allgemeines

1. Die AQAP-2210 setzt normalerweise voraus, dass ein dokumentiertes Qualitätsmanagementsystem (QMS) der Organisation, auch in Verbindung mit den Softwareaspekten, besteht, das nicht nur die technischen Prozesse von Softwareentwurf und -entwicklung, sondern auch die Managementprozesse enthält.
2. Das QMS sollte den Bereich der Software abdecken, den der Auftragnehmer erstellt. Je nach Art der Anwendung, Projektgröße, Anzahl der Beteiligten usw. können unterschiedliche Methoden, Verfahren und Werkzeuge erforderlich werden.
3. Eine regelmäßige, systematische und dokumentierte Bewertung des Zustandes und der Angemessenheit des QMS sollte durch oder im Namen der obersten Führungsebene erfolgen, um sicherzustellen, dass deren Ziele erreicht werden, und um Abweichungen oder Unregelmäßigkeiten in den Systemelementen festzustellen, die Verbesserung erfordern.
4. Das QMS ist wirksam, wenn es die Anforderungen der AQAP-2210 und alle zusätzlichen durch den Vertrag festgelegten Anforderungen unterstützt.
5. Die projektbezogenen Software-Qualitätsmanagementtätigkeiten sollten die zur erfolgreichen Durchführung des Projekts erforderlichen Tätigkeiten der Planung und Umsetzung umfassen. Die in Abschnitt 5.1 erwähnten Projektarbeiten (Punkte 3a, 3b, 3c, 3d) werden in den Abschnitten 5.5 bis 5.8 ausführlich behandelt. In jedem Abschnitt sind Leitlinien zu diesen Tätigkeiten enthalten.
6. Es ist wichtig zu beachten, dass sich der Punkt 3c auch auf nicht auszuliefernde Software bezieht (d. h. sie ist verfügbar, aber kein vertraglicher Liefergegenstand, und kann bei der Entwicklung zu liefernder Software verwendet werden, z. B. Emulatoren, Werkzeuge für automatisch generierten Code und/oder automatisch generierte Tests, Test-Harnische (test harnesses) und Treiberprogramme, Vorschalttroutinen (stub routines) usw.); es ist von entscheidender Bedeutung, dass diese Software dem Konfigurationsmanagement unterstellt wird, da sie unmittelbare Auswirkungen auf die Integrität der zu liefernden Software hat.
7. Die Tiefe der projektbezogenen Software-Qualitätsmanagementtätigkeiten wird durch die vertraglichen Anforderungen und Randbedingungen wie Komplexität, Kritikalität, Größe, Beteiligung des Auftraggebers usw. beeinflusst. Aus diesem Grund sollte der Auftragnehmer als Voraussetzung für die Planung der Aktivitäten eine formelle Vertragsprüfung (Review) durchführen, um sicherzustellen, dass alle Anforderungen und Randbedingungen klar definiert und verstanden worden sind.
8. Die Bewertung der Aktivitäten durch den Auftraggeber sollte zunächst den objektiven Nachweis der eigenen Prüfungen (Reviews) des Auftragnehmers miteinbeziehen. Wenn kein objektiver Nachweis vorhanden ist, dass solche Prüfungen durchgeführt worden sind, sollte dies als ein schwerwiegender Mangel des Qualitätssystems und demzufolge als Risiko angesehen werden.

5.2 Software-Qualitätssicherungsplan des Projekts (PSQP)

1. Der PSQP und sein Inhalt sollten vom Auftraggeber und vom Auftragnehmer als ein Zeichen für Einvernehmen, Verpflichtung und Übereinstimmung mit den Qualitätsanforderungen des Vertrags anerkannt werden.
2. Auftragnehmer sollten in der frühestmöglichen Phase des Vertrags mit der Planung ihrer qualitätsrelevanten Aktivitäten beginnen.
3. Qualitätsberichte können in Form von EVV-Berichten (vgl. Abschnitt 5.8), Testergebnissen, Fehlerbehebungsberichten usw. vorgelegt werden. Sie können die offiziellen Ergebnisse der Aktivitäten sowohl des Haupt- als auch des Unterauftragnehmers sein.
4. Der PSQP sollte „vertragspezifische“ Qualitätssicherungsaktivitäten behandeln und nicht die QMS-Forderungen nachbilden, wie sie im Qualitätshandbuch/der Qualitätsdokumentation des Auftragnehmers aufgeführt sind; eine Bezugnahme auf diese QMS-Forderungen könnte jedoch im PSQP erforderlich sein.
5. Mit der Qualitätsplanung kann die Anwendung des QMS auf ein bestimmtes Projekt, Produkt oder einen bestimmten Vertrag zugeschnitten werden.
6. Ein PSQP kann aufgrund einer Ausschreibung/Aufforderung zur Angebotsabgabe oder gemäß Vertrag erforderlich sein und sollte als Grundlage für den Softwareentwurfs- und -entwicklungsprozess erstellt werden.
7. Ein mögliches Layout des PSQP ist in AQAP-2105 dargestellt.
8. Weitere Leitlinien können ISO/IEC/IEEE 90003:2018 (Abschnitt 8.1.1) entnommen werden.

