VOLKSWAGEN

AKTIENGESELLSCHAFT

Konzern Grundanforderungen Software

Grundanforderungen, die der Volkswagen-Konzern an im Fahrzeug verbaute und fahrzeugbezogene Software/softwarebestimmt Systeme und deren Entwicklungsprozesse stellt.

Technische Entwicklung, Querschnittslastenheft: LAH.893.909

Erstausgabe	06.09.2002
Änderungsstand	04.04.2024
Lastenheftversion	4.3

Inhaltsverzeichnis

1 Vorwort ... 4
1.1 Zielsetzung ... 4
1.2 Besitzer dieses Dokumentes ... 4
1.3 Cybersecurity Incident Management ... 5
2 Geltungsbereich ... 7
3 Rechte des Auftraggebers und Pflichten des Auftragnehmers ... 8
4 Terminologie ... 9
4.1 Feature ... 9
4.2 System ... 9
4.3 Systemelement ... 9
4.4 Software ... 9
4.5 Softwareelement ... 9
4.6 Software Komponente ... 9
4.7 Software Unit ... 10
4.8 Modellelement ... 10
4.9 Weitere Definitionen ... 10
5 System- und Softwareentwicklung ... 11
5.1 Prozessübergreifende Anforderungen ... 11
5.2 Projektmanagement ... 12
5.2.1 Projektmetriken ... 13
5.3 Dokumentation des Lieferumfangs ... 13
5.4 System- und Softwareanforderungsspezifikation ... 14
5.5 System- und Softwarearchitekturspezifikation ... 15
5.6 Softwarefeinspezifikation (Detailed Design) ... 15
5.7 Softwareerstellung ... 16
5.7.1 Programmiersprachen ... 16
5.7.2 Manuelle Quellcodeerstellung ... 16
5.7.2.1 Quellcodemetriken ... 17
5.7.3 Grafische und modellbasierte Programmierung ... 17
5.7.3.1 Metriken für Grafische Programmierung ... 18
5.7.4 Maschinelles Lernen (ML) ... 18
5.7.5 Tool-Qualifizierung ... 19
5.8 Test ... 19
5.8.1 Testplanung ... 19
5.8.2 Testfallspezifikation ... 19
5.8.3 Testdurchführung allgemein ... 20
5.8.4 Software Unittest ... 20
5.8.5 Softwareintegrationstest ... 20
5.8.6 Softwareverifikation ... 21
5.8.7 Systemintegrationsverification ... 21
5.8.8 Systemverifikation ... 21
5.9 Qualitätssicherung und -management ... 21
5.9.1 Qualitätsmanagement ... 21
5.9.2 Review der Arbeitsprodukte ... 22
5.9.3 Prüfung der Entwicklungsprozesse ... 22
5.10 Konfigurationsmanagement ... 22
5.11 Problemlösungsmanagement ... 22
5.12 Software von Dritten ... 23
5.13 Free and Open Source Software ... 24
5.14 Cybersecurity-relevante Entwicklung ... 26

5.14.1 Allgemeine Cybersecurity-Anforderungen 26
5.14.2 Cybersecurity-Terminologie 27
5.14.3 Cybersecurity-Management 29
5.14.4 Cybersecurity-Risikoanalyse 29
5.14.5 Cybersecurity-Risikomanagement 30
5.14.6 Cybersecurity-Architektur und Cybersecurity-Design 30
5.14.7 Cybersecurity-Implementierung 31
5.14.8 Cybersecurity-Nachweis 31
5.15 Cybersecurity-Aktivitäten nach der Entwicklungsphase (Serien-/Feldbetreuung) 32
6 Referenzierte Unterlagen 35
6.1 Dokumente der Volkswagen AG 35
6.2 Dokumente des Verbands der Automobilindustrie (VDA) 36
6.3 Internationale Standards und Normen 36
7 Release Notes 37
8 Vertraulichkeitshinweis 38

1 Vorwort

1.1 Zielsetzung

[I: KGAS_4110]

Die KGAS definiert den von der Volkswagen AG vorgegebenen Rahmen, um die Entwicklung einer Software, für den Einsatz im Fahrzeug und im Fahrzeugumfeld, nach Stand der Technik sicher zu stellen.

[I: KGAS_4111]

Als Grundlage für die Softwareentwicklung nach Stand der Technik dient das Prozessmodell Automotive SPICE® (siehe KGAS_3887).

[I: KGAS_4112]

Die KGAS präzisiert und ergänzt anhand von Prozessanforderungen das Prozessmodell Automotive SPICE® (siehe KGAS_3887).

[I: KGAS_4113]

Die KGAS richtet sich an alle Projektbeteiligten eines Softwareentwicklungsprojektes.

[I: KGAS_4114]

Die KGAS schafft die Grundlage, Softwarequalität im Produkt durch stabile Prozesse zu erreichen (siehe KGAS_3043).

[I: KGAS_3090]

Die Methoden, Bewertungsgrundlagen und Konsequenzen der Qualitätssicherungsaktivitäten des Volkswagen Konzerns bei Lieferanten sind in der "Formel Q Fähigkeit Software" (KGAS_2834) beschrieben.

[I: KGAS_3633]

Die in diesem Dokument verwendeten Begriffe "Auftraggeber" und "Auftragnehmer" sind gleichbedeutend mit den in der Formel Q Fähigkeit Software (KGAS_2834) verwendeten Begriffen "Kunde" und "Lieferant".

1.2 Besitzer dieses Dokumentes

[I: KGAS_2085]

Volkswagen AG
Konzern Qualität
Qualität Software
Brieffach 81250
38436 Wolfsburg
Deutschland

Kontakt: software.qualitaet.vwag.r.wob@volkswagen.de

Audi AG
Qualitätssteuerung Software
85045 Ingolstadt
Deutschland

Kontakt: software-quality.qq@audi.de

Porsche AG
Qualität Software
Porscheplatz 1
70435 Stuttgart
Deutschland

Kontakt: software.quality@porsche.de

CARIAD SE
Berliner Ring 2,
Brieffach 1080/2
38440 Wolfsburg
Deutschland

Kontakt: quality@cariad.technology

1.3 Cybersecurity Incident Management

[R: KGAS_3890]

Bei Sicherheitsvorkommnissen ist das Incident Team der Marke oder Region zu benachrichtigen, die die Entwicklungsverantwortung für das betroffene Produkt im Volkswagen Konzern hat. Im Fall einer unklaren Verantwortung ist das Incident Team von Volkswagen PKW (KGAS_3876) zu benachrichtigen.

[I: KGAS_3876]

Volkswagen PKW und Volkswagen Nutzfahrzeuge (leichte Nutzfahrzeuge)

Kontakt: CSI-wob@volkswagen.de

[I: KGAS_3891]

Porsche AG

Kontakt: csi@porsche.de

[I: KGAS_3892]

MAN SE

Kontakt: carsecurity@man.eu

[I: KGAS_3926]

Audi AG

Kontakt: vulnerability@audi.de

[I: KGAS_3958]

Kontakt für die in der Entwicklung befindlichen Produkte: Ansprechpartner ist der jeweilige Projektverantwortliche aus der jeweiligen technischen Entwicklung.

[I: KGAS_3993]

Škoda Auto a.s.

Kontakt: teamcsi@skoda-auto.cz

[I: KGAS_4017]

SEAT S.A.

Kontakt: csi@seat.es

[I: KGAS_4117]

Bentley

Kontakt: csi@bentley.co.uk

[I: KGAS_4118]

Lamborghini

Kontakt: csi@lamborghini.com

[I: KGAS_4119]

Cariad SE

Kontakt für Cybersecurity ausnutzbare Schwachstellen (Vulnerabilities): vuln@cariad.technology

Kontakt für Cybersecurity-Vorfälle (Incidents): sirt@cariad.technology

[I: KGAS_3959]

Region China

Kontakt: csi-cn@volkswagen.com.cn

2 Geltungsbereich

[A: KGAS_4120]

Die KGAS gilt für Entwicklungsprozesse und die dazugehörigen entwicklungsbegleitenden Prozesse für Software und softwarebestimmte Systeme (z.B. Steuergeräte mit Software), die zur Realisierung einer Funktion im Fahrzeug beitragen.

[R: KGAS_3028]

Die Anforderungen in diesem Dokument gelten für den gesamten Volkswagen Konzern und seine Auftragnehmer.

[A: KGAS_4121]

Die KGAS ist ab Beginn der Entwicklung von Software und softwarebestimmten Systemen (z.B. Steuergeräte mit Software) umzusetzen.

3 Rechte des Auftraggebers und Pflichten des Auftragnehmers

[R: KGAS_4058]

Anforderungen sind nachweislich einzuhalten.

[R: KGAS_4059]

Rahmenbedingungen sind einzuhalten.

[I: KGAS_4060]

Informationen [I] dienen dem zusätzlichen Verständnis oder als Hinweis zu einer möglichen Umsetzung der Anforderung.

[A: KGAS_3885]

Alle Entwicklungsartefakte und Dokumente müssen in englischer oder deutscher Sprache verfasst werden.

[A: KGAS_1806]

Der Auftragnehmer muss auf Anfrage des Auftraggebers Nachweise über die Einhaltung der KGAS erbringen.

[A: KGAS_27]

Der Auftragnehmer muss alle von ihm eingesetzten Unterauftragnehmer auf die Erfüllung der KGAS verpflichten und deren Umsetzung sicherstellen.

[A: KGAS_2933]

Können der Auftragnehmer oder von ihm eingesetzte Unterauftragnehmer die KGAS nicht vollständig erfüllen, muss der Auftragnehmer die Abweichungen schriftlich vor Projektstart durch die Qualitätssicherung des Auftraggebers genehmigen lassen. Die vereinbarten und genehmigten Änderungen sind an die Konzern Qualität (Kontakt siehe KGAS_2085) zu senden.

