

ISO/IEC/IEEE 12207:2026 Systems and software engineering, Software life cycle processes (2. Ausgabe, April 2026)

Prozessrahmen für Software-Lebenszyklus

63 Anforderungen. Für diese Norm gibt es kein akkreditiertes Zertifikat. Stand 07/2026.

ISO/IEC/IEEE 12207 beschreibt einen Rahmen aus 30 Prozessen für den gesamten Lebenszyklus von Software, von der Vereinbarung über Entwicklung und Betrieb bis zur Außerbetriebnahme. Es gibt kein akkreditiertes Zertifizierungsschema: Die Norm regelt Konformität ausdrücklich als Selbstdeklaration in drei Varianten (vollständige Konformität zu den Ergebnissen, vollständige Konformität zu den Aufgaben oder maßgeschneiderte Konformität mit dokumentiertem Zuschnitt), das Ergebnis einer Prüfung ist also eine Konformitätserklärung und kein Zertifikat. Wer eine externe Bewertung braucht, lässt ein Prozess-Assessment nach der ISO/IEC-33000-Reihe durchführen, das liefert ein Fähigkeitsprofil als Bericht.

Betrieb	Datum	Bearbeiter
---------	-------	------------

6.1 Vereinbarungsprozesse (Agreement Processes)6

6.1.1Der Beschaffungsprozess ist etabliert: Bedarf an zugekaufter Software oder Dienstleistung wird identifiziert, Lieferanten werden ausgewählt und eine Vereinbarung wird geschlossen und überwacht.Dokument

Woran man es festmacht: Belege sind Ausschreibungs- oder Anforderungsunterlagen, Angebotsvergleiche, unterzeichnete Verträge oder Bestellungen sowie die Abnahmeprotokolle. In kleinen Teams reicht ein kurzes Beschaffungsticket je Zukauf im Issue-Tracker mit Anforderung, Alternativenvergleich und Freigabe.

OffenIn ArbeitErfülltNicht anwendbar

Nachweis / Begründung

6.1.1.2Die Ergebnisse der Beschaffung liegen vor: Anforderungen an das zugekaufte Element sind definiert, die Vereinbarung ist geschlossen und die Erfüllung ist bestätigt.

Woran man es festmacht: Abnahmekriterien im Vertrag oder im Ticket, dazu der Abnahmetest oder das Abnahmeprotokoll. Bei Open-Source-Abhängigkeiten dient die Lockfile mit dokumentiertem Auswahlkriterium und der Lizenz-Scan als Abnahmenachweis.

OffenIn ArbeitErfülltNicht anwendbar

Nachweis / Begründung

6.1.1.3Die Beschaffungsaktivitäten sind durchgeführt: Bedarfsermittlung, Lieferantenauswahl, Vertragsschluss, Fortschrittsüberwachung und Abnahme sind nachvollziehbar erfolgt.

Woran man es festmacht: Aufzeichnungen aus dem Issue-Tracker und aus dem Vertragsordner, ergänzt um Statusmeldungen des Lieferanten. Ein Lieferantenverzeichnis mit Bewertungsdatum genügt für kleine Betriebe.

OffenIn ArbeitErfülltNicht anwendbar

Nachweis / Begründung

6.1.2Der Lieferprozess ist etabliert: Die Organisation prüft Anfragen, gibt ein Angebot ab, schließt eine Vereinbarung, liefert das Produkt oder die Dienstleistung und schließt die Vereinbarung formal ab.Dokument

Woran man es festmacht: Angebote, Auftragsbestätigungen, Liefer- und Release-Notes, Übergabeprotokolle und Rechnungen. Die Release-Notes je Version sind der praktikabelste Lieferbeleg für Software-Teams.

OffenIn ArbeitErfülltNicht anwendbar

Nachweis / Begründung

6.1.2.2Die Ergebnisse der Lieferung liegen vor: Der Lieferumfang ist vereinbart, das Produkt ist übergeben und die Verantwortung ist geregelt.

Woran man es festmacht: Signierte Release-Artefakte mit Versionsnummer, Changelog und Übergabemail an den Kunden. Ergänzend Support- und Gewährleistungsregelung in den Vertragsbedingungen.

OffenIn ArbeitErfülltNicht anwendbar

Nachweis / Begründung

6.1.2.3 Die Lieferaktivitäten sind durchgeführt: Machbarkeit wurde bewertet, die Lieferung wurde ausgeführt und die Vereinbarung wurde nach Erfüllung abgeschlossen.

Woran man es festmacht: Aufzeichnung der Machbarkeits- oder Aufwandsschätzung im Angebot, Deployment-Log der CI-Pipeline als Lieferbeleg und ein kurzes Abschlussprotokoll je Auftrag.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.2 Organisationale, projektermöglichende Prozesse (Organizational Project-Enabling Processes)

9

6.2.1 Das Lebenszyklusmodell-Management ist etabliert: Die Organisation definiert, pflegt und stellt die Lebenszyklusmodelle, Prozesse und Verfahren bereit, die in Projekten verwendet werden.

[Dokument](#)

Woran man es festmacht: Eine schlanke Prozessbeschreibung im Repository, etwa CONTRIBUTING.md plus eine Seite Entwicklungs-Workflow mit Branching-Modell, Definition of Done und Release-Kadenz. Änderungen daran sind über die Git-Historie datiert nachweisbar.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.2.1.2 Die Ergebnisse liegen vor: Die eingesetzten Lebenszyklusmodelle sind festgelegt, sind den Projekten bekannt und werden regelmäßig auf Wirksamkeit überprüft.

Woran man es festmacht: Retrospektiven-Notizen oder ein jährlicher Review-Eintrag im Repository, der die Anpassung des Workflows begründet. Pull Requests auf die Prozessdokumente sind der Wirksamkeitsnachweis.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.2.2 Das Infrastrukturmanagement ist etabliert: Die für Entwicklung, Test und Betrieb benötigte Infrastruktur ist definiert, bereitgestellt und wird gepflegt.