5.3 Untersuchung der Softwarekritikalität

Keine Leitlinien erforderlich.

5.4 Softwarequalitätsmodell

Keine Leitlinien erforderlich.

5.5 Identifizierung und Prüfung der Softwareanforderungen

1. Die Softwareanforderungen können aus einem vom Auftraggeber angedeuteten (aber nicht unbedingt spezifizierten) Bedarf abgeleitet werden. Häufig ist sich der Auftragnehmer über die Problemstellung des Auftraggebers und das Anwendungsgebiet nicht vollständig im Klaren. Beide Vertragsparteien können zusammenarbeiten, um eine formelle vertragliche Vereinbarung über den Funktionsumfang der fertigen Software zu erzielen.
2. Der Schlüssel zu einem effektiven Softwareentwurf und einer effektiven Softwareentwicklung sollte für den Auftragnehmer wie auch den Auftraggeber darin bestehen, ein gemeinsames Verständnis der Anforderungen zu erzielen. Deshalb sollte der Auftragnehmer sicherstellen, dass die Softwareanforderungen dergestalt beschrieben werden, dass ihre Deutung keine Fragen offenlässt. Alle Lücken, Missverständnisse oder

Unstimmigkeiten in den Anforderungen sollten so früh wie möglich im Softwareentwurfs- und -entwicklungsprozess angesprochen werden, wenn sie noch leichter zu korrigieren sind. Der Auftragnehmer sollte sich auch davon überzeugen, dass jede Anforderung so definiert ist, dass ihre Erfüllung letztendlich durch eine vorgeschriebene Methode validiert werden kann. Bestehen hieran Zweifel, sollte die Angelegenheit dem Auftraggeber zur Kenntnis gebracht werden.

3. Oft sind die Softwareanforderungen von Anforderungen einer übergeordneten Ebene abgeleitet (System oder Untersystem); daher ist es wichtig sicherzustellen, dass alle geltenden übergeordneten Forderungen korrekt in Softwareanforderungen umgesetzt und keine neuen Forderungen eingebracht worden sind.

4. „Entwicklungsvorgaben“ sind Einschränkungen für den Entwurfs- und Entwicklungsprozess, durch die größere Entwurfsverantwortung auf die Organisation übergeht, welche die Einschränkungen festlegt. Sie müssen deshalb von den Softwarequalitätsmerkmalen getrennt werden. Beispiele sind:

- Entwurfsnormen und -regeln
- Sprachen
- Computer-Hardware und
- vom Auftraggeber beigestellte Software.

5. Softwarequalitätsdefinitionen sind in ISO/IEC 25010:2011 (2017 *bestätigt*) zu finden. Diese umfassen:

- Funktionelle Eignung
- Leistungseffizienz
- Kompatibilität
- Gebrauchstauglichkeit
- Zuverlässigkeit
- Sicherheit („Security“)
- Instandsetzungsfreundlichkeit und
- Übertragbarkeit.

6. Weitere Leitlinien können ISO/IEC/IEEE 90003:2018 (§ 8.2.2) entnommen werden.

5.6 Softwareentwurfs- und -entwicklungsprozess

1. Der Softwareentwurfs- und -entwicklungsprozess hat eine starke Auswirkung auf die Qualität des Softwareprodukts.

2. Softwareentwurfs- und -entwicklungsmodelle sind vereinfachte abstrakte Darstellungen eines systematischen Ansatzes für den Softwareentwurfs- und -entwicklungsprozess; sie bilden zusammen mit Methoden und Werkzeugen wichtige Elemente des Qualitätsmanagements und sollten folgende qualitätsbezogene Kriterien erfüllen:

- Vereinfachung des Entwicklungsprozesses zur Sicherstellung von Transparenz und Kontrolle
- Beschreibung der Software- und Systemintegration
- Beschreibung der Softwaresystemarchitektur
- Nutzung anerkannter Grundsätze des Software Engineering
- Nutzung des Datenfeedbacks vorausgegangener Entwürfe
- genaue Beschreibung der Tätigkeiten und ihrer erwarteten Ergebnisse

- Identifizierung der für Qualität und Projekterfolg kritischen Aufgaben
- Festlegung und chronologische Zuordnung von Kontrollpunkten, an denen der ordnungsgemäße Ablauf des Prozesses und die korrekte Übertragung von Ergebnissen verifiziert werden kann
- Beschreibung, wie ungeplante Tätigkeiten kontrolliert werden
- Bereitstellung eindeutiger Kriterien für Anfang und Ende von allen Prozessen
- genaue Bezeichnung und Zuordnung aller qualitätsbezogenen Aufgaben innerhalb der projektspezifischen Organisationsstrukturen
- Anwendung von bewährten und qualifizierten konstruktiven und analytischen Qualitätsmaßnahmen
- Bereitstellung von qualitätsbezogenen Daten für das wirksame Management des Entwicklungsprozesses
- Zuordnung von Planungs-, Überwachungs- und Freigabetätigkeiten zu Maßnahmen des Software Engineering
- Verringerung des Risikos durch den Einsatz von Computerressourcen zur größtmöglichen Entlastung der am Softwareentwicklungsprozess Beteiligten von fehleranfälligen, sich wiederholenden Tätigkeiten.

3. Es gibt verschiedene Modelltypen (z. B. Wasserfallmodell, evolutionäres oder inkrementelles Vorgehensmodell, Spiralmodell, usw.), und AQAP-2210 stellt dem Auftragnehmer die Wahl des Entwicklungsmodells frei. Auswahl, Definition und Anwendung eines bestimmten Modells sind abhängig von Komplexität, Kritikalität und Art der zu entwickelnden Software. Jedes ausgewählte Modell kann so zugeschnitten werden, dass es den jeweiligen vertraglichen Anforderungen entspricht. Soweit möglich, sollten internationale oder nationale Normen berücksichtigt werden, in denen diese Modelle definiert sind.

4. Das Modell sollte eine eindeutige Beschreibung aller (technischen) Hauptprozesse (z. B. Entwurf, Kodierung, Validierung usw.) mit allen während des Software-Lebenszyklus angewandten unterstützenden (Projekt-)Prozessen (z. B. Projektmanagement, Konfigurationsmanagement usw.) und organisatorischen (das Projekt ermöglichenden) Prozessen (z. B. Arbeitskräfte, Infrastruktur usw.) liefern. Die Beschreibung der Prozesse sollte nicht nur die Identifizierung der Aktivitäten und Aufgaben enthalten, sondern auch die Rollen (Architect, Tester usw.), die Eingaben (Inputs), die Ergebnisse (Outputs), die Kriterien für Anfang und Ende, die Kontrollpunkte, (ggf.) die entsprechenden Messungen sowie alle technischen und Managementaspekte. Das Ziel dabei ist, die Komplexität des Softwareentwurfs- und -entwicklungsprozesses zu reduzieren und somit die Transparenz, Integrität und Kontrolle des Softwareprodukts selbst zu verbessern.

5. Der Auftragnehmer sollte für die Softwareentwurfs- und -entwicklungstätigkeiten anerkannte Methoden, Werkzeuge, Ressourcen und Verfahren des Software Engineering verwenden. Der Auftragnehmer sollte außerdem Konventionen für grafische oder formale sprachliche Darstellungs- und Schreibweisen angeben und standardisieren. Die verwendeten Methoden, Werkzeuge, Ressourcen und Verfahren sollten den Softwarelebenszyklus bei Folgendem unterstützen:

- Umsetzung von Softwareanforderungen einschließlich Qualitätsmerkmalen;
- Umwandlung von auftraggeber-/benutzerorientierten Softwarequalitätsanforderungen in Merkmale, die sich am Software Engineering orientieren, und Zuordnung jener Merkmale zu der entsprechenden Entwurfsebene;
- Sicherstellung der Rückverfolgbarkeit auf allen Entwurfs- und Durchführungsebenen;
- Minimierung von Fehlern;
- Unterstützung von EVV-Aktivitäten (Bewertung, Verifizierung und Validierung) während Softwareentwurf und -entwicklung.