[R: KGAS_51]

Der Auftragnehmer muss dem Auftraggeber die Möglichkeit einräumen, die Einhaltung der KGAS durch Quellcodeanalysen, werkzeuggestützte Analysen und andere geeignete Verfahren zu überprüfen.

[R: KGAS_4000]

Der Auftraggeber kann seine Analysen (KGAS_51), die sich aus der KGAS ergeben, vollständig oder teilweise durch Dritte wahrnehmen lassen.

[R: KGAS_2949]

Der Auftragnehmer muss die Analyse (KGAS_51) durch die Bereitstellung des Quellcodes in der entsprechenden Steuergerätekonfiguration in Räumen und in Anwesenheit des Auftragnehmers unterstützen.

[R: KGAS_3546]

Steht eine Anforderung der KGAS in Widerspruch zu einer Anforderung aus einer mitgeltenden Unterlage, so muss der Auftragnehmer eine spezifische Vereinbarung zwischen dem Auftragnehmer und dem Auftraggeber initiieren.

4 Terminologie

[I: KGAS_1984]

In diesem Kapitel wird definiert, wie relevante Fachbegriffe im Rahmen der KGAS zu interpretieren sind.

4.1 Feature

[I: KGAS_3665]

Ein Feature ist ein durch einen Stakeholder definierter Funktionsumfang, der durch eine Teilmenge der Anforderungen abgebildet wird.

4.2 System

[I: KGAS_2877]

Das System ist der gesamte vom Auftragnehmer zu erbringende Lieferumfang.

[I: KGAS_2879]

Das System besteht aus Systemelementen.

4.3 Systemelement

[I: KGAS_3604]

Die Definition Systemelement erfolgt gemäß ASPICE PAM 4.0 .

4.4 Software

[I: KGAS_2876]

Software ist die gesamte im Lieferumfang enthaltene Software.

[I: KGAS_3523]

Software besteht aus einem oder mehreren Softwareelementen.

[I: KGAS_2878]

Typische Bestandteile von Software sind Applikation, Treiber, Hardware-Abstraktionen, Betriebssystem, implementierte Algorithmen.

[I: KGAS_2880]

Zur Software zählen auch Plattformelemente, Software von Dritten und programmierte Schaltungen.

4.5 Softwareelement

[I: KGAS_3095]

Die Definition Softwareelement erfolgt gemäß ASPICE PAM 4.0 .

4.6 Software Komponente

[I: KGAS_3651]

Die Definition Software Komponente erfolgt gemäß ASPICE PAM 4.0 .

4.7 Software Unit

[I: KGAS_2998]

Die Definition Software Unit erfolgt gemäß ASPICE PAM 4.0 .

4.8 Modellelement

[I: KGAS_3527]

Ein Modellelement ist die logische Repräsentation eines oder mehrerer Basisobjekte in einem Tool zur modellbasierten Codegenerierung.

[I: KGAS_3528]

Ein Basisobjekt ist ein atomares Objekt innerhalb eines Tools zur modellbasierten Codegenerierung, das nicht weiter in Unterobjekte teilbar ist.

4.9 Weitere Definitionen

[I: KGAS_3820]

Free and Open Source Software (FOSS) ist jegliche Software, die unter Nutzungs- und Lizenzbestimmungen vertrieben wird, zu deren wesentlicher Verpflichtung typischerweise die Weitergabe oder Offenlegung des Quellcodes der Software bei deren Verbreitung gehört.

[I: KGAS_3953]

Gelieferte Software, ist die vom Auftragnehmer gelieferte Software (u.a. Auftragnehmer Software, Software von Dritten).

[I: KGAS_4066]

Die Inhalte des Lieferumfangs sind im Rahmen einer Beauftragung festgelegt.

[I: KGAS_3954]

Software von Dritten ist Fremdsoftware, welche nicht Auftragnehmer-Software ist.

[I: KGAS_4067]

Whitebox-Test ist ein Testverfahren, das auf der inneren Struktur einer Komponente oder eines Systems basiert.

[I: KGAS_4068]

Blackbox-Test ist ein Testverfahren, das auf einer Analyse der Spezifikation einer Komponente oder eines Systems basiert, gemäß ISO/IEC/IEEE 29119 (KGAS_3479).

[I: KGAS_3956]

Toter Code ist in der Auslieferung enthaltener Code, der durch den von der Spezifikation vorgegebenen Programmablauf (inkl. Fehlerhandling) nicht ausgeführt werden kann.

[I: KGAS_3962]

Toter Code ist gegeben, wenn der Code nicht ausgeführt wird, weil dieser
- nicht mehr benötigt wird,
- nicht aufgerufen wird,
- nicht aufgerufen werden kann.

[I: KGAS_3964]

Toter Code ist nicht gegeben, wenn
- eine Anforderung geplant, umgesetzt, aber durch den Auftraggeber nicht eingesetzt wird,
- der Code durch Parametrierung (z.B. Zieldatencontainer) nicht aufgerufen wird.

5 System- und Softwareentwicklung

[I: KGAS_3124]

Dieses Kapitel beinhaltet Anforderungen an die Organisation, die Entwicklungsprozesse, die Arbeitsprodukte und die Infrastruktur des Auftragnehmers.

5.1 Prozessübergreifende Anforderungen

[A: KGAS_2074]

Das gesamte im Lieferumfang enthaltene softwarebestimmte System oder die Software muss mit Prozessen entwickelt sein, die mindestens einen Reifegrad **„Level 2“** in einem Automotive SPICE® Assessment gemäß Formel-Q Fähigkeit Software erreichen.

[A: KGAS_4122]

Jedes an den Auftraggeber geliefertes Release muss in Bezug auf die mit dem Kunden für dieses Release vereinbarten Anforderungen vollständig gemäß KGAS entwickelt, implementiert und verifiziert sein.

[A: KGAS_4123]

Der Auftragnehmer muss auch für bereits entwickelte Software nachweisen, dass die Softwareentwicklungsprozesse, mit denen die Software entwickelt wurde, dem aktuellen Stand der Technik entsprechen.

[A: KGAS_4145]

Der Nachweis von KGAS_4123 muss über den ASPICE Prozess REU.2 oder vergleichbar erfolgen.

[I: KGAS_4146]

Unter KGAS_4123 wird auch Legacy, Plattform und Beifang (siehe KGAS_4147) betrachtet.

[I: KGAS_4124]

Bereits entwickelte Software beinhaltet z.B. Softwareanteile, die bereits vor Nominierung entwickelt oder eingekauft wurden.

[A: KGAS_3896]

Es muss sichergestellt sein, dass kein unbenutzter Code (z.B. unzugänglicher oder toter Code) vorhanden ist.

[A: KGAS_3679]

Der Auftragnehmer muss bis zu 15 Jahre nach End of Production (EoP) in der Lage sein, alle vom Auftraggeber für erforderlich erachteten Fehlerkorrekturen in der Software umzusetzen. Der Auftragnehmer muss sicherstellen, dass die gelieferte Software vorgehalten wird und alle notwendigen Voraussetzungen zur Bearbeitung und Lieferung der Software unter Beachtung der Anforderungen der KGAS gegeben sind.

[A: KGAS_2035]

Alle vom Prozess vorgeschriebenen Arbeitsprodukte müssen zum Zeitpunkt eines Releases an den Auftraggeber inhaltlich konsistent zueinander sein.

[A: KGAS_3552]

Zur Verfeinerung von Spezifikationselementen (z.B. Anforderungen, Architekturelemente) von einer Abstraktionsebene bzw. Hierarchieebene auf die darunter liegende Abstraktionsebene muss der Auftragnehmer Kriterien definieren und deren Einhaltung sicherstellen.

[A: KGAS_4125]

Wenn die Kriterien aus KGAS_3552 in Einzelfällen nicht eingehalten werden, muss die Abweichung nachweislich begründet werden.

[I: KGAS_4126]

Ein gängiges Kriterium der KGAS_3552 ist ein Verhältnis einer Abstraktionsebene auf die darunter liegende Abstraktionsebene von 1 zu 10.

[R: KGAS_3968]

Um einen datenschutzkonformen Einsatz der einzelnen Steuergeräte im Feld gewährleisten zu können, sind die datenschutzrechtlichen Anforderungen bereits ab dem Entwicklungsbeginn zu berücksichtigen. Die "Richtlinie für die datenschutzrechtlichen Anforderungen bei der (Weiter-)Entwicklung von Steuergeräten mit Speicherfunktion" ist einzuhalten (KGAS_3966).

5.2 Projektmanagement

[A: KGAS_4127]

Das Projektmanagement muss Termin- und Feature-Treue für jedes Release sicherstellen.

[I: KGAS_3595]

Aufwandsschätzungen für alle Arbeitspakete sind durchgeführt und nachvollziehbar.

[I: KGAS_3146]

Für alle Arbeitspakete sind vorhandene Abhängigkeiten zu anderen Arbeitspaketen ersichtlich.

[A: KGAS_3167]

Für Änderungs- und Problemlösungsumfänge müssen angemessene Pauschalaufwände eingeplant sein.

[I: KGAS_3154]

Es ist eine Feature-Release-Planung zu erstellen, die eine Aufteilung der Features auf die Meilensteine des Auftraggebers beinhaltet.

[A: KGAS_3594]

Wenn ein Feature über mehrere Releases umgesetzt wird, so muss dieses Feature in der Feature-Release-Planung weiter verfeinert werden, so dass pro Release ein exakt zu prüfender Umfang umgesetzt werden kann.

[A: KGAS_3157]

Die in der Feature-Release-Planung enthaltenen Features müssen den Anforderungen aus der System-und Softwareanforderungsspezifikation zugeordnet werden.