Woran man es festmacht: Infrastructure-as-Code im Repository (Terraform, Docker-Compose, CI-Runner-Konfiguration), dazu die Liste der genutzten Dienste mit Verantwortlichen. Für kleine Teams genügt eine Seite mit Umgebungen, Zugängen und Backup-Regelung.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.2.2.3 Die Infrastruktur-Aktivitäten sind durchgeführt: Bedarf ist ermittelt, Ressourcen sind beschafft, betrieben und bei Bedarf außer Betrieb genommen.

Woran man es festmacht: Änderungshistorie der Infrastruktur-Konfiguration in Git, Monitoring-Dashboards und Tickets zu Umgebungsänderungen. Abgeschaltete Umgebungen sind im Ticket mit Datum dokumentiert.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.2.3 Das Portfoliomanagement ist etabliert: Projekte werden initiiert, priorisiert, mit Ressourcen versehen, fortlaufend bewertet und bei fehlendem Nutzen beendet.

Woran man es festmacht: Produkt-Roadmap oder Portfolio-Board mit Prioritäten und Entscheidungsdatum, ergänzt um Aufzeichnungen zu gestoppten Vorhaben. Ein Roadmap-Dokument mit Quartalsreview reicht für kleine Organisationen aus.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.2.4 Das Personalmanagement ist etabliert: Benötigte Kompetenzen sind bestimmt, Personal ist qualifiziert und Wissen wird aufgebaut und erhalten.

Woran man es festmacht: Kompetenzmatrix je Rolle, Schulungs- und Zertifikatsnachweise, Onboarding-Checkliste. In kleinen Teams genügt eine Skill-Tabelle mit Lücken und geplanten Schulungen plus Nachweis absolvierter Trainings.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.2.5	Das Qualitätsmanagement ist etabliert: Qualitätsziele, Qualitätspolitik und Verantwortlichkeiten sind festgelegt und die Zielerreichung wird bewertet.	Dokument
Woran man es festmacht: Kurze Qualitätsleitlinie im Repository, definierte Qualitätsziele (Testabdeckung, Fehlerdichte, Antwortzeit bei Störungen) und die zugehörigen Kennzahlen aus CI und Issue-Tracker.		
OffenIn ArbeitErfülltNicht anwendbar		
Nachweis / Begründung		

6.2.5.2	Die Ergebnisse des Qualitätsmanagements liegen vor: Qualitätsziele sind messbar, Abweichungen werden erkannt und Korrekturmaßnahmen werden verfolgt.	
Woran man es festmacht: Trendberichte aus der CI-Pipeline (Coverage, fehlgeschlagene Builds), Fehlerstatistik aus dem Issue-Tracker und Tickets zu abgeleiteten Maßnahmen mit Abschlussdatum.		
OffenIn ArbeitErfülltNicht anwendbar		
Nachweis / Begründung		

6.2.6	Das Wissensmanagement ist etabliert: Wissen, Fertigkeiten und wiederverwendbare Artefakte werden erfasst, geteilt und für spätere Projekte verfügbar gehalten.	
Woran man es festmacht: Interne Wissensbasis oder Wiki, Architecture Decision Records im Repository, wiederverwendbare Bibliotheken mit Dokumentation. Postmortem-Notizen nach Störungen sind ein starker Beleg.		
OffenIn ArbeitErfülltNicht anwendbar		
Nachweis / Begründung		

6.3 Technische Managementprozesse (Technical Management Processes)		16
6.3.1	Die Projektplanung ist etabliert: Umfang, Ergebnisse, Termine, Aufwände, Rollen und Schnittstellen sind geplant und der Plan wird gepflegt.	Dokument
Woran man es festmacht: Projektplan oder Backlog mit Meilensteinen, Schätzungen und Zuständigkeiten. Ein gepflegtes Sprint- oder Milestone-Board im Issue-Tracker erfüllt die Anforderung, wenn Umfang und Termine erkennbar sind.		
OffenIn ArbeitErfülltNicht anwendbar		
Nachweis / Begründung		

6.3.1.2	Die Planungsergebnisse liegen vor: Aufgaben und Ressourcen sind zugeordnet, Rollen sind benannt und die Pläne sind abgestimmt.	
Woran man es festmacht: Zugewiesene Issues mit Verantwortlichem und Zielversion, dazu die Meilenstein-Übersicht. Kapazitätsannahmen kurz im Planungsticket festhalten.		
OffenIn ArbeitErfülltNicht anwendbar		
Nachweis / Begründung		

6.3.1.3	Die Planungsaktivitäten sind durchgeführt: Der Projektumfang ist definiert, die Pläne sind erstellt und bei Änderungen aktualisiert.	
Woran man es festmacht: Versionshistorie des Plans oder des Roadmap-Dokuments in Git, Änderungskommentare an Meilensteinen. Damit ist die Aktualisierung datiert belegbar.		
OffenIn ArbeitErfülltNicht anwendbar		
Nachweis / Begründung		

6.3.2	Projektbewertung und -steuerung sind etabliert: Fortschritt gegen Plan wird bewertet, Abweichungen werden erkannt und Steuerungsmaßnahmen werden getroffen.	Dokument
Woran man es festmacht: Statusberichte, Sprint-Reviews, Burndown- oder Durchsatzkennzahlen aus dem Issue-Tracker sowie Protokolle der Entscheidungen bei Abweichungen. Ein kurzes wöchentliches Statusnotat genügt.		
OffenIn ArbeitErfülltNicht anwendbar		
Nachweis / Begründung		