6. Die verwendeten Methoden, Werkzeuge und Verfahren sollten bewertet und dokumentiert werden und die anerkannten Grundsätze und Konzepte des Software Engineering unterstützen, die die Softwarequalität beeinflussen.

7. Softwarewerkzeuge können mit speziellen, festgelegten Methoden und Techniken in Verbindung stehen oder andere Aspekte des Software-Lebenszyklus unterstützen. Einige Werkzeuge können phasenunabhängig sein, z. B. jene, die mit dem Konfigurationsmanagement oder den Qualitätssicherungsaktivitäten in Zusammenhang stehen.

8. Softwarewerkzeuge sollten zum Nachweis ihrer Leistung und Integrität anhand einer definierten Methode validiert werden; diese Validierung kann eine oder mehrere der folgenden Maßnahmen umfassen:

- Bescheinigung von einer anerkannten Stelle, dass das Werkzeug speziellen Tests oder Validierungsprozessen unterworfen wurde;
- Feststellung im Einvernehmen mit dem Werkzeuganbieter, dass das Werkzeug die geforderten Kriterien auf Grundlage des Qualitätssystems des Auftragnehmers und der Nachweise entsprechender Tests erfüllt;
- Identifizierung entsprechender Tests, die mit dem Werkzeug und seinen weiteren Versionen durchzuführen sind;
- überwachter Einsatz des Werkzeugs während der Unterstützung der Entwicklung des Softwareprodukts;
- Feedback von einer Nutzergruppe.

9. Entwurfs- und Entwicklungsmodelle stellen die Grundlage für die detaillierte Planung der projektbezogenen Software-Qualitätsmanagementtätigkeiten, einschließlich der terminlichen und finanziellen Aspekte, dar und unterstützen die fortlaufende Verbesserung des Softwareentwurfs- und -entwicklungsprozesses.

10. Weitere Leitlinien können ISO/IEC/IEEE 90003:2018 (Abschnitt 8.3.2.1 und Abschnitt 8.3.2.2) entnommen werden.

5.7 Management

5.7.1 Rollen, Verantwortlichkeiten und Befugnisse in der Organisation

1. Es ist wichtig, die Wechselbeziehungen der organisatorischen Elemente und Gruppen festzulegen, da sich Tätigkeiten überlappen und wiederholen können. Es ist auch wichtig, dass die Organisationsstruktur die Zusammenarbeit und Absprache zwischen Elementen oder Gruppen erkennen lässt und auch auf die Ansprechstellen für den Auftraggeber hinweist.

2. Für die Personen, die Bewertungen, Verifizierungen und Validierungen durchführen, kann der erforderliche Grad der Unabhängigkeit von den Gegebenheiten des jeweiligen Vertrags und/oder des betreffenden Auftragnehmers abhängen. In den meisten Fällen lässt sich geeignetes unabhängiges Personal bei Fachkollegen der Personen finden, die das Softwareprodukt entwickelt oder die Tätigkeit durchgeführt haben, das bzw. die Gegenstand der Bewertung, Verifizierung oder Validierung ist. Manchmal kann es erforderlich sein, dieses Personal aus anderen Bereichen oder Organisationen innerhalb oder außerhalb des Auftragnehmers zu rekrutieren. Wo besondere Unabhängigkeitsforderungen gelten, wie für sicherheitskritische Software, sollten diese im Vertrag definiert werden.

3. Es sollte festgelegt werden, welcher Grad an Unabhängigkeit für die Verifizierungsmaßnahme benötigt wird, abhängig von der Wahrscheinlichkeit eines

unentdeckten Fehlers in einem System oder einer Software, welcher Folgendes verursachen kann:

- Tod oder Verletzungen
- Fehlschlag der Mission
- Katastrophaler Verlust von Gerät oder Schaden daran
- technische Reife und Risiken in Verbindung mit der anzuwendenden Softwaretechnologie
- finanzielle Verluste.

4. Schulungen des Personals (sowohl des Auftraggebers als auch des Auftragnehmers) sind ebenfalls in Erwägung zu ziehen; das mit speziellen Aufgaben betraute Personal sollte auf der Grundlage angemessener Ausbildung/Schulung und/oder Erfahrung qualifiziert werden im Hinblick auf:

- Entwurfsmethoden
- spezielle Programmiersprachen
- Werkzeuge, Techniken
- Computerplattformen und Zielumgebung.