[I: KGAS_3177]

Der Terminplan beinhaltet alle Aktivitäten, die sich aus Einträgen des Problemlösungsmanagements und des Änderungsmanagements ergeben.

[I: KGAS_3171]

Der kritische Pfad des Terminplans muss systematisch identifizierbar sein.

[I: KGAS_3178]

Eindeutige Definitionen für Erfüllungsgrade von Arbeitspaketen und Aktivitäten existieren und werden angewendet.

[I: KGAS_3191]

Projektrisiken sind nachweislich identifiziert, bewertet und mit entgegenwirkenden Maßnahmen versehen.

[A: KGAS_3727]

Der Status, Fortschritt und offene Punkte aller Aktivitäten müssen zu jeder Zeit für Auftraggeber und Auftragnehmer transparent sein.

5.2.1 Projektmetriken

[A: KGAS_3612]

Zur Projektsteuerung muss der Auftragnehmer von Projektbeginn an Metriken erheben.

[A: KGAS_3915]

Der Auftragnehmer muss zu jedem Release und auf Anfrage die erhobenen Metriken dem Auftraggeber zur Verfügung stellen, mindestens aber alle 4 Wochen.

[A: KGAS_4107]

Der Mindestsatz der Projektmetriken ist definiert in KGAS_4093 sofern vom Auftraggeber nicht anders vorgegeben.

[A: KGAS_4129]

Das Austauschformat wird durch den Auftraggeber festgelegt.

[A: KGAS_3624]

Bei festgestellten Abweichungen, die durch den Einsatz der Metriken (KGAS_3612) erkennbar werden, müssen Verbesserungsmaßnahmen mit Zielterminen festgelegt werden.

5.3 Dokumentation des Lieferumfangs

[A: KGAS_4115]

Die Dokumentation erfolgt in den ReleaseNotes KGAS_4116, sofern zwischen Auftraggeber und Auftragnehmer nicht anders vereinbart.

[I: KGAS_3214]

Das Freigabelevel des Lieferumfangs (z.B. Entwicklungsstand ohne Straßennutzung, Entwicklungsstand mit Straßennutzung oder Serienfreigabe) ist zu dokumentieren.

[I: KGAS_3215]

Die umgesetzten Änderungen des Lieferumfangs sind zu dokumentieren, inklusive Auflistung durchgeführter Fehlerbehebungen.

[I: KGAS_3938]

Die Releasenotes und Feature-Übersichten aller Umfänge (z.B. Module) der Unterauftragnehmer sind zu dokumentieren.

[I: KGAS_3216]

Die für den Lieferumfang auszuführenden Tests und deren Testergebnisse sind zu dokumentieren.

[I: KGAS_3219]

Jede mit der Softwareversion des Lieferumfangs kompatible Hardwareversion ist zu dokumentieren.

[I: KGAS_3888]

Die Buildumgebung, Buildkonfiguration, Definitionen, Compileroptionen und -optimierungen inkl. Änderungshistorie sind zu dokumentieren.

5.4 System- und Softwareanforderungsspezifikation

[A: KGAS_4130]

Alle Anforderungen müssen zu Ihrer Quelle zurück verfolgbar sein.

[A: KGAS_3794]

Alle Anforderungen, die keinen nachweislichen Bezug zu den Anforderungen des Auftraggebers haben, müssen durch den Auftragnehmer angezeigt werden.

[A: KGAS_3406]

Alle getroffenen Annahmen müssen als Anforderungen spezifiziert und mit dem Auftraggeber abgestimmt werden.

[A: KGAS_3548]

Eigene Anforderungen des Auftragnehmers (z. B. Anforderungen zur Fertigung, Anforderungen aus Plattformanteilen, usw.) müssen in den System- und Softwareanforderungsspezifikationen dokumentiert sein.

[I: KGAS_3266]

Alle Anforderungen sind nachweislich unter Berücksichtigung mindestens folgender Aspekte zu erstellen und analysieren:

- Machbarkeit
- Verifizierbarkeit
- Widerspruchsfreiheit
- Verständlichkeit
- Eindeutigkeit
- Atomarität

[I: KGAS_3535]

Alle Anforderungen sind einem Release bzw. Feature zuzuordnen.

[A: KGAS_3257]

Alle Anforderungen müssen mindestens bezüglich Safety-, Gesetzes- und Cybersecurityrelevanz kategorisiert werden.

[A: KGAS_3263]

Für jede funktionale Anforderung müssen alle technisch möglichen Szenarien spezifiziert werden (z.B. Sollverhalten, Fehlerfall, Alternativpfad, Grenzfälle und Worst-Case Szenarien).

[I: KGAS_3262]

Anforderungen sind von einer höheren Anforderungsebene zu einer niedrigeren Anforderungsebene nicht zusammenzufassen, wenn dadurch Informationsverlust entsteht.

[I: KGAS_3264]

Jede nichtfunktionale Anforderung ist in daraus abgeleiteten Anforderungen und Arbeitsprodukten nachweislich zu berücksichtigen.

5.5 System- und Softwarearchitekturspezifikation

[A: KGAS_3278]

Jedes System- und Softwareelement muss eine textuelle Beschreibung mit mindestens Ziel und Zweck enthalten.

[A: KGAS_3275]

Für die Beschreibung der System- und Softwareelemente innerhalb der System- bzw. Softwarearchitekturspezifikationen müssen Syntax und Semantik definiert sein.

[I: KGAS_3279]

Gemeinsam genutzte Ressourcen (z. B. globale Variablen) sind als Schnittstellen anzusehen und entsprechend vollständig zu beschreiben.

[A: KGAS_3282]

Für alle Softwareelemente müssen die Anforderungen für den entsprechenden Ressourcenbedarf spezifiziert sein. Diese müssen mindestens den maximalen CPU-Zeitverbrauch, den maximalen flüchtigen Speicherverbrauch, den maximalen nicht-flüchtigen Speicherverbrauch beinhalten.

5.6 Softwarefeinspezifikation (Detailed Design)

[A: KGAS_3285]

Die Softwarefeinspezifikation muss für jede Komponente sowie jede darin enthaltene Unit eine textuelle Beschreibung mit Ziel, Zweck und internem Aufbau enthalten, um Nachvollziehbarkeit, Qualität, Transparenz und Wartbarkeit des daraus abgeleiteten und implementierten Codes zu gewährleisten.

[A: KGAS_3288]

Für die Beschreibung der Softwarefeinspezifikation müssen Syntax und Semantik definiert sein.

[A: KGAS_3289]

Alle zu implementierenden Units und Unitelemente müssen in der Softwarefeinspezifikation beschrieben werden.

[I: KGAS_4061]

In der Softwarefeinspezifikation ist der Lösungsansatz (KGAS_4062) für das nach außen wahrnehmbare Verhalten einer Unit zu beschreiben.

[I: KGAS_4062]

Der Lösungsansatz definiert Algorithmen, Berechnungen, Schnittstellen, Funktionsaufrufe und Makros und das Verhalten im Fehlerfall, soweit jeweils anwendbar.

[I: KGAS_4063]

Alle notwendigen Informationen zur Umsetzung eines Lösungsansatzes (KGAS_4062) sind zu beschreiben oder zu referenzieren.

[I: KGAS_3298]

Gemeinsam genutzte Ressourcen (z. B. Libraries, Parameter, globale und komponentenglobale Variablen) sind als Schnittstellen anzusehen und entsprechend vollständig zu beschreiben.

[A: KGAS_3455]

Die Softwarefeinspezifikation muss auch im Falle einer grafischen bzw. modellbasierten Programmierung erstellt werden.

[A: KGAS_3682]

Für alle Schnittstellen muss eine Gültigkeitsprüfung gegenüber der Schnittstellenbeschreibung spezifiziert sein.

[A: KGAS_3683]

Bei negativen Gültigkeitsprüfungen von Schnittstellen muss ein definiertes System- bzw. Softwareverhalten spezifiziert sein.

5.7 Softwareerstellung

5.7.1 Programmiersprachen

[A: KGAS_2050]

Als Programmiersprache des Softwareprodukts muss eine international standardisierte (z. B. ISO/IEC) Hochsprache verwendet werden.

[I: KGAS_2837]

Die Verwendung anderer Programmier- oder Scriptsprachen im Softwareprodukt ist nur nach Begründung, Eignungsnachweis und Genehmigung durch den Auftraggeber zulässig.

5.7.2 Manuelle Quellcodeerstellung

[R: KGAS_3948]

Dieses Kapitel gilt nur für Software (Lieferumfang), bei denen Methoden der handcodierten Programmierung zum Einsatz kommen.

[A: KGAS_3910]

Der Auftragnehmer muss für die gesamte Quellcodeerstellung nachweislich für das Projekt geeignete Codierrichtlinien anwenden. Es gelten auch die relevanten Richtlinien aus KGAS_3908.

[A: KGAS_3321]

Innerhalb des Quellcodes müssen definierte Namensregeln verwendet werden (z.B. für Funktionsnamen, Makros, Variablen, Typdefinition).

[A: KGAS_3878]

Alle Abweichungen von den angewandten Codierrichtlinien müssen begründet und dokumentiert werden.

[I: KGAS_3328]

Jede Unit ist mit mindestens einer Kurzbeschreibung der Unit, der Inputparameter und der Rückgabewerte zu kommentieren.

[I: KGAS_3325]

Der Quellcode ist an allen Entscheidungspunkten bezüglich der Bedeutung bzw. Logik zu kommentieren (z. B. bei if-else, for, switch, while).

[I: KGAS_3326]

Der Quellcode ist bei allen Berechnungen mit mehreren Variablen oder Parametern bezüglich der Bedeutung bzw. Logik zu kommentieren.

[A: KGAS_3324]

Jede Unit muss nachweislich durch ein Quellcodereview geprüft werden.