6.3.3	Das Entscheidungsmanagement ist etabliert: Entscheidungen werden strukturiert vorbereitet, Alternativen bewertet, die Auswahl begründet und aufgezeichnet.	Dokument
Woran man es festmacht: Architecture Decision Records im Repository mit Kontext, Optionen, Entscheidung und Konsequenz. Für kleinere Entscheidungen reicht der begründete Kommentar im Pull Request.		
Offen In Arbeit Erfüllt Nicht anwendbar		
Nachweis / Begründung		
6.3.3.2	Die Ergebnisse des Entscheidungsmanagements liegen vor: Entscheidungsbedarf ist erkannt, Alternativen sind bewertet und die getroffene Entscheidung ist mit ihrer Begründung nachvollziehbar.	
Woran man es festmacht: Der ADR-Ordner mit fortlaufender Nummerierung und Status (vorgeschlagen, akzeptiert, abgelöst). Die Git-Historie belegt Zeitpunkt und Autor.		
Offen In Arbeit Erfüllt Nicht anwendbar		
Nachweis / Begründung		
6.3.4	Das Risikomanagement ist etabliert: Risiken werden fortlaufend identifiziert, bewertet, behandelt und überwacht.	Dokument
Woran man es festmacht: Risikoregister mit Eintrittswahrscheinlichkeit, Auswirkung, Maßnahme und Verantwortlichem. Technische Risiken lassen sich als Issues mit Label risk führen, dazu die Ergebnisse von Abhängigkeits- und Schwachstellen-Scans.		
Offen In Arbeit Erfüllt Nicht anwendbar		
Nachweis / Begründung		
6.3.4.2	Die Ergebnisse des Risikomanagements liegen vor: Risiken sind priorisiert, Schwellen für Maßnahmen sind festgelegt und die Wirksamkeit der Behandlung wird verfolgt.	
Woran man es festmacht: Aktualisierungshistorie des Risikoregisters, geschlossene Risiko-Tickets mit Begründung sowie Reports aus dem automatisierten Dependency-Scanning mit Behebungsdatum.		
Offen In Arbeit Erfüllt Nicht anwendbar		
Nachweis / Begründung		
6.3.5	Das Konfigurationsmanagement ist etabliert: Konfigurationselemente sind identifiziert, Baselines sind definiert, Änderungen sind kontrolliert und der Status ist jederzeit feststellbar.	Dokument
Woran man es festmacht: Git als Versionskontrolle mit Tags für Releases, geschützte Hauptbranches, Lockfiles für Abhängigkeiten und eine Software-Stückliste (SBOM) je Release. Die Tag-Historie ist die Baseline-Aufzeichnung.		
Offen In Arbeit Erfüllt Nicht anwendbar		
Nachweis / Begründung		
6.3.5.2	Die Ergebnisse des Konfigurationsmanagements liegen vor: Baselines sind gesetzt, Änderungen sind rückverfolgbar und Auslieferungen sind reproduzierbar.	
Woran man es festmacht: Reproduzierbarer Build aus einem Git-Tag über die CI-Pipeline, dazu Build-Logs und Artefakt-Registry mit Prüfsummen.		
Offen In Arbeit Erfüllt Nicht anwendbar		
Nachweis / Begründung		
6.3.5.3	Die Aktivitäten des Konfigurationsmanagements sind durchgeführt: Änderungsanträge werden bewertet, freigegeben oder abgelehnt und ihr Einbau wird verifiziert.	
Woran man es festmacht: Pull-Request-Historie mit verpflichtendem Review und verknüpftem Issue, Merge-Regeln im Repository (Branch Protection) als Beleg der Freigabekontrolle.		
Offen In Arbeit Erfüllt Nicht anwendbar		
Nachweis / Begründung		

6.3.6	Das Informationsmanagement ist etabliert: Zu erzeugende und aufzubewahrende Informationen sind bestimmt, sie werden gepflegt, geschützt und den Berechtigten bereitgestellt.	Dokument		
Woran man es festmacht: Ablagekonzept mit Speicherort, Verantwortlichem, Aufbewahrungsfrist und Zugriffsrechten. Repository-Berechtigungen und Backup-Protokolle belegen Schutz und Verfügbarkeit.				
	Offen	In Arbeit	Erfüllt	Nicht anwendbar
Nachweis / Begründung				

6.3.7	Der Messprozess ist etabliert: Informationsbedarfe sind bestimmt, passende Kennzahlen sind definiert, erhoben, ausgewertet und für Entscheidungen genutzt.			
Woran man es festmacht: Kennzahlen aus CI und Issue-Tracker, etwa Durchlaufzeit, Fehlerrate, Testabdeckung, Deployment-Häufigkeit. Wichtig ist der belegte Weg von der Kennzahl zur Entscheidung, etwa im Retrospektiven-Protokoll.				
	Offen	In Arbeit	Erfüllt	Nicht anwendbar
Nachweis / Begründung				

6.3.7.2	Die Ergebnisse der Messung liegen vor: Die erhobenen Daten sind gültig, die Auswertung ist dokumentiert und die Ergebnisse werden kommuniziert.			
Woran man es festmacht: Automatisch erzeugte Reports der Pipeline mit Zeitstempel und ein regelmäßiger Kennzahlen-Auszug im Team-Kanal oder Wiki.				
	Offen	In Arbeit	Erfüllt	Nicht anwendbar
Nachweis / Begründung				

6.3.8	Die Qualitätssicherung ist etabliert: Produkte und Prozesse werden unabhängig gegen die festgelegten Vorgaben geprüft, Abweichungen werden aufgezeichnet und behoben.	Dokument		
Woran man es festmacht: Verpflichtendes Vier-Augen-Review im Pull Request, Linter- und Testläufe als Merge-Gate, Aufzeichnung der gefundenen Abweichungen als Issues. Die Unabhängigkeit erreicht ein kleines Team durch rollierende Reviewer statt einer eigenen QS-Abteilung.				
	Offen	In Arbeit	Erfüllt	Nicht anwendbar
Nachweis / Begründung				

6.3.8.2	Die Ergebnisse der Qualitätssicherung liegen vor: Prozesskonformität und Produktkonformität sind bewertet und Korrekturmaßnahmen sind bis zum Abschluss verfolgt.			
Woran man es festmacht: Review-Kommentare mit Erledigungsstatus, geschlossene Bug-Tickets mit Verweis auf den Fix-Commit und der grüne Pipeline-Lauf des Release-Tags.				
	Offen	In Arbeit	Erfüllt	Nicht anwendbar
Nachweis / Begründung				

6.4 Technische Prozesse (Technical Processes)					26
6.4.1	Die Geschäfts- oder Missionsanalyse ist durchgeführt: Das zu lösende Problem, der Lösungsraum und die Randbedingungen des Vorhabens sind bestimmt.				
Woran man es festmacht: Produktvision, Problem Statement oder Business Case im Wiki, ergänzt um Marktbetrachtung oder Nutzerinterviews. Ein Epic mit Problembeschreibung und Zielsetzung im Issue-Tracker genügt für kleine Vorhaben.					
	Offen	In Arbeit	Erfüllt	Nicht anwendbar	
Nachweis / Begründung					