5. Weitere Leitlinien können ISO/IEC/IEEE 90003:2018 (Abschnitt 5.3 und Abschnitt 7.1.2) entnommen werden.

5.7.2 Unterauftragnehmermanagement

1. Der Hauptauftragnehmer sollte sicherstellen, dass im Unterauftrag beschaffte Produkte und Leistungen den Forderungen und Bedingungen des Hauptvertrages entsprechen, selbst wenn das gesamte Softwarepaket im Unterauftrag vergeben wird.

2. Der Auftragnehmer sollte Unterauftragnehmer durch Nutzung eines geeigneten Verfahrens auswählen und sie auf der Grundlage ihrer Fähigkeit benennen, die Forderungen des Unterauftrages einschließlich der Qualitätsanforderungen zu erfüllen. Dabei sollte auch das bereits nachgewiesene Leistungsvermögen der Unterauftragnehmer berücksichtigt werden.

3. Der PSQP des Unterauftragnehmers sollte sich auf den PSQP des Hauptauftragnehmers beziehen. Diese Verknüpfung der Pläne ist notwendig für das Konfigurationsmanagement und besonders zur Koordinierung von Änderungen an den Konfigurationseinheiten.

4. Weitere Leitlinien können ISO/IEC/IEEE 90003:2018 (Abschnitt 8.4.1.1 und Abschnitt 8.4.1.2) entnommen werden.

5.7.3 Software-Konfigurationsmanagement (SCM)

1. Bei der Softwareentwicklung und/oder -pflege besteht eine enge Beziehung zwischen Software-Konfigurationsmanagement (SCM) und Software-Qualitätssicherung. Ohne einen systematischen SCM-Prozess fehlt eines der Werkzeuge für die Qualitätssicherung.

2. Das Konfigurationsmanagement ist eine Vorgehensweise zur Identifizierung, Kontrolle, Verfolgung und Überprüfung der Versionen jeder Software-Konfigurationseinheit (SCI). SCM sollte hinsichtlich Organisation, Methoden, Werkzeugen und Verfahren in einer wirtschaftlichen Weise angewandt werden und dennoch die notwendige Integrität und Rückverfolgbarkeit des Softwareprodukts sicherstellen.

3. Vorübergehende Änderungen an gelieferter Software, manchmal als „Patches“ bezeichnet, sollten streng kontrolliert werden. Wo solche Änderungen in die Software eingebracht werden, sollten sie in Übereinstimmung mit festgelegten Verfahren durchgeführt werden. In jedem Fall sollte durch anschließende Maßnahmen die Gültigkeit der Änderung bestätigt und, falls zutreffend, die Änderung formell gemäß den normalen Konfigurationsmanagementverfahren eingeführt werden.
4. Nichtkonforme Software sollte eindeutig als solche gekennzeichnet und von vertragskonformer Software getrennt werden.
5. Sobald „vertragskonforme“ Software freigegeben und für die Nutzung verfügbar ist (z. B. für Testbereiche oder in der Softwarebibliothek), muss ihr Status klar angegeben und bekannt gegeben werden.
6. Wenn die Software zu nichtkonformer Software wird (z. B. nach einem nicht bestandenen Test oder nach einer bestätigten Fehlermeldung durch den Kunden), muss sie aussortiert und deutlich als nichtkonform gekennzeichnet werden; dabei sind auch entsprechende Maßnahmen zu treffen, um den Zugang zu dieser Software zu kontrollieren.
7. Weitere Leitlinien können ISO/IEC/IEEE 90003:2018 (Abschnitt 8.5.2) entnommen werden.

5.7.4 Fremdentwickelte/Off-The-Shelf-Software

1. Die Bewertung und Validierung der Fähigkeit der Off-the-Shelf-Software, die geforderten Funktionen auszuführen, sollte solche Überlegungen wie die Rechte an geistigem Eigentum, Lizenzvereinbarungen und Konfigurationsmanagementkontrollen einschließen.
2. Off-the-Shelf-Software sollte dem Konfigurationsmanagement unterstellt werden, da sie die Integrität der entwickelten Software beeinflusst, und zwar unabhängig davon, ob es sich um eine Komponente der in der Entwicklung befindlichen Software handelt oder um ein Werkzeug, das bei der Entwicklung dieser Software hinzugezogen wird.
3. Der Auftragnehmer sollte in der Lage sein, objektiv den Nachweis (Validierungsberichte, Konfigurationsberichte usw.) zu erbringen, dass die Verwendung von Off-the-Shelf-Software bewertet worden ist und der Kontrolle unterliegt.
4. Dokumentationsanforderungen für Off-the-Shelf-Software sollten Funktions- und Schnittstellenspezifikationen einschließen.
5. Off-the-Shelf-Software schließt „auftraggeberseitig beigestellte Software“ ein. Bei auftraggeberseitig beigestellter Software wird die Entwicklungsfreiheit und -verantwortung des Auftragnehmers eingeschränkt.
6. Weitere Leitlinien können ISO/IEC/IEEE 90003:2018 (Abschnitt 8.4) entnommen werden.