[I: KGAS_3562]

Ziele von Quellcodereviews (KGAS_3324) sind mindestens: Prüfung, ob der Quellcode der Softwarefeinspezifikation entspricht, Prüfung nicht funktionaler Anforderungen, Prüfung Einhaltung nicht automatisch prüfbarer Codierrichtlinien.

5.7.2.1 Quellcodemetriken

[I: KGAS_4099]

Die Quellcodemetriken dienen zur Messbarkeit der Qualität des Quellcodes.
Die Metriken sind Indikatoren gemäß der Qualitätskriterien der ISO 25010 (KGAS_3043).

[A: KGAS_4100]

Bei manueller Quellcodeerstellung muss der Auftragnehmer geeignete Quellcodemetriken nach Stand der Technik mit definierten Grenzwerten anwenden.

[A: KGAS_3570]

Die Auswahl und Eignung der Quellcodemetriken (KGAS_4100) muss begründet sein (z.B. in einem entsprechenden Strategiedokument).

[A: KGAS_4065]

Grenzwertverletzungen müssen nachvollziehbar dokumentiert werden.

[A: KGAS_4101]

Grenzwertverletzungen müssen auf der Ebene nachvollziehbar begründet werden, auf der diese erfasst werden.

[A: KGAS_4102]

Grenzwertverletzungen müssen bezüglich Risiken und Auswirkungen bewertet werden.

[A: KGAS_3571]

Basierend auf Risikobetrachtungen müssen angemessene Maßnahmen getroffen werden, um die Softwarequalität sicherzustellen.

[I: KGAS_4103]

Für die Programmiersprache "C" beschreibt KGAS_4104 geeignete Quellcodemetriken. Für andere Programmiersprachen sind diese sinngemäß zu übertragen.

5.7.3 Grafische und modellbasierte Programmierung

[R: KGAS_3947]

Dieses Kapitel gilt nur für Software (Lieferumfang), bei denen Methoden der grafischen Programmierung und/oder modellbasierten Programmierung zum Einsatz kommen.

[A: KGAS_3862]

Der Auftragnehmer muss für die gesamte Modellierung nachweislich für das Projekt geeignete Modellierungsrichtlinien anwenden. Es gelten auch die relevanten Richtlinien aus KGAS_3908.

[A: KGAS_3886]

Alle Abweichungen von der/den angewandten Modellierungsrichtlinie(n) (KGAS_3862) müssen begründet und dokumentiert werden.

[I: KGAS_3889]

Die Hierarchieebene im Modell, aus der Code generiert wird, wird als Implementierung angesehen. Diese Implementierungsebene besteht i.d.R. aus Basisobjekten, die nicht weiter verfeinert werden können, und stellt das letzte menschlich erstellte Artefakt in der Softwareerstellungskette dar.

[A: KGAS_3313]

Jedes Modellelement muss nachweislich durch ein Review geprüft werden.

[I: KGAS_4131]

Ziele von Reviews (KGAS_3313) sind mindestens: Prüfung, ob das Modell der Softwarefeinspezifikation entspricht, Prüfung nicht funktionaler Anforderungen, Prüfung Einhaltung nicht automatisch prüfbarer Modellierungsrichtlinien.

[I: KGAS_3314]

Für jedes Modellelement ist eine Beschreibung zu erstellen, die mindestens Ziel und Zweck enthält.

[I: KGAS_3456]

Im Modell sind alle Entscheidungspunkte bezüglich der Bedeutung bzw. Logik zu kommentieren.

5.7.3.1 Metriken für Grafische Programmierung

[A: KGAS_4105]

Die Modellmetriken dienen zur Messbarkeit der Qualität der grafischen Programmierung.
Die Metriken sind Indikatoren gemäß der Qualitätskriterien der ISO 25010 (siehe KGAS_3043).

[A: KGAS_3865]

Bei der grafischen Programmierung muss der Auftragnehmer geeignete Modellmetriken nach Stand der Technik mit definierten Grenzwerten anwenden.

[A: KGAS_3866]

Die Auswahl und Eignung der Modellmetriken (KGAS_3865) muss begründet sein (z.B. in einem entsprechenden Strategiedokument).

[A: KGAS_4064]

Grenzwertverletzungen müssen nachvollziehbar dokumentiert werden.

[A: KGAS_3902]

Grenzwertverletzungen müssen auf der Ebene nachvollziehbar begründet werden, auf der diese erfasst werden.

[A: KGAS_4106]

Grenzwertverletzungen müssen bezüglich Risiken und Auswirkungen bewertet werden.

[A: KGAS_3867]

Basierend auf Risikobetrachtungen müssen Maßnahmen getroffen werden, um die Softwarequalität sicherzustellen.

5.7.4 Maschinelles Lernen (ML)

[R: KGAS_4009]

Für Software (Lieferumfang) bei der Maschinelles Lernen, Neuronale Netze oder vergleichbare datenbasierte Komponenten zum Einsatz kommt, gilt die KGAS_2074.

5.7.5 Tool-Qualifizierung

[I: KGAS_3117]

Jedes softwarebasierte Tool in der Softwareerstellungs-Toolkette muss qualifiziert sein basierend auf den normativen Anforderungen der ISO 26262:2018 (KGAS_3895) sowie den Anforderungen der ISO/SAE 21434 (KGAS_4094).

[A: KGAS_3481]

Herstellerinformationen (z. B. Handbücher, Richtlinien, Fehlerverzeichnis) jedes softwarebasierten Tools müssen im Projekt nachweislich berücksichtigt werden.

5.8 Test

5.8.1 Testplanung

[A: KGAS_3556]

Ein Testplan inklusive Teststrategie gemäß ISO/IEC/IEEE 29119 (KGAS_3479) muss erstellt werden.

[I: KGAS_3334]

Im Testplan sind projektspezifische Testziele zu dokumentieren.

[I: KGAS_3335]

Im Testplan ist zu beschreiben, wie die vollständige Testabdeckung der Spezifikationen (z. B. Pflichtenheft, Schnittstellenspezifikation, Softwareanforderungsspezifikation, Softwarearchitekturspezifikation, Softwarefeinspezifikation) erreicht wird.

[A: KGAS_3364]

Black-Box-Tests müssen vor White-Box-Tests spezifiziert werden.

[I: KGAS_3657]

Der Testplan kann eine gemeinsame Teststrategie des Auftraggebers und Auftragnehmers beinhalten.

5.8.2 Testfallspezifikation

[A: KGAS_3500]

Die Testfallspezifikation muss die Anforderungen der ISO/IEC/IEEE 29119 (KGAS_3479) erfüllen.

[A: KGAS_54]

Jede Testfallspezifikation muss von jemandem erstellt werden, der das Testobjekt weder umgesetzt noch spezifiziert hat.

[A: KGAS_3359]

Etwaige Grenzwerte müssen für jede(n) Anforderung, Schnittstelle, Parameter und Entscheidungspunkt getestet sein.

[I: KGAS_3366]

Falls mehr als 10 Testfälle für die Verifikation einer Anforderung notwendig sind, muss die Qualität der Anforderung überprüft werden (z. B. daraufhin, ob diese atomar ist). Wenn die Anforderung nicht optimiert werden kann, muss eine geeignete Strukturierung der Testfälle verwendet werden.

5.8.3 Testdurchführung allgemein

[A: KGAS_3370]

Jedes Testergebnis muss einem eindeutigen Konfigurationsstand des Testobjekts (Version der Software und ggf. Hardware, Mechanik usw.) zugeordnet sein.

[A: KGAS_3372]

Die Testumgebung muss dokumentiert sein (z.B. welche Testumgebung, an welchem Prüfstand, Software- und Hardwareversion).

[A: KGAS_3685]

Wenn für einen Lieferumfang Testfälle fehlgeschlagen sind, muss der Auftragnehmer die damit verbundenen Risiken analysieren und diese dem Auftraggeber mitteilen.

5.8.4 Software Unittest

[A: KGAS_3376]

Die Software Unittests müssen eine 100%ige Abdeckung der Softwarefeinspezifikation nachweisen.

[A: KGAS_3377]

Die Software Unittests müssen mindestens eine 100%ige Zweigabdeckung (C1 oder Branch Coverage) des Quellcodes nachweisen.

[A: KGAS_3378]

Abweichungen von der in KGAS_3377 geforderten 100%igen Zweigabdeckung müssen begründet sein.

[I: KGAS_3584]

Durch Black-Box-Tests nicht abgedeckter Quellcode (siehe auch KGAS_3378) kann durch White-Box-Tests verifiziert werden.

[I: KGAS_3554]

Die Testfallspezifikation berücksichtigt mindestens die folgenden Softwarefehlerarten: Division durch Null, Bereichsüberschreitungen, Wertebereichsverletzungen, Endlosschleifen, Typ-Fehler, Initialisierungsfehler, unberechtigte Zugriffe, unerreichbarer Quellcode.

[A: KGAS_4148]

Alle Software Units müssen statisch verifiziert werden.

5.8.5 Softwareintegrationstest

[A: KGAS_3502]

Die Schnittstellen aller Softwareelemente und Komponenten müssen hinsichtlich statischer Struktur, Inhalt und Zeitverhalten getestet sein.

[I: KGAS_3383]

Softwareintegrationstests testen gegen alle Anforderungen, die durch die Softwarearchitekturspezifikation erstellt werden. Dazu zählen unter anderem Datenschnittstellen (inkl. Strukturen, zeitlicher Verlauf), Funktionsaufrufe, Zugriffe auf globale Variablen, Aufrufreihenfolgen, Ressourcenauslastung, Performance, Scheduling von Tasks, Prozesse und Interrupt Service Routines (ISR).

[A: KGAS_3636]

Für alle Tasks, Prozesse und Interrupt Service Routines (ISR) müssen die maximalen und durchschnittlichen Nettolaufzeiten (siehe KGAS_3638) auf der Zielhardware für jedes Release ermittelt und dokumentiert sein.