6.4.2	Die Definition der Stakeholder-Bedarfe und -Anforderungen ist durchgeführt: Interessengruppen sind identifiziert, ihre Bedarfe erhoben und in überprüfbare Stakeholder-Anforderungen überführt.	Dokument		
Woran man es festmacht: Stakeholder-Liste, Interviewnotizen, User Stories mit Akzeptanzkriterien. Die Akzeptanzkriterien im Ticket sind die überprüfbare Anforderungsfassung.				
	Offen	In Arbeit	Erfüllt	Nicht anwendbar
Nachweis / Begründung				

6.4.2.2 Die Ergebnisse liegen vor: Die Stakeholder-Anforderungen sind abgestimmt, priorisiert und ihre Rückverfolgbarkeit zum Bedarf ist hergestellt.

Woran man es festmacht: Priorisiertes Backlog mit Verknüpfung von Story zu Epic und Freigabekommentar des Auftraggebers im Ticket.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.4.3 Die Systemanforderungsdefinition ist durchgeführt: Aus den Stakeholder-Anforderungen sind die technischen Anforderungen an das Software-System abgeleitet, einschließlich Qualitätsmerkmalen und Randbedingungen.

[Dokument](#)

Woran man es festmacht: Anforderungsdokument oder strukturierte Issues mit funktionalen und nicht funktionalen Anforderungen (Leistung, Sicherheit, Barrierefreiheit). Verlinkung von Anforderung zu Testfall stellt die Prüfbarkeit sicher.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.4.3.2 Die Ergebnisse der Anforderungsdefinition liegen vor: Die Anforderungen sind eindeutig, widerspruchsfrei, überprüfbar und bis zu den Stakeholder-Anforderungen rückverfolgbar.

Woran man es festmacht: Rückverfolgbarkeitsmatrix, ersatzweise Issue-Verlinkung von Story zu Test und zu Commit. Automatisierte Akzeptanztests sind zugleich der Nachweis der Überprüfbarkeit.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.4.4 Die Systemarchitekturdefinition ist durchgeführt: Eine Architektur mit Sichten, Bausteinen und Schnittstellen ist festgelegt und gegen die Anforderungen bewertet.

[Dokument](#)

Woran man es festmacht: Architekturübersicht im Repository (etwa arc42 oder C4-Diagramme als Bild plus Text) sowie Architecture Decision Records mit den bewerteten Alternativen.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.4.4.2 Die Ergebnisse der Systemarchitekturdefinition liegen vor: Die Architektur ist gegenüber den Anforderungen bewertet, Alternativen sind abgewogen und die Wahl ist begründet.

Woran man es festmacht: ADR mit Abwägung von Optionen und Konsequenzen, dazu Ergebnisse eines Architektur-Reviews im Pull Request oder Protokoll.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.4.5 Die Entwurfsdefinition ist durchgeführt: Für die Architekturbausteine liegen Entwürfe vor, die eine Implementierung ermöglichen, einschließlich Datenmodell und Schnittstellenverträgen.

Woran man es festmacht: Schnittstellenspezifikationen (OpenAPI, GraphQL-Schema, Typdefinitionen), Datenbank-Migrationsskripte und Entwurfsnotizen im Pull Request. Der Code selbst mit Typen und Modulgrenzen ist ein zulässiger Entwurfsnachweis, wenn die Entscheidungen dokumentiert sind.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.4.5.2 Die Ergebnisse der Entwurfsdefinition liegen vor: Die Entwurfsmerkmale sind auf die Architektur rückverfolgbar und die Umsetzbarkeit ist bestätigt.

Woran man es festmacht: Verweis vom Pull Request auf ADR und Anforderung, Prototypen oder Spikes mit Ergebnisnotiz als Machbarkeitsnachweis.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.4.6	Die Systemanalyse ist durchgeführt: Technische Fragestellungen werden mit analytischen Mitteln untersucht, um Entscheidungen zu Architektur, Entwurf oder Risiken abzusichern.				
	Woran man es festmacht: Lastmessungen und Benchmark-Ergebnisse, Kosten- oder Trade-off-Analysen, Threat-Model-Notizen. Ergebnisse gehören als kurze Analysenotiz an den zugehörigen ADR.				
	Offen	In Arbeit	Erfüllt	Nicht anwendbar	
	Nachweis / Begründung				
6.4.7	Die Implementierung ist durchgeführt: Die Softwareelemente sind gemäss Entwurf und Codierregeln realisiert und die Ergebnisse sind unter Konfigurationskontrolle.				
	Woran man es festmacht: Git-Historie mit Commits, die auf Issues verweisen, Linter- und Formatierregeln als Merge-Gate, sowie erfolgreiche Builds in der CI-Pipeline.				
	Offen	In Arbeit	Erfüllt	Nicht anwendbar	
	Nachweis / Begründung				
6.4.7.3	Die Implementierungsaktivitäten sind durchgeführt: Codierstandards sind festgelegt, die Umsetzung ist geprüft und die Elemente sind zur Integration bereitgestellt.				
	Woran man es festmacht: Konfigurationsdateien der Linter und Formatter im Repository, bestandene Pull-Request-Reviews und veröffentlichte Build-Artefakte in der Registry.				
	Offen	In Arbeit	Erfüllt	Nicht anwendbar	
	Nachweis / Begründung				
6.4.8	Die Integration ist durchgeführt: Softwareelemente sind schrittweise zu einem funktionsfähigen System zusammengeführt und die Schnittstellen sind geprüft.				
	Woran man es festmacht: Continuous-Integration-Läufe bei jedem Merge, Integrationstests und Contract-Tests gegen die Schnittstellen, dazu die Build-Logs mit Zeitstempel.				
	Offen	In Arbeit	Erfüllt	Nicht anwendbar	
	Nachweis / Begründung				
6.4.8.2	Die Ergebnisse der Integration liegen vor: Die Integrationsstrategie ist festgelegt, die Schnittstellen sind konsistent und Integrationsfehler sind aufgezeichnet und behoben.				
	Woran man es festmacht: Pipeline-Definition im Repository als Integrationsstrategie, fehlgeschlagene und anschließend grüne Läufe als Fehlerbehebungsnachweis, Bug-Tickets mit Fix-Commit.				
	Offen	In Arbeit	Erfüllt	Nicht anwendbar	
	Nachweis / Begründung				
6.4.9	Die Verifikation ist durchgeführt: Zu jeder spezifizierten Anforderung liegt ein prüfbarer, nachvollziehbarer Beleg vor, dass die gebaute Software sie tatsächlich erfüllt.				Dokument
	Woran man es festmacht: Automatisierte Unit-, Integrations- und Regressionstests mit Testbericht aus der CI-Pipeline, statische Analyse, Code-Reviews. Der Testreport je Release-Tag ist der zentrale Beleg.				
	Offen	In Arbeit	Erfüllt	Nicht anwendbar	
	Nachweis / Begründung				
6.4.9.2	Die Ergebnisse der Verifikation liegen vor: Verifikationskriterien sind definiert, die Prüfungen sind durchgeführt und Abweichungen sind aufgezeichnet.				
	Woran man es festmacht: Definierte Abnahmeschwellen (etwa Mindest-Testabdeckung, keine kritischen Findings), archivierte Testberichte und Issues zu jeder Abweichung.				
	Offen	In Arbeit	Erfüllt	Nicht anwendbar	
	Nachweis / Begründung				