5.7.5 Infrastruktur

7. Der Auftragnehmer sollte die Speicherung, die Sicherheit der Speicherung, die Umgebung, den Zugriff auf den Speicherort und die Freigabe aus dem Speicherort in einem Verfahren beschreiben, das auch erläutert, wie diese Tätigkeiten kontrolliert werden.
8. Datenträger, auf denen Software gespeichert ist, sollten so behandelt werden, dass die Integrität und Vertraulichkeit der gespeicherten Informationen gesichert ist.

9. Es ist deshalb notwendig, dass Aktivitäten, die aller Wahrscheinlichkeit nach Einfluss auf die Qualität haben, erkannt werden und Maßnahmen ergriffen werden, um eine Beeinträchtigung des Materials oder der Informationen zu vermeiden.

10. Ein angemessener Schutz durch Antivirusprogramme und Firewall sollte bereitgestellt werden.

11. Weitere Leitlinien können ISO/IEC/IEEE 90003:2018 (Abschnitt 7.1.3) entnommen werden.

5.7.6 Managementbewertung

Keine Leitlinien erforderlich.

5.8 Bewertung, Verifizierung und Validierung (EVV)

5.8.1 Allgemeines

1. Obwohl der EVV-Prozess ein Bestandteil des Managementprozesses und des technischen Prozesses (Abschnitte 5.5, 5.6, 5.7 und 5.9) ist, wird er wegen seiner Bedeutung im Qualitätsmanagement in diesem Abschnitt getrennt behandelt.

2. Aufgrund der Wechselbeziehung zwischen Bewertung, Verifizierung und Validierung sollten diese Tätigkeiten als Ganzes geplant werden. Die Zuordnung der Mittel und der Zeit sowie die Auswahl der Methoden und Techniken sollten so erfolgen, dass der gesamte EVV-Prozess optimiert wird.

3. Die richtige Durchführung der EVV-Aufgaben hat einen erheblichen Einfluss auf die Qualität des Endprodukts. Dieser Prozess erfordert im Allgemeinen den Einsatz beträchtlicher Ressourcen, so dass er in Bezug auf Verfügbarkeit qualifizierten Personals, Zeitplan, Kosten und Testumgebung sorgfältig geplant werden sollte.

4. Der Aufwand der EVV sollte an die Komplexität und/oder der Kritikalität der Software und die Forderungen des Vertrages angepasst werden, wobei die bestehenden Verfahren und Normen optimal genutzt werden sollten.

5. Verifikationskriterien für Softwareeinheiten sollten, in Übereinstimmung mit dem Softwareentwurf, eine Software-Integrationsstrategie mit folgenden Merkmalen ermöglichen:

- Es werden Softwareeinheiten entwickelt, die eine Übereinstimmung mit den ihnen zugeordneten Softwareanforderungen sicherstellen;
- Softwareeinheiten werden anhand der definierten Kriterien verifiziert;
- es werden Softwareeinheiten produziert, die durch die Integrationsstrategie definiert werden;
- die Ergebnisse der Integrationsprüfung werden festgehalten;
- zwischen dem Softwareentwurf und den Softwareeinheiten wird eine Übereinstimmung und Rückverfolgbarkeit hergestellt;
- es wird eine Regressionsstrategie entwickelt und angewandt, um Softwareeinheiten erneut zu verifizieren, wenn Änderungen an Softwareeinheiten (einschließlich dazugehöriger Anforderungen, Entwurf und Kodierung) auftreten.

6. Dieser Absatz bezieht sich nebenbei auch auf die Bewertung und Verbesserung der Software-Aspekte des QMS, d. h. die Anwendung der festgelegten Verfahren wird überwacht und die Richtigkeit und Effizienz dieser Verfahren beurteilt. Dieser Bewertungsprozess sollte

auf den von Projektgruppen zur Verfügung gestellten Daten beruhen und ist eine vertragsunabhängige Tätigkeit.