[I: KGAS_3638]

Die Nettolaufzeiten sind die Laufzeiten abzüglich der Laufzeitveränderungen, die durch die Messungen verursacht werden.

[A: KGAS_3637]

Die maximalen und durchschnittlichen Ressourcenverbräuche (KGAS_3282) aller Softwareelemente müssen auf der Zielhardware für jedes Release ermittelt, dokumentiert und gegen die Ressourcenanforderungen (KGAS_3282) geprüft sein.

5.8.6 Softwareverifikation

[A: KGAS_3503]

Die Softwareverifikation muss eine 100%ige Abdeckung der Softwareanforderungen nachweisen.

5.8.7 Systemintegrationsverification

[A: KGAS_3619]

Die Schnittstellen jedes Systemelements müssen hinsichtlich statischer Struktur, Inhalt und Zeitverhalten getestet sein.

5.8.8 Systemverifikation

[A: KGAS_3506]

Die Systemverifikation muss eine 100%ige Abdeckung der Systemanforderungen nachweisen.

5.9 Qualitätssicherung und -management

5.9.1 Qualitätsmanagement

[A: KGAS_53]

Die Prozess- und Produktqualitätssicherung des Auftragnehmers muss von der Entwicklung des Produkts personell und organisatorisch unabhängig sein.

[A: KGAS_2904]

Die Ziele, Bewertungsmethoden, -aktivitäten und -kriterien der Qualitätssicherung des Auftragnehmers dürfen nicht durch die Projektleitung beeinflusst werden.

[A: KGAS_3129]

Die Ziele der Qualitätssicherung müssen messbar sein.

[A: KGAS_2911]

Die Qualitätssicherung des Auftragnehmers muss am Freigabeprozess der Software-Lieferung beteiligt sein, mindestens in Form einer Qualitätsaussage.

[A: KGAS_2913]

Mitarbeitende der Qualitätssicherung des Auftragnehmers müssen die fachlichen Qualifikationen besitzen, um die fachgerechte Durchführung (inhaltlich und formal) der Reviews bestätigen zu können.

5.9.2 Review der Arbeitsprodukte

[A: KGAS_2941]

Die Reviews müssen regelmäßig von der Qualitätssicherung begleitet werden, so dass eine fachgerechte Durchführung der Reviews bestätigt werden kann.

[I: KGAS_3508]

Die Prüfkriterien eines Reviews enthalten mindestens die folgenden Punkte:

- Formale Anforderungen
- Inhaltliche Anforderungen
- Konsistenz
- Plausibilität (sowohl innerhalb des Arbeitsproduktes als auch zu dem Arbeitsprodukt, aus dem es abgeleitet wurde)
- Eindeutigkeit
- Widerspruchsfreiheit
- Wartbarkeit
- Verständlichkeit

5.9.3 Prüfung der Entwicklungsprozesse

[A: KGAS_3477]

Die Einhaltung aller Prozesse muss regelmäßig, mindestens alle 2 Monate, durch die Qualitätssicherung des Auftragnehmers geprüft werden.

[A: KGAS_2922]

Der Auftragnehmer muss den Auftraggeber über alle für den Auftraggeber relevanten Projektrisiken informieren, die aus identifizierten Defiziten resultieren.

5.10 Konfigurationsmanagement

[A: KGAS_3389]

Zu jedem Projektmeilenstein, Qualitätsmeilenstein und Release müssen die Konfigurationselemente reproduzierbar und wiederherstellbar sein.

[A: KGAS_3759]

Für alle Softwareelemente muss die verwendete Version und ggf. Patch Level dokumentiert sein, z.B. in Form einer Software Bill of Materials.

5.11 Problemlösungsmanagement

[A: KGAS_3608]

Der Auftragnehmer muss dem Auftraggeber bei jedem Release an den Auftraggeber alle offenen, für den Auftraggeber relevanten Produktprobleme mitteilen.

[A: KGAS_3417]

Die Problembeschreibung muss den Prozessschritt enthalten, in dem das Produktproblem oder die Arbeitsproduktabweichung gefunden wurde (z. B. Softwarefeinspezifikation-Review, Quellcode-Review, Unittest, Softwaretest, Systemtest).

[I: KGAS_3418]

Bei allen Produktproblemen sind folgende Informationen ersichtlich und nachvollziehbar:

- Hardwareversion
- Softwareversion
- Ausgangssituation
- Fehlerschwere
- Durchgeführte Schritte
- Erwartete Ergebnisse
- Beobachtete Ergebnisse
- Bezüge zu verletzten Spezifikationen
- Informationen zur Reproduzierbarkeit des Problems
- Quelle des Problems

[I: KGAS_3421]

In allen Beschreibungen von Produktproblemen sind für die Reproduzierbarkeit notwendige Logdateien, Traces und Messergebnisse zu verlinken.

[I: KGAS_3609]

Die Quelle des Problems ist das erste, originäre fehlerhafte Arbeitsprodukt (z. B. Anforderung, Spezifikation, Quellcode, Testspezifikation).

[A: KGAS_4132]

Probleme im Produkt müssen systematisch bis zu ihrer Fehlerursache analysiert werden.

[A: KGAS_4133]

Wenn Probleme auf Prozessschwächen zurückzuführen sind, so müssen diese abgestellt werden.

[A: KGAS_4134]

Das Problemlösungsmanagement soll verhindern, dass bekannte Probleme erneut auftreten.

5.12 Software von Dritten

[R: KGAS_3940]

Dieses Kapitel gilt für Systeme und Software (Lieferumfang), bei denen Software von Dritten zum Einsatz kommt.

[I: KGAS_3941]

Free and Open Source Software (siehe Kapitel 5.14) ist eine Variante von Software von Dritten.

[A: KGAS_3438]

Der Auftragnehmer ist verpflichtet, jede verwendete Software von Dritten in Softwareelementen einzukapseln.

[A: KGAS_3883]

Die Kapselung (KGAS_3438) ist so auszulegen, dass nur die durch Softwareanforderungen und Softwarearchitektur spezifizierten Funktionen und Schnittstellen der eingekapselten Software angesprochen werden können.

[A: KGAS_3442]

Vom Auftragnehmer entwickelte Systeme dürfen nur komplette Softwareelemente von Dritten verwenden. Eine teilweise Verwendung (z. B. über Copy & Paste Ansätze) ist nicht erlaubt.

[A: KGAS_3142]

Jedes verwendete Softwareelement von Dritten muss in der Softwarearchitekturspezifikation einzeln gekennzeichnet werden.

[A: KGAS_3531]

Für jedes verwendete Softwareelement von Dritten müssen Herkunft und Urheber bzw. Rechtsinhaber dokumentiert werden.

[A: KGAS_3437]

Für alle Softwareelemente von Dritten müssen die ursprünglichen Anforderungen, nach denen die Software von Dritten entwickelt wurde, den Softwareanforderungen zugeordnet werden.

[A: KGAS_3440]

Die Auswahl der verwendeten Softwareelemente von Dritten (inkl. Version und Patch Level) muss begründet und mit dem Auftraggeber abgestimmt werden.

[A: KGAS_3443]

Der Auftragnehmer muss sicherstellen, dass alle Softwareelemente von Dritten daraufhin geprüft werden, dass diese nur die spezifizierten Funktionen erbringen und keine anderen, möglicherweise unerwünschten Funktionen beinhalten.

[R: KGAS_3446]

Bei der Verwendung von Softwareelementen von Dritten muss der Einsatz aller für die Softwareentwicklung notwendigen Testmethoden und Teststufen für die Gesamtsoftware weiterhin möglich sein.

[R: KGAS_3923]

Der Auftragnehmer trägt die alleinige Verantwortung dafür, dass die Nutzung der Gelieferten Software vertrags- und bestimmungsgemäß zulässig ist.

5.13 Free and Open Source Software

[R: KGAS_3942]

Dieses Kapitel gilt für Systeme und Software (Lieferumfang), bei denen Free and Open Source Software zum Einsatz kommt (siehe KGAS_3820).

[A: KGAS_3822]

Die Verwendung von FOSS ist nur zulässig, wenn der Auftragnehmer den jeweiligen FOSS-Prozess des Auftraggebers beachtet und erfolgreich abgeschlossen hat, sämtliche Lizenzanforderungen der eingesetzten FOSS und die Vorgaben dieser Ziffer 5.14 erfüllt. Dies gilt auch dann, wenn die einschlägigen Lizenzbedingungen diese Verwendung sowohl in ursprünglicher als auch in bearbeiteter oder sonstiger Form ausdrücklich gestatten. Sofern die Marken Volkswagen, Volkswagen NFZ oder die AUDI AG Auftraggeber sind, darf FOSS zudem nur eingesetzt werden, wenn eine vorherige Zustimmung des Auftraggebers in Textform vorliegt.

[I: KGAS_3821]

Eine Copyleft-Lizenz ist eine Form von Nutzungs- und Lizenzbestimmungen für Open Source Software, die Bedingungen enthält, die dazu führen können, dass die mit der jeweiligen Open Source Software integrierten oder verbundenen Softwareelemente ebenfalls nur unter den jeweiligen Nutzungs- und Lizenzbestimmungen dieser Copyleft-Lizenz verbreitet werden dürfen (Auswirkung des sogenannten Copyleft Effekts). Der Auftragnehmer muss sicherstellen, dass die Gelieferte Software keine Lizenzinkompatibilitäten beinhaltet.

[A: KGAS_3833]

Der Auftragnehmer darf FOSS im Lieferumfang nicht in einer Art einsetzen, die einen Copyleft-Effekt für im Rahmen des Vertrages neu entwickelte oder vorbestehende proprietäre Software auslöst. Ausgenommen sind Anpassungen innerhalb von vorbestehenden FOSS-Komponenten (z.B. Fehlerbehebungen und Anpassungen an die konkrete Hardware) und mit dem Auftraggeber abgestimmte Einzelfälle.