6.4.9.3 Die Verifikationsaktivitäten sind durchgeführt: Verifikationsumfang und -verfahren sind geplant, ausgeführt und die Ergebnisse sind ausgewertet.

Woran man es festmacht: Testplan oder Teststrategie als kurze Seite im Repository, Pipeline-Konfiguration mit den Prüfstufen, Auswertung in der Release-Freigabe.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.4.10 Der Überleitungsprozess ist durchgeführt: Das System ist in die Zielumgebung eingeführt, betriebsbereit übergeben und die Betriebsfähigkeit ist bestätigt.[Dokument](#)

Woran man es festmacht: Deployment-Runbook, Deployment-Logs der Pipeline, Smoke-Tests nach dem Rollout und ein dokumentierter Rollback-Plan. Release-Notes belegen die Übergabe an Nutzer und Betrieb.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.4.11 Die Validierung ist durchgeführt: Unter den vorgesehenen Einsatzbedingungen ist belegt, dass die Software den ursprünglichen Bedarf der Stakeholder tatsächlich deckt.[Dokument](#)

Woran man es festmacht: Abnahmetests mit dem Auftraggeber, Beta- oder Pilotphase mit Nutzerfeedback, Akzeptanzkriterien im Ticket als erfüllt markiert. Nutzungsdaten aus dem Betrieb stützen die Validierung zusätzlich.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.4.11.2 Die Ergebnisse der Validierung liegen vor: Validierungskriterien sind festgelegt, die Validierung ist in einer repräsentativen Umgebung erfolgt und die Abweichungen sind aufgezeichnet.

Woran man es festmacht: Protokoll der Abnahme mit Datum und Teilnehmern, Ergebnisse aus Staging- oder Pilotumgebung, Feedback-Tickets mit Erledigungsstand.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.4.12 Der Betriebsprozess ist etabliert: Das System wird bestimmungsgemäss betrieben, überwacht und Nutzer werden unterstützt.

Woran man es festmacht: Monitoring und Alerting mit definierten Schwellwerten, Incident-Tickets mit Reaktionszeit, Support-Postfach oder Ticketkanal. Ein Betriebshandbuch von wenigen Seiten reicht für kleine Betriebe.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.4.12.2 Die Ergebnisse des Betriebs liegen vor: Betriebsstrategie und Kennzahlen sind festgelegt, Störungen werden erfasst und die Leistung wird gegen die Erwartung bewertet.

Woran man es festmacht: Verfügbarkeits- und Fehlerkennzahlen aus dem Monitoring, Störungsprotokolle und Postmortems mit abgeleiteten Maßnahmen.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.4.13 Der Instandhaltungsprozess ist etabliert: Fehlerbehebungen, Anpassungen und Verbesserungen werden erfasst, bewertet, umgesetzt und ausgeliefert, ohne die Systemintegrität zu gefährden.[Dokument](#)

Woran man es festmacht: Bug- und Change-Tickets mit Priorität und Ursache, Hotfix-Branches, Regressionstests vor der Auslieferung und Release-Notes je Wartungsversion. Regelmäßige Updates der Abhängigkeiten sind über die Lockfile-Historie belegbar.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.4.13.2 Die Ergebnisse der Instandhaltung liegen vor: Änderungen sind analysiert, die Auswirkung ist bewertet und die Wirksamkeit der Behebung ist bestätigt.

Woran man es festmacht: Impact-Analyse im Ticket, Regressionstestlauf nach dem Fix, Bestätigung durch den Melder. Sicherheitsupdates sind über die Reports der Schwachstellen-Scans mit Behebungsdatum nachweisbar.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.4.14 Der Außerbetriebnahmeprozess ist etabliert: Die Stilllegung von Systemen oder Systemteilen ist geplant, Daten werden migriert oder gelöscht und Betroffene werden informiert.

[Dokument](#)

Woran man es festmacht: Stilllegungsplan mit Terminen, Nachweis der Datenmigration oder Löschung (inklusive Backups), Abschaltprotokoll der Infrastruktur und die Ankündigung an Nutzer oder Kunden.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.4.14.2 Die Ergebnisse der Außerbetriebnahme liegen vor: Aufbewahrungspflichten sind erfüllt, Umgebungen sind abgeschaltet und die Verantwortung ist beendet.

Woran man es festmacht: Archiv der aufbewahrungspflichtigen Daten mit Fristangabe, Ticket zur Ressourcen-Abschaltung mit Datum, Löschbestätigung des Hosters.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

A Konformität und Zuschnitt (Tailoring)

6

4.1 Der Geltungsbereich der Anwendung ist festgelegt (Abschnitt 4.1, Allgemeines): Es ist bestimmt, ob die Norm auf ein Projekt, eine Organisation oder eine Vereinbarung angewendet wird und welche Prozesse davon betroffen sind.