7. Weitere Leitlinien können ISO/IEC/IEEE 90003:2018 (Abschnitt 8.3.4.1, Abschnitt 8.3.4.2 und Abschnitt 8.3.4.3) entnommen werden.

5.8.2 Test-Durchführung

1. Im Allgemeinen sind Tests umso wirksamer, je früher sie im Rahmen des Softwareentwicklungsprozesses angesprochen/durchgeführt werden.

2. Die Planung und Spezifizierung von Tests sollte deshalb so früh wie möglich stattfinden. Während der Testplanung sollte, falls vertraglich gefordert, die Beteiligung von Auftraggeberpersonal an Testaktivitäten berücksichtigt werden.

3. Weitere Leitlinien können ISO/IEC/IEEE 90003:2018 (Abschnitt 8.3.4.4) entnommen werden.

5.8.3 Prüfungen (Reviews)

1. Auf die Software bezogene Review-Aktivitäten können unter verschiedenen Bezeichnungen geführt werden, dazu gehören Design-Reviews, Peer-Reviews, Walk-throughs, Inspektion, Dokument-Reviews, Schreibtischtests usw.

2. Die Erfahrung hat gezeigt, dass bedeutende Softwarefehler in den frühen Phasen des Softwareentwicklungsprozesses entstehen. Man sollte deshalb den Schwerpunkt auf Design-Reviews in diesen Phasen legen, um eine frühe Erkennung und Behebung von Fehlern zu fördern.

3. Weitere Leitlinien können ISO/IEC/IEEE 90003:2018 (Abschnitt 8.3.4.1) entnommen werden.

5.9 Wartung/Pflege

1. Bei der Softwarepflege sollte der Auftragnehmer außerdem Folgendes berücksichtigen:

- Art, Nutzung und beabsichtigte Lebensdauer des Softwareprodukts
- Forderungen und Rückmeldungen des Auftraggebers
- Erfassung/Analyse von Nutzungsdaten (Leistung, Zuverlässigkeit, Erfahrungen usw.)
- Kontrolle, Aktualisierung und Bereitstellung technischer Dokumentation im Zusammenhang mit der Nutzung des Softwareprodukts
- Kontrollen, die für außerhalb der Organisation getätigte Arbeiten erforderlich sind
- Produkt-/Kundenunterstützung (Anfragen, Schulung, Gewährleistungen, Obsoleszenz usw.).

2. Werden nach der Lieferung Probleme festgestellt, sollte der Auftragnehmer geeignete Maßnahmen einschließlich Untersuchung und Meldung ergreifen.

3. Zur Erleichterung der Softwareproduktwartung ist die Verfügbarkeit längerfristiger Unterstützung für Softwarewerkzeuge ein wichtiger Aspekt, der nicht unberücksichtigt bleiben sollte.

4. Weitere Leitlinien können ISO/IEC/IEEE 90003:2018 (Abschnitt 8.5.1.7) entnommen werden.

ANHANG A

Muster für die Begründung von Anpassungen im Rahmen der AQAP-2210

Begründung: Der Grund, warum der Absatz/Unterabsatz nicht auf den Vertrag anwendbar ist und gestrichen werden kann.

Risiko: Das/die Risikoereignis(se) und die Folgen für die Erreichung von Zielen, die aufgrund der Streichung des Absatzes/Unterabsatzes auftreten können.

Minderung der Auswirkungen: Eine Aktivität oder Maßnahme (falls erforderlich), von der erwartet wird, dass sie die Auswirkungen des Risikoereignisses im Falle seines Auftretens verringert.

Risikominderung: Eine Aktivität oder Maßnahme (falls erforderlich), von der erwartet wird, dass sie die Wahrscheinlichkeit des Eintretens des Risikoereignisses verringert.

Anpassung von Absätzen der AQAP-2210 Ausg. B, 1. Fassung

Absatz	Bezeichnung	Unterabsatz	Gestrichen	Begründung	Risiko	Minderung der Auswirkungen	Risikominderung
5.1	Allgemeines	Alle		Kann nicht gestrichen werden			
5.2	Software-Qualitätssicherungsplan des Projekts (PSQP)	Alle		Kann nicht gestrichen werden			
5.3	Untersuchung der Softwarekritikalität	Alle		Kann nicht gestrichen werden			
5.4	Softwarequalitätsmodell	1,3		Kann nicht gestrichen werden			
	<i>Softwarequalitätsmodell</i>	2					
5.5	Identifizierung und Prüfung der Softwareanforderungen	1, 2, 3, 5, 6, 7		Kann nicht gestrichen werden			
	<i>Identifizierung und Prüfung der Softwareanforderungen</i>	4					