[R: KGAS_3830]

Der Auftragnehmer darf in der Gelieferten Software nur solche FOSS einsetzen, die die vertrags- und bestimmungsgemäße Nutzung seiner Leistung durch den Auftraggeber und Unternehmen der Volkswagen Gruppe nicht beschränkt.

[A: KGAS_4097]

Der Auftragnehmer erteilt dem Auftraggeber mit Lieferung der Software das unterlizenzierbare und übertragbare Recht, in den Vertragsprodukten enthaltene proprietäre Software für den eigenen Gebrauch zu modifizieren und Reverse Engineering zum Zwecke der Fehlerbehebung (Debugging) solcher Bearbeitung vorzunehmen, soweit diese proprietäre Software mit unter der GNU Lesser General Public License v2.1 (LGPL-2.1) lizenzierten Software Komponenten verbunden ist.

[A: KGAS_4098]

Der Auftragnehmer stellt sicher, dass er dem Auftraggeber dieses Recht (KGAS_4097) auch in Bezug auf etwaige Softwarebestandteile Dritter einräumen kann.

[A: KGAS_3801]

Der Auftragnehmer muss dem Auftraggeber Informationen über alle in der Gelieferten Software verwendeten Free and Open Source Softwareelemente zur Verfügung stellen. Für jedes verwendete FOSS-Element müssen mindestens die folgenden Informationen enthalten sein:

- Komponenten-/Unitname
- Eindeutige Versionskennzeichnung
- Lizenzname mit eindeutiger Lizenzversionsnummer
- Vollständiger Lizenztext
- Download-Link des Lizenztexts sowie des Quellcodes inklusive letztem Zugriffsdatum
- Quellcode und Urheberrechtsvermerke
- Information, ob Quellcode und Urheberrechtsvermerke weiterzugeben bzw. zu veröffentlichen sind
- Etwaige Subelemente, die zur Verwendung des Softwareelements erforderlich sind, inklusive der vorgenannten Angaben zur Lizenzierung
- Information, ob die Lizenz eine obligatorische Bereitstellung der Lizenzinformationen an den Endkunden vorschreibt
- Schnittstelleninformationen zur Integration von Open Source Softwarekomponenten unter Ausschluss der Auslösung von Copyleft-Effekten

- Etwaige Dateien, die in der Softwarekomponente enthalten sind und unter abweichender Lizenz stehen, inklusive der vorgenannten Angaben zur Lizenzierung.

[A: KGAS_3834]

Der Auftragnehmer muss dem Auftraggeber die in KGAS_3801 geforderten Informationen mit jeder bereitgestellten Version der Software (Release, Update, Version etc.) sowie auf Anfrage des Auftraggebers zur Verfügung stellen, wobei jeweils sowohl eine vollständige Übersicht zur Verfügung gestellt werden muss als auch eine Delta-Übersicht, welche die Änderungen im Vergleich zum vorherigen Stand kenntlich macht.

[A: KGAS_3824]

Der Auftragnehmer muss die Gelieferte Software vor Auslieferung mit einer marktüblichen Analysesoftware auf enthaltene FOSS Elemente inklusive deren Abhängigkeiten und etwaiger Subelemente (u.a. Dateien) prüfen.

[A: KGAS_3828]

Auf Anfrage des Auftraggebers muss der Auftragnehmer dem Auftraggeber die Angaben, Materialien, Unterlagen und Ergebnisse der durchgeführten Analyse (KGAS_3824) zur Verfügung stellen.

[R: KGAS_3810]

Wenn der Auftragnehmer eine vom Auftraggeber patentierte oder zum Patent angemeldete technische Lösung umsetzt, dürfen keine Open Source Software Lösungen verwendet werden, deren Lizenzen die kostenpflichtige Lizenzierung des Patentes verhindern.

[A: KGAS_4135]

Die Verwendung von FOSS durch den Auftragnehmer darf außerdem nur so erfolgen, dass kein Konflikt mit der digitalen Signatur oder dem authentisierten Fahrzeugprogrammierverfahren des Auftraggebers besteht und dass Authentisierungsinformationen, kryptographische Schlüssel oder andere Informationen in Bezug auf die im Fahrzeug verwendete Software unberührt bleiben und insbesondere nicht an Dritte herausgegeben werden müssen und Dritten auch ansonsten keine Neuinstallation von (geändertem) Code im Fahrzeug ermöglicht werden muss.

[A: KGAS_4136]

Sofern der Auftraggeber vor Vertragsschluss eine Zertifizierung nach ISO/IEC 5230:2020(E) vom Auftragnehmer verlangt, übernimmt es der Auftragnehmer als wesentliche Vertragspflicht, die durch einen externen Zertifizierungsdienstleister erfolgte Zertifizierung entweder in geeigneter Form bei Vertragsschluss nachzuweisen oder diese durch einen solchen durchführen zu lassen und binnen sechs Monaten nach Vertragsschluss nachzuweisen.

5.14 Cybersecurity-relevante Entwicklung

5.14.1 Allgemeine Cybersecurity-Anforderungen

[R: KGAS_3687]

Dieses Kapitel gilt für Systeme und Software (Lieferumfang), die von der Marken-Security-Abteilung des Auftraggebers als Cybersecurity-relevant eingestuft wurden.

[A: KGAS_3738]

Der Auftragnehmer muss Cybersecurity-Risikoanalysen (Kap. 5.14.4) für den Lieferumfang auf System- und Softwareebene (auf Basis von Anforderungen und Architektur) durchführen und dokumentieren.

5.14.2 Cybersecurity-Terminologie

[I: KGAS_3703]

Bedrohungsanalyse und Risiko Assessment/Threat analysis and risk assessment - TARA
Bedrohungsanalyse und Risiko Assessment sind methodische Vorgehen, mit denen ermittelt werden kann, inwieweit das System/Element und seine Umgebung von einem Bedrohungsszenario betroffen sein können.

[I: KGAS_4138]

Eine Bedrohungsanalyse und Risiko Assessment ist auf dem eigenen Lieferumfang zu betrachten.

[I: KGAS_3704]

Wert/Asset
Werte sind für eine Institution im Sinne der Cybersecurity schützenswerte Güter.

[I: KGAS_3705]

Cybersecurity-Ziel/Cybersecurity goal
Cybersecurity-Anforderungen auf Konzeptebene, die mit einem oder mehreren Bedrohungsszenarien verbunden sind.

[I: KGAS_4139]

Schutzziel/Cybersecurity properties
Attribut, das schützenswert sein kann.

[I: KGAS_3706]

Bedrohung/Threat
Eine Bedrohung ist eine mögliche Ursache für die Kompromittierung von einem oder mehreren Schutzzielen, um ein Schadensszenario zu realisieren.

[I: KGAS_3868]

Backdoor
Eine Backdoor ist ein Zugang zu einer Software oder zu einem Hardwaresystem, die den spezifizierten Zugriff umgeht. Dabei kann der Zugang gewollt implementiert oder heimlich installiert sein.

[I: KGAS_3708]

Angriff/Attack
Eine Kette absichtlicher Handlungen zur Realisierung eines Bedrohungsszenarios.

[I: KGAS_3709]

Angriffsvektor/Attack vector
Ein Angriffsvektor ist ein potenzieller Weg, einen Angriff durchzuführen.

[I: KGAS_3710]

Risiko/Risk
Ein Risiko ist eine bezüglich möglicher Schäden durch Verletzung von Schutzzielen sowie bezüglich des für eine erfolgreiche Umsetzung nötigen Angriffsaufwands bewertete Bedrohung oder die Zusammenfassung mehrerer bewerteter Bedrohungen.

[I: KGAS_3969]

Cybersecurity-Information
Alle Informationen, die im Rahmen des Monitoring-Prozesses erfasst werden und deren Cybersecurity Relevanz (potenzielle Schwachstelle) noch nicht eingestuft ist.

[I: KGAS_3970]

Cybersecurity-Event
Cybersecurity-Information, welche als potenzielle Schwachstellen (ohne Risikobewertung) für das Unternehmen oder seine Produkte als relevant eingestuft wird und eine weitere Behandlung (CSI-Prozess, Information Assessment, etc.) bedingt.

[I: KGAS_3971]

Cybersecurity-Schwachstelle/Cybersecurity Weakness
Defekt oder Eigenschaft welche zu einem unerwünschten Verhalten führt.

[I: KGAS_3707]

Cybersecurity ausnutzbare Schwachstelle/Cybersecurity Vulnerability
Die Schwachstelle eines Wertes, die durch eine oder mehrere Angriffe ausgenutzt werden kann.

[I: KGAS_3927]

Cybersecurity-Vorfall/Cybersecurity Incident
Situation im Feld, die durch die Ausnutzung von einer Cybersecurity-Schwachstelle eintreten konnte.

[I: KGAS_3811]

Reaktionsprozess/Incident response process
Ein Reaktionsprozess ist ein definierter Prozess, der das Ziel hat, Cybersecurity Informationen und Cybersecurity-Events zu bewerten und in Entwicklung und in Serie befindliche Produkte bei einer erkannten Schwachstelle schnellstmöglich anzupassen. Damit sollen die Risiken minimiert werden (evtl. mit funktionalen Einschränkungen) und die Schwachstellen unter Wiederherstellung der vollen Funktionalität beseitigt werden.

[I: KGAS_3711]

Cybersecurity-Anforderung/Cybersecurity Requirement
Cybersecurity-Anforderungen definieren Anforderungen an den Lieferumfang, die Eigenschaften zur Abwendung bzw. Reduktion von Bedrohungen spezifizieren.