[Dokument](#)

Woran man es festmacht: Kurzer Geltungsbereichsvermerk in der Prozessdokumentation, der Projekt, Produkt und angewandte Prozessgruppen benennt. Eine Seite im Repository reicht aus.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

4.2 Die gewählte Konformitätsvariante ist erklärt: Entweder vollständige Konformität zu allen Ergebnissen der beanspruchten Prozesse oder vollständige Konformität zu allen Aufgaben der beanspruchten Prozesse.

[Dokument](#)

Woran man es festmacht: Eine Konformitätserklärung als eigenes Dokument, die die Variante benennt und je Prozess auf die Belege verweist. Es entsteht daraus kein Zertifikat, sondern eine Selbstdeklaration.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

4.3 Bei maßgeschneiderter Konformität ist der Zuschnitt dokumentiert: Es ist angegeben, welche Prozesse, Aufgaben oder Ergebnisse entfallen, verändert oder ergänzt wurden.

[Dokument](#)

Woran man es festmacht: Zuschnitt-Tabelle mit Spalten Prozess, Entscheidung (übernommen, angepasst, entfallen), Begründung und Risiko. Genau dieses Dokument ist für kleine Teams der entscheidende Nachweis.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

5 Die Schlüsselkonzepte sind auf die Organisation angewendet: Der Bezug zwischen Lebenszyklusmodell, Projektstruktur und den eingesetzten Prozessen ist verständlich dargestellt.

Woran man es festmacht: Eine Übersichtsseite, die das gewählte Vorgehensmodell (etwa iterativ, kontinuierliche Auslieferung) den 30 Prozessen zuordnet. Ein Diagramm plus Tabelle genügt.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

A.1 Der Zuschnitt ist als bewusste Entscheidung geführt: Zuschnitt ist zulässig, er wird jedoch nur vorgenommen, wenn die Randbedingungen des Vorhabens es rechtfertigen, und er wird begründet.[Dokument](#)

Woran man es festmacht: Begründungsspalte in der Zuschnitt-Tabelle, die auf Projektgröße, Risikoklasse, Vertragslage oder gesetzliche Vorgaben verweist. Pauschale Streichungen ohne Begründung sind kein zulässiger Zuschnitt.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

A.2 Der Zuschnittprozess ist durchgeführt: Einflussfaktoren sind ermittelt, Interessengruppen sind einbezogen, Zuschnittentscheidungen sind getroffen und dokumentiert.[Dokument](#)

Woran man es festmacht: Protokoll oder Ticket zur Zuschnitt-Entscheidung mit den beteiligten Rollen, dazu die versionierte Zuschnitt-Tabelle in Git. Auswirkungen auf Risiken werden im Risikoregister vermerkt.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

ISO/IEC 25010:2023 Systeme und Software-Engineering, SQuaRE, Produktqualitätsmodell

Qualitätsmodell für Softwareprodukte

40 Anforderungen. Für diese Norm gibt es kein akkreditiertes Zertifikat. Stand 07/2026.

ISO/IEC 25010:2023 ist ein Qualitätsmodell, keine Anforderungsnorm mit verbindlichen Klauseln, und deshalb nicht akkreditiert zertifizierbar. Das Modell dient als Raster, um Qualitätsanforderungen vollständig zu formulieren; gemessen wird nach ISO/IEC 25020 und bewertet nach ISO/IEC 25040, Ergebnis ist ein nachvollziehbarer Bewertungsbericht, kein Zertifikat. Werbung mit einer angeblichen ISO 25010-Zertifizierung ist unzulässig.

Betrieb

Datum

Bearbeiter

1 Funktionale Eignung (Functional Suitability)

3

1.1 Der Funktionsumfang deckt alle festgelegten Aufgaben und Nutzerziele ab, für die das Produkt eingesetzt wird (Functional Completeness). Lücken zwischen dokumentierten Anforderungen und umgesetzten Funktionen sind bekannt und bewertet.

Woran man es festmacht: Nachweis über eine Rückverfolgbarkeitsmatrix von Anforderung zu Testfall, messbar als Abdeckungsgrad der Anforderungen in Prozent. In kleinen Teams reicht ein Issue-Tracker mit Label je Anforderung plus ein Abnahmetest je Nutzerziel.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

1.2 Das Produkt liefert für die definierten Eingaben Ergebnisse mit der geforderten Genauigkeit (Functional Correctness). Abweichungen und Toleranzen sind festgelegt.

Woran man es festmacht: Messbare Größe ist die Fehlerdichte in Defekten je tausend Codezeilen oder je Funktion, belegt über Unit- und Integrationstests mit Soll-Ist-Vergleich sowie Bug-Statistik aus dem Tracker. Für Rechenlogik zusätzlich Referenzdatensätze mit erwarteten Ergebnissen hinterlegen.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

1.3 Die bereitgestellten Funktionen unterstützen die Aufgaben der Nutzer ohne überflüssige Schritte und passen zum Einsatzzweck (Functional Appropriateness).

Woran man es festmacht: Beleg über Task-Erfolgsquote und Anzahl Schritte je Nutzeraufgabe aus einem kleinen Usability-Test mit fünf Personen, protokolliert je Aufgabe. Ergänzend Nutzungsdaten oder Support-Anfragen, die Umwege sichtbar machen.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

2 Leistungseffizienz (Performance Efficiency)

3

2.1 Antwort-, Verarbeitungs- und Durchsatzzeiten liegen unter festgelegten Schwellwerten (Time Behaviour). Die Schwellwerte sind für die typischen Nutzungsszenarien definiert.

Woran man es festmacht: Messbar über Ladezeit und Interaktionslatenz, erhoben mit Lighthouse oder Web Vitals (LCP, INP) sowie Server-Antwortzeiten aus dem Log als 95. Perzentil. Schwellwerte je Szenario in einem kurzen Performance-Budget festhalten.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

2.2 Der Verbrauch an Prozessorzeit, Arbeitsspeicher, Speicherplatz und Netzwerkbandbreite bleibt im festgelegten Rahmen (Resource Utilization).