Absatz	Bezeichnung	Unterabsatz	Gestrichen	Begründung	Risiko	Minderung der Auswirkungen	Risikominderung
5.6	Softwareentwurfs- und -entwicklungsprozess	1					
		2					
		3					
		4					
		5					
		6					
		7					
5.7	MANAGEMENT						
5.7.1	Rollen, Verantwortlichkeiten und Befugnisse in der Organisation	Alle	Kann nicht gestrichen werden				
5.7.2	Unterauftragnehmermanagement	Alle	Kann nicht gestrichen werden				
5.7.3	Software-Konfigurationsmanagement (SCM)	Alle	Kann nicht gestrichen werden				
5.7.4	Fremdentwickelte/Off-The-Shelf-Software	1					
		2					
		3					
		4					
5.7.5	Infrastruktur	1	Kann nicht gestrichen werden				
5.7.6	Managementbewertung						
5.7.6.1	Allgemeines	1					
5.8	BEWERTUNG, VERIFIZIERUNG UND VALIDIERUNG (EVV)						

Absatz	Bezeichnung	Unterabsatz	Gestrichen	Begründung	Risiko	Minderung der Auswirkungen	Risiko-minderung
5.8.1	Allgemeines	1					
		2					
		3					
		4					
		5					
		6					
5.8.2	Test-Durchführung	1					
		2					
		3					
		4					
5.8.3	Prüfungen (Reviews)	1					
		2					
5.9	WARTUNG/PFLEGE	1					
		2					
		3					

Anpassung von Absätzen aus AS9115-

Absatz	Bezeichnung	Gestrichen	Begründung	Risiko	Minderung der Auswirkungen	Risiko-minderung
4.3	Festlegung des Umfangs des Qualitätsmanagementsystems					
4.4	Das Qualitätsmanagementsystem und seine Prozesse					
6.1	Maßnahmen zur Bewältigung von Risiken und Chancen					

Absatz	Bezeichnung	Gestrichen	Begründung	Risiko	Minderung der Auswirkungen	Risiko-minderung
7.1.3	Infrastruktur					
7.1.4	Umgebung für den Betrieb von Prozessen					
7.1.5.1	Allgemeines					
7.2	Kompetenz					
7.3	Bewusstsein (Awareness)					
7.5.3	Kontrolle dokumentierter Informationen					
8.1	Betriebliche Planung und Kontrolle					
8.1.1	Operatives Risikomanagement					
8.1.1.1	Produktintegrität					
8.1.2	Konfigurationsmanagement					
8.1.3	Produktsicherheit					
8.1.4	Vermeidung von gefälschten Teilen					
8.2.2	Festlegung der Anforderungen an Produkte und Dienstleistungen					
8.2.3	Prüfung der Anforderungen an Produkte und Dienstleistungen					

Absatz	Bezeichnung	Gestrichen	Begründung	Risiko	Minderung der Auswirkungen	Risiko-minderung
8.3.2	Entwurfs- und Entwicklungsplanung					
8.3.3	Entwurfs- und Entwicklungseingaben (Inputs)					
8.3.4	Entwurfs- und Entwicklungskontrollen					
8.3.5	Entwurfs- und Entwicklungsergebnisse (Outputs)					
8.3.6	Entwurfs- und Entwicklungsänderungen					
8.4.1	Allgemeines					
8.4.2	Art und Umfang der Kontrolle					
8.4.3	Informationen für externe Anbieter					
8.5.1	Kontrolle der Produktion und der Dienstleistungserbringung					
8.5.1.1	Kontrolle von Gerät, Werkzeugen und Softwareprogrammen					
8.5.1.3	Verifizierung von Produktionsprozessen					
8.5.2	Kennzeichnung und Rückverfolgbarkeit					
8.5.4	Erhaltung					
8.5.6	Änderungssteuerung					

Absatz	Bezeichnung	Gestrichen	Begründung	Risiko	Minderung der Auswirkungen	Risiko-minderung
8.6	Freigabe von Produkten und Dienstleistungen					
8.7	Kontrolle nichtkonformer Ergebnisse					
9.1.3	Analyse und Bewertung					
9.2	Internes Audit					
9.3.2	Eingaben (Inputs) für die Managementbewertung					

AQAP-2210-SRD.1(B)(1)