[I: KGAS_3712]

Cybersecurity-Maßnahme/Cybersecurity Control
Eine Cybersecurity-Maßnahme beschreibt die (technische) Umsetzung von Cybersecurity-Anforderungen zur Reduzierung von Risiken und bildet eine logische Gruppierung für Cybersecurity-Anforderungen, die notwendig sind, um diese Cybersecurity-Maßnahme umzusetzen.

[I: KGAS_3713]

Cybersecurity-Konzept/Cybersecurity Concept
Das Cybersecurity-Konzept ist ein Arbeitsergebnis zur Dokumentation der Cybersecurity-relevanten Aspekte des Lieferumfangs, die Bedrohungen entgegenwirken. Es umfasst insbesondere Cybersecurity-Maßnahmen, berücksichtigte Einschränkungen, Architekturen sowie die getroffenen Annahmen und Randbedingungen.

[I: KGAS_3715]

Schützenswerte Daten/Data worthy of protection
Schützenswerte Daten sind Daten, die durch das Cybersecurity-Konzept bzw. durch Cybersecurity-Maßnahmen geschützt werden müssen.

[I: KGAS_3717]

Vertrauenswürdig/Trustworthy
Als vertrauenswürdig kann ein System, eine Datenquelle, usw. eingestuft werden, wenn ein Nachweis existiert, dass sich auf dieses bis zu einem bestimmten Ausmaß verlassen werden kann und keine Kompromittierung vorliegt.

[I: KGAS_3718]

Vertrauensgrenze/Trust boundary
Eine Vertrauensgrenze beschreibt den Übergang zwischen verschiedenen Ebenen des Vertrauens.

[I: KGAS_3720]

OWASP (Open Web Application Security Project)
OWASP ist eine Online Community, die unter anderem einen Standard zur Durchführung von Sicherheitsverifizierungen auf Applikationsebene zur Verfügung stellt.
Referenz: https://www.owasp.org/

[I: KGAS_3721]

CWE (Common Weakness Enumeration)
CWE ist eine Sammlung von Softwareschwachstellen, die durch eine Online Community zur Verfügung gestellt wird.
Referenz: https://cwe.mitre.org/

[I: KGAS_3904]

CVE (Common Vulnerabilities and Exposures)
CVE® ist eine Liste von Einträgen, die jeweils eine Identifikationsnummer, eine Beschreibung und mindestens eine öffentliche Referenz für bekannte Schwachstellen in Bezug auf Cybersecurity enthalten.
Referenz: https://cve.mitre.org/

5.14.3 Cybersecurity-Management

[A: KGAS_3851]

Nach einem erkannten unautorisierten Zugriff auf das Konfigurationsmanagementsystem muss der ursprüngliche Zustand der Konfigurationselemente wiederhergestellt werden und der Auftraggeber ist darüber zu informieren.

5.14.4 Cybersecurity-Risikoanalyse

[A: KGAS_4141]

Der Auftragnehmer muss Cybersecurity Risikoanalysen auf Systemebene und je nach Notwendigkeit auf Softwareebene und auf Softwareelementebene durchführen.

[A: KGAS_3740]

Im Rahmen der Cybersecurity-Risikoanalysen sind auch alle zu verarbeitenden Daten als Asset zu identifizieren.

[A: KGAS_3741]

Im Rahmen der Cybersecurity-Risikoanalysen sind alle Schnittstellen von und zu der beauftragten Software als Asset zu identifizieren.

[A: KGAS_3974]

Der Auftragnehmer muss aktuelle Bedrohungs- und Maßnahmenkataloge pflegen.

[A: KGAS_3743]

Für jede Bedrohung in einer Cybersecurity-Risikoanalyse ist das Risiko systematisch nach einem vom Auftragnehmer spezifizierten Bewertungsschema zu bewerten (z.B. Tabelle G.7 in ISO/SAE 21434).

[I: KGAS_4143]

Wenn Auswirkungen auf das Bauteil nicht bewertbar sind, können zur Evaluierung die Auswirkungen auf die Funktion herangezogen werden.

[A: KGAS_3744]

Der Auftragnehmer muss identifizierte und/oder vorgegebene Cybersecurity-Anforderungen in den Cybersecurity-Risikoanalysen berücksichtigen.

[A: KGAS_3750]

Cybersecurity-Maßnahmen müssen nachweislich zu Cybersecurity-Anforderungen führen.

5.14.5 Cybersecurity-Risikomanagement

[A: KGAS_3745]

Nach Änderungen auf System- und/oder Softwareebene müssen die Cybersecurity-Risikoanalysen sowie das Cybersecurity-Konzept entsprechend aktualisiert werden.

[A: KGAS_3980]

Identifizierte Schwachstellen müssen bis zur akzeptablen Minimierung des Risikos nachweislich gemanagt werden.

5.14.6 Cybersecurity-Architektur und Cybersecurity-Design

[A: KGAS_3755]

Alle Datenquellen müssen identifiziert und als vertrauenswürdig oder nicht vertrauenswürdig klassifiziert werden.

[I: KGAS_3855]

Datenquellen, die sich außerhalb der definierten Vertrauensgrenzen befinden, sind nicht vertrauenswürdig, und Datenquellen, die sich innerhalb der definierten Vertrauensgrenzen befinden, sind vertrauenswürdig. Der Lieferumfang muss nicht zwingend eine Vertrauensgrenze bilden. Der Lieferumfang kann auch mehrere Vertrauensgrenzen haben, z.B. bei mehreren µC.

[A: KGAS_3756]

Alle Daten, die aus nicht vertrauenswürdigen Quellen stammen, müssen vor der Verarbeitung validiert werden.

[A: KGAS_3758]

Fehlermeldungen, Logeinträge und Diagnoseeinträge dürfen keine sensitiven Daten enthalten, über die die Cybersecurity des Steuergerätes bzw. der Software gefährdet werden könnte.

[A: KGAS_3929]

Für die Analyse seines Lieferumfangs muss der Auftragnehmer geeignete Quellen für die Identifikation von Schwachstellen bestimmen und gegen diese prüfen.

[A: KGAS_3957]

Der Lieferumfang darf keine bekannten Schwachstellen enthalten. Abweichungen müssen in den Risikoanalysen berücksichtigt und begründet werden.

[A: KGAS_3930]

Quellen für die Identifikation von Schwachstellen müssen unter anderem Publikationen der CWE (KGAS_3721), der CVE (KGAS_3904) oder Meldungen des Auftraggebers sein oder vergleichbare.

[A: KGAS_3761]

Alle Architekturelemente, die keinen funktionellen Aspekt erfüllen, müssen kenntlich gemacht werden (z.B. Testschnittstellen). Diese potentiellen Einfallstore dürfen zur Seriensoftware nicht mehr zugänglich sein.

5.14.7 Cybersecurity-Implementierung

[A: KGAS_3762]

Neben der Anwendung geeigneter Codierrichtlinien oder Modellierungsrichtlinien (KGAS_3908) muss der Auftragnehmer Cybersecurity-Codierrichtlinien anwenden.

[A: KGAS_3772]

Der Auftragnehmer muss Codeanalysen durchführen, in denen die Einhaltung der KGAS_3762 geprüft wird.

[I: KGAS_3872]

Die Cybersecurity-Codeanalysen können sowohl manuell als auch toolgestützt durchgeführt werden.

[A: KGAS_4144]

Es muss sichergestellt werden, dass keine ungewollten Zugänge (Backdoors) implementiert sind.

[A: KGAS_3764]

Jede Abweichung zu den Anforderungen KGAS_3762, KGAS_4144, KGAS_3896 muss begründet und dokumentiert werden.

[A: KGAS_3765]

Umfasst der beauftragte Lieferumfang eine Web-Applikation, müssen die Richtlinien der OWASP (KGAS_3720) eingehalten werden.

5.14.8 Cybersecurity-Nachweis

[A: KGAS_3775]

Ein Cybersecurity-Nachweis muss vom Auftragnehmer zum Meilenstein "Function Complete" (100 % Software-Funktionalität ist implementiert) erbracht werden.

[A: KGAS_3931]

Der Cybersecurity-Nachweis muss spätestens zur 0-Serie, um den Nachweis des gegen unbefugte Zugriffe und Manipulation abgesicherten Flash Prozesses, ergänzt werden.

[A: KGAS_3818]

Der Cybersecurity-Nachweis muss bei Änderungen bis Serienlieferung stetig aktualisiert werden.

[A: KGAS_3776]

Der Cybersecurity-Nachweis muss die Ergebnisse der im Cybersecurity-Plan geplanten Cybersecurity-Aktivitäten beinhalten.

[A: KGAS_3817]

Der Cybersecurity-Nachweis muss eine Zusammenfassung der Ergebnisse der Cybersecurity-Risikoanalysen beinhalten.

[A: KGAS_3983]

Der Cybersecurity-Nachweis muss die Angemessenheit und die Effektivität der Cybersecurity-Maßnahmen beinhalten.

[I: KGAS_3873]

Die Risikoanalysen sowie die detaillierten Ergebnisse der Risikoanalysen können im Rahmen einer Technischen Revision beim Auftragnehmer eingesehen werden.

[A: KGAS_3777]

Der Cybersecurity-Nachweis muss aufzeigen, dass die Cybersecurity-Anforderungen umgesetzt und verifiziert wurden.

[A: KGAS_3819]

Der Cybersecurity-Nachweis muss aufzeigen, dass die Cybersecurity-Codierrichtlinien (Codierrichtlinien siehe KGAS_3762) eingehalten wurden.

[A: KGAS_3932]

Der Cybersecurity-Nachweis muss aufzeigen, dass der Reaktionsprozess zum Umgang mit identifizierten Schwachstellen (KGAS_3877) und die aktive Überwachung des Lieferumfangs (KGAS_3784) etabliert sind.

[A: KGAS_3984]

Der Cybersecurity-Nachweis muss durch eine vom Projekt unabhängige Instanz geprüft werden.