Woran man es festmacht: Beleg über Monitoring-Werte für CPU, RAM und Datenvolumen unter Last sowie die Größe des ausgelieferten Bundles aus dem Build-Report. Für kleine Betriebe genügt ein Messprotokoll aus einem Lasttest mit den Grenzwerten der Zielhardware.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

2.3 Die maximale Anzahl gleichzeitiger Nutzer, Transaktionen oder Datensätze ist bekannt und wird eingehalten (Capacity).

Woran man es festmacht: Nachweis über einen Lasttest (zum Beispiel k6 oder JMeter) mit dokumentierter Sättigungsgrenze und dem Punkt, ab dem Fehlerraten steigen. Ergebnis als kurzer Kapazitätsbericht mit Ist-Grenze und geplantem Bedarf.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

3 Kompatibilität (Compatibility)

2

3.1 Das Produkt arbeitet in einer gemeinsam genutzten Umgebung, ohne andere Produkte zu beeinträchtigen (Co-existence).

Woran man es festmacht: Beleg über einen Testlauf in der Zielumgebung mit parallel laufender Software, protokolliert werden Portkonflikte, Speicher- und Dateizugriffe. Containerisierung mit definierten Ressourcengrenzen ist der praktikable Weg für kleine Teams.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

3.2 Der Datenaustausch mit anderen Systemen funktioniert über festgelegte Schnittstellen und Formate (Interoperability).

Woran man es festmacht: Nachweis über eine versionierte Schnittstellenbeschreibung (OpenAPI oder Schema-Datei) und Vertragstests gegen diese Beschreibung. Import- und Exportformate mit echten Beispieldateien gegenprüfen und die Ergebnisse ablegen.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

4 Interaktionsfähigkeit (Interaction Capability)

8

4.1 Nutzer erkennen ohne fremde Hilfe, ob das Produkt zu ihrem Bedarf passt (Appropriateness Recognizability). Zweck und Nutzen sind aus der Oberfläche und den Einstiegstexten ersichtlich.

Woran man es festmacht: Beleg über einen kurzen Erst-Eindruck-Test: Testpersonen beschreiben nach 30 Sekunden den Zweck des Produkts, Trefferquote wird protokolliert. Ergänzend Startseite und Onboarding-Screens als Screenshot-Nachweis.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

4.2 Neue Nutzer erreichen die Kernaufgaben in einer festgelegten Zeit und mit begrenzter Fehlerzahl (Learnability).

Woran man es festmacht: Messbar als Zeit bis zur ersten erfolgreichen Kernaufgabe und Fehlversuche je Aufgabe, erhoben in einem Test mit fünf neuen Nutzern. Onboarding, Tour oder Demodaten als unterstützende Maßnahme dokumentieren.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

4.3 Die Bedienung ist effizient und steuerbar, wiederkehrende Aufgaben lassen sich ohne Umwege ausführen (Operability).

Woran man es festmacht: Nachweis über Klickpfad-Analyse je Standardaufgabe (Soll-Schritte gegen Ist-Schritte) und Tastaturbedienbarkeit ohne Maus. Protokoll aus dem Usability-Test mit Abbruchquote je Aufgabe.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

4.4 Das Produkt verhindert Bedienfehler oder macht sie umkehrbar (User Error Protection). Kritische Aktionen sind bestätigungspflichtig oder rückgängig zu machen.

Woran man es festmacht: Beleg über Testfälle mit ungültigen Eingaben und Grenzwerten sowie Nachweis von Validierung, Bestätigungsdialog und Undo-Funktion. Fehleingaben aus dem Produktivlog auswerten und die Häufigkeit als Kennzahl führen.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

4.5 Die Interaktion ist ansprechend gestaltet und hält die Nutzer motiviert bei der Aufgabe (User Engagement).

Woran man es festmacht: Messbar über Wiederkehrquote, Sitzungsdauer und Abschlussquote begonnener Vorgänge aus einer datenschutzfreundlichen Nutzungsstatistik. Ergänzend eine kurze Zufriedenheitsbefragung mit Skalenwert, das genügt auch für kleine Betriebe.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

4.6 Das Produkt ist für Menschen mit unterschiedlichen Fähigkeiten, Sprachen und Kulturen nutzbar (Inclusivity).

Woran man es festmacht: Nachweis über eine Prüfung nach WCAG 2.2 Stufe AA: Kontrastwerte, Tastaturnavigation, Screenreader-Bezeichner, geprüft mit axe oder Lighthouse-Accessibility und einer manuellen Stichprobe. Sprachumfang und Lokalisierung über die Vollständigkeit der Übersetzungsdateien belegen.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

4.7 Nutzer erhalten bei Bedarf Hilfe, die zur Aufgabe passt, auch bei Fehlern oder Sonderfällen (User Assistance).

Woran man es festmacht: Beleg über Handbuch, kontextsensitive Hilfetexte und verständliche Fehlermeldungen mit Handlungsvorschlag. Kennzahl: Anteil der Support-Anfragen, die durch vorhandene Hilfe hätten gelöst werden können, aus dem Ticketsystem.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

4.8 Die Oberfläche erklärt sich selbst: Zustände, Auswirkungen von Aktionen und benötigte Eingaben sind ohne Zusatzwissen erkennbar (Self-descriptiveness).

Woran man es festmacht: Nachweis über einen Review der Oberfläche gegen eine Checkliste: aussagekräftige Beschriftungen, sichtbarer Systemstatus, Fortschritts- und Ladeanzeigen, Einheiten an Zahlenfeldern. Beobachtungsprotokoll aus dem Nutzertest ergänzt den Beleg.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

5 Zuverlässigkeit (Reliability)**4****5.1 Das Produkt arbeitet im normalen Betrieb ohne Ausfälle durch Fehler in der Software (Faultlessness).**

Woran man es festmacht: Messbar über die Fehlerdichte und die mittlere Zeit zwischen Ausfällen (MTBF) aus Produktivlogs sowie über die Testabdeckung als vorbeugende Kennzahl. Ein Fehler-Monitoring wie Sentry liefert die Rohdaten mit geringem Aufwand.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

5.2 Das Produkt ist in den vereinbarten Zeitfenstern erreichbar und nutzbar (Availability). Der Zielwert der Verfügbarkeit ist festgelegt.