5.15 Cybersecurity-Aktivitäten nach der Entwicklungsphase (Serien-/Feldbetreuung)

[A: KGAS_3874]

Der Auftragnehmer hat eine für die Sicherheit seines Projektumfangs verantwortliche Person (Chief Information Security Officer, Chief Product Security Officer, o.ä.) zu benennen und eine funktionsfähige E-Mail-Adresse als Ansprechpartner für Nachrichten des Auftraggebers einzurichten. Der Auftragnehmer hat die Informationen in der Lieferantendatenbank (zugänglich über die Business-Plattform One.Group) zu speichern und bei Bedarf zu aktualisieren.

[A: KGAS_3985]

Der Auftragnehmer muss zum Austausch von Daten die von VW eingesetzten Mechanismen und Standards zur E-Mail-Verschlüsselung gem. IT-Sicherheitsrichtlinien unterstützen.
(siehe KGAS_4087 und KGAS_4088)

[A: KGAS_3877]

Der Auftragnehmer hat einen Ansprechpartner für einen bidirektionalen Antwortprozess zur Bearbeitung von Cybersicherheitsmeldungen zu benennen und Anfragen des Auftraggebers innerhalb einer angemessenen Frist zu beantworten.

[A: KGAS_3784]

Der Auftragnehmer muss einen Prozess zur aktiven und kontinuierlichen Überwachung des Cybersicherheitsstatus für den Projektumfang gemäß ISO21434 etablieren.

[A: KGAS_3785]

Wenn Cybersecurity-Informationen, Ereignisse, Cybersecurity-Schwachstellen, Cybersecurity ausnutzbare Schwachstellen und Cybersecurity-Vorfälle auftreten, muss der etablierte Reaktionsprozess (KGAS_3877) befolgt werden.

[A: KGAS_3933]

Sollte der Auftragnehmer eine Unterauftragnehmerkette benötigen, um das Produkt für den Auftraggeber zu liefern, muss der Auftragnehmer auch in seiner Unterauftragnehmerkette Reaktionsfähigkeit und Effektivität sicherstellen.

[A: KGAS_3934]

Erlangt der Auftragnehmer Kenntnis von Fällen der Kategorie Schwachstelle, ausnutzbare Schwachstelle oder Vorfall, die im Rahmen des Projekts enthalten sind, so hat er den Auftraggeber

unverzüglich zu informieren. Bei Verdacht auf bereits in Verkehr gebrachte Produkte des Auftraggebers ist die Meldung an das Störungsteam des Auftraggebers unter der in KGAS angegebenen E-Mail-Adresse zu richten.

[A: KGAS_3986]

Erfolgt die erste Anzeige (KGAS_3934) durch den Auftraggeber, so hat der Auftragnehmer innerhalb einer üblichen Frist von zwei Werktagen eine Empfangsbestätigung zu übersenden, die eine Rückmeldung des Auftragnehmers enthält, ob die gelieferten Produkte vom Gegenstand der Mitteilung betroffen oder nicht betroffen sind. Bei verspäteter Information des Auftragnehmers ist einvernehmlich eine gemeinsame Statusbesprechung zu vereinbaren.

[A: KGAS_3988]

Die Empfangsbestätigung muss einen eindeutigen Verweis enthalten. Der Auftraggeber und der Auftragnehmer einigen sich auf eine eindeutige Referenz, die in der Kommunikation verwendet wird.

[A: KGAS_3939]

Jegliche Kommunikation des Auftragnehmers in Bezug auf Cybersecurity-Fälle erfolgt nach dem Need-to-know-Prinzip.

[A: KGAS_3936]

Plant der Auftragnehmer eine externe Kommunikation, die den Auftraggeber betrifft, so ist diese mit dem Incident Team des Auftraggebers abzustimmen. Dies gilt nicht für die Kommunikation aufgrund gesetzlicher Vorgaben. Im Falle einer Kommunikation aufgrund gesetzlicher Vorgaben ist der Auftraggeber über das jeweilige Cybersecurity Incident Management zu informieren.

[A: KGAS_3937]

Innerhalb von in der Regel 10 Arbeitstagen nach Bestätigung des Anliegens muss eine detaillierte technische Analyse, Risikobewertung, einschließlich Ursache, Auswirkungen und möglicher Abhilfemaßnahmen an das zuständige Cybersecurity Incident Management des Volkswagen Konzerns (KGAS_3890) übermittelt werden.

[A: KGAS_3989]

Die Analyse muss in Übereinstimmung mit den Metriken der ISO21434 erfolgen und mindestens die Kategorisierung und Beschreibung des Verdachtsfalls, eine Beschreibung des Angriffsweges, eine Beschreibung der Auswirkungen, eine Bewertung der Durchführbarkeit des Auftretens, der betroffenen Produkte, Scopes, Teile, Komponenten, Systeme und Projekte sowie die spezifische Softwareversion und das wahrscheinliche Risiko umfassen. Bei Bedarf kann der Umfang der Erstanalyse erweitert werden, wenn dies zwischen dem Auftragnehmer und dem Auftraggeber vereinbart wird. Bei verspäteter Information des Auftragnehmers ist einvernehmlich eine gemeinsame Statusbesprechung zu vereinbaren.

[A: KGAS_3990]

Im Falle einer vom Auftragnehmer bereitgestellten Lösung ist diese auf Anfrage detailliert zu dokumentieren und muss folgendes enthalten:

- Unterschied zu vorher und nachher der Veränderung des Produktes (Bspw. bei Soft- oder Hardware)
- Beschreibung der Tests/Szenarien zur Wirksamkeitskontrolle
- die Testergebnisse

[A: KGAS_3991]

Auf Verlangen hat der Auftragnehmer auf eigene Kosten Hard- und/oder Softwaremuster zur Verfügung zu stellen, die es der Volkswagen AG ermöglichen, die bereitgestellte Lösung und die Schwachstelle zu verifizieren.

[A: KGAS_3992]

Identifizierte Cybersecurity ausnutzbare Schwachstellen müssen in aktuellen Entwicklungen berücksichtigt werden (Siehe KGAS_3746).

6 Referenzierte Unterlagen

6.1 Dokumente der Volkswagen AG

[I: KGAS_2834]

Formel Q Fähigkeit Software: Qualitätsfähigkeit Lieferanten Beurteilungsrichtlinie für Software-Entwicklungsprozesse [Volkswagen AG; Software-Qualitätssicherung]
Erhältlich unter http://www.vwgroupsupply.com/

[I: KGAS_3908]

Liste von Codier-/Modellierungsrichtlinien: Auflistung üblicher Codierrichtlinien und Modellierungsrichtlinien im Automotive Kontext [Volkswagen AG; Software-Qualitätssicherung]
Erhältlich unter http://www.vwgroupsupply.com/

[I: KGAS_4093]

Smart Quality Analytics (SQA): Das ist der Mindestsatz von Projektmetriken. [Volkswagen AG; Software-Qualitätssicherung]
Erhältlich unter http://www.vwgroupsupply.com/

[I: KGAS_4116]

ReleaseNotes: Dokumentation des Lieferumfangs und definierter Metriken.
[Volkswagen AG; Software-Qualitätssicherung]
Erhältlich unter http://www.vwgroupsupply.com/

[I: KGAS_3966]

Richtlinie für die datenschutzrechtlichen Anforderungen bei der (Weiter-)Entwicklung von Steuergeräten mit Speicherfunktion
Erhältlich unter http://www.vwgroupsupply.com/

[I: KGAS_4003]

Allow-List FOSS-Lizenzen KGAS: Diese Liste enthält FOSS-Lizenzen, die vom Auftragnehmer in der Regel bedenkenlos eingesetzt werden können.
Erhältlich unter http://www.vwgroupsupply.com/

[I: KGAS_4087]

Leitfaden - Sicherer Datenaustausch
Erhältlich unter http://www.vwgroupsupply.com/

[I: KGAS_4088]

IS-Regelung Nr. 02.06. Handlungsleitlinie für Dritte
Erhältlich unter http://www.vwgroupsupply.com/
(für Beschäftigte der Volkswagen AG siehe KGAS_4089)

[I: KGAS_4089]

IS-Regelung Nr. 02.02 Handlungsleitlinie für Beschäftigte (Gültig für Volkwagen AG)
Erhältlich unter https://volkswagen-net.de/wikis/display/ISRegelwerk/IS+Regelungen

[I: KGAS_4147]

LAH.893.909.D Besondere Merkmale in Software und/oder Umgang mit nicht beauftragten Softwareumfängen
Erhältlich unter http://www.vwgroupsupply.com/

[I: KGAS_4104]

VW SW Source Code Metrics: Das ist der Satz der Quellcodemetriken. [Volkswagen AG; Software-Qualitätssicherung]
Erhältlich unter http://www.vwgroupsupply.com/

6.2 Dokumente des Verbands der Automobilindustrie (VDA)

[I: KGAS_3887]

Automotive SPICE® Process Assessment / Reference Model (PAM/PRM) - RELEASE 4.0 oder höher.

6.3 Internationale Standards und Normen

[I: KGAS_3043]

ISO/IEC 25010:2023 Systems and software engineering -- Systems and software Quality Requirements and Evaluation ("SQuaRE") – Product quality model

[I: KGAS_3479]

ISO/IEC/IEEE 29119:2022 Software and systems engineering - Software testing

[I: KGAS_3895]

ISO 26262:2018 Road vehicles -- Functional safety

[I: KGAS_4094]

ISO/SAE 21434:2021 Road vehicles – Cybersecurity engineering

7 Release Notes

[I: KGAS_4055]

Die Tabelle mit den Änderungen zur vorherigen Version finden sie unter http://www.vwgroup-supply.com/.

8 Vertraulichkeitshinweis

[R: KGAS_3488]