Woran man es festmacht: Nachweis über Uptime-Messung eines externen Monitorings mit Verfügbarkeit in Prozent je Monat, verglichen mit dem vereinbarten Zielwert. Ausfälle mit Dauer und Ursache in einem einfachen Störungslog führen.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

5.3 Bei Hardware- oder Softwarefehlern bleibt das Produkt in einem definierten, nutzbaren Zustand (Fault Tolerance).

Woran man es festmacht: Beleg über Testfälle mit simuliertem Ausfall (Netz weg, Datenbank weg, Fremdsystem antwortet nicht) und dem dokumentierten Verhalten, etwa Offline-Modus, Wiederholversuch oder Ersatzanzeige. Ergebnisse als Testprotokoll ablegen.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

5.4 Nach einem Ausfall lassen sich Daten und Betriebszustand innerhalb der festgelegten Zeit wiederherstellen (Recoverability).

Woran man es festmacht: Messbare Größen sind Wiederherstellungszeit (RTO) und maximal tolerierter Datenverlust (RPO), belegt durch einen tatsächlich durchgeführten Restore-Test mit Datum und Dauer. Für kleine Betriebe genügt ein halbjährlicher Rücksicherungstest mit kurzem Protokoll.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6 Sicherheit (Security)**6****6.1 Daten sind nur für berechtigte Personen und Systeme zugänglich (Confidentiality). Zugriffsrechte sind nach Bedarf vergeben.**[Dokument](#)

Woran man es festmacht: Nachweis über ein Rollen- und Rechtekonzept, Verschlüsselung bei Übertragung und Speicherung sowie Tests, die den Zugriff ohne Berechtigung ablehnen. Ergänzend ein Header- und TLS-Check der Anwendung.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.2 Unbefugte Änderungen an Programmen und Daten werden verhindert oder erkannt (Integrity).

Woran man es festmacht: Beleg über Prüfsummen oder Signaturen der Artefakte, Eingabevalidierung und einen Abhängigkeits-Scan (npm audit, Dependabot) mit Nachweis behobener Befunde. Datenbank-Constraints und Migrationsprotokolle stützen den Punkt zusätzlich.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.3 Durchgeführte Aktionen lassen sich später nicht abstreiten (Non-repudiation).[Dokument](#)

Woran man es festmacht: Nachweis über manipulationsgeschützte Protokolle mit Zeitstempel und Urheber, bei rechtlich relevanten Vorgängen über Signaturen. Ein zentrales, schreibgeschütztes Logziel reicht für kleine Betriebe aus.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.4 Handlungen im System sind eindeutig einer verantwortlichen Person oder einem Dienst zuzuordnen (Accountability).[Dokument](#)

Woran man es festmacht: Beleg über Audit-Logs mit Benutzerkennung je sicherheitsrelevanter Aktion, verbunden mit dem Verbot geteilter Konten. Stichprobe: einen Vorgang im Log vollständig zurückverfolgen und das Ergebnis dokumentieren.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.5 Die Identität von Nutzern und Systemen wird zuverlässig überprüft (Authenticity).

Woran man es festmacht: Nachweis über das Authentifizierungsverfahren, Passwortrichtlinie und Zwei-Faktor-Verfahren für administrative Zugänge sowie über die Prüfung von Tokens und Zertifikaten. Testfälle mit abgelaufenen und gefälschten Tokens belegen die Wirksamkeit.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

6.6 Das Produkt hält Angriffen stand und bleibt dabei funktionsfähig (Resistance).

Woran man es festmacht: Beleg über einen Schwachstellen-Scan, statische Codeanalyse (SAST) und, wo verhältnismäßig, einen Penetrationstest mit Befundliste und Behebungsstand. Für kleine Betriebe ist ein automatisierter Scan in der Pipeline plus jährlicher externer Kurztest der praktikable Weg.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

7 Wartbarkeit (Maintainability)					5
7.1	Das Produkt besteht aus abgegrenzten Bausteinen, deren Änderung möglichst wenig andere Bausteine betrifft (Modularity).				
Woran man es festmacht: Messbar über Kopplung und Kohäsion aus einer statischen Analyse sowie über Abhängigkeitsgraphen (madge, dependency-cruiser) mit Nachweis, dass keine Zyklen bestehen. Modulgrenzen in einer knappen Architekturskizze festhalten.					
	Offen	In Arbeit	Erfüllt	Nicht anwendbar	
Nachweis / Begründung					
7.2	Bausteine sind so gestaltet, dass sie in mehreren Zusammenhängen wiederverwendet werden können (Reusability).				
Woran man es festmacht: Beleg über gemeinsam genutzte Bibliotheken oder Komponenten mit eigener Versionierung und dokumentierter Schnittstelle. Kennzahl: Anteil wiederverwendeter Komponenten und Grad an Codeduplizierung aus der statischen Analyse.					
	Offen	In Arbeit	Erfüllt	Nicht anwendbar	
Nachweis / Begründung					
7.3	Ursachen von Fehlern und Auswirkungen geplanter Änderungen lassen sich effizient bestimmen (Analysability).				
Woran man es festmacht: Nachweis über strukturiertes Logging, Fehler-Monitoring und lesbare Codemetriken wie zyklomatische Komplexität aus SonarQube oder ESLint. Kennzahl: mittlere Zeit bis zur Ursachenfindung aus dem Ticketsystem.					
	Offen	In Arbeit	Erfüllt	Nicht anwendbar	
Nachweis / Begründung					
7.4	Änderungen sind ohne Einführung neuer Fehler und ohne Qualitätsverlust möglich (Modifiability).				
Woran man es festmacht: Beleg über Regressionstests in der Pipeline und die Quote der Fehler, die nach Änderungen entstehen (Change Failure Rate). Codereview-Pflicht und Versionskontrolle mit nachvollziehbaren Commits stützen den Punkt.					
	Offen	In Arbeit	Erfüllt	Nicht anwendbar	
Nachweis / Begründung					
7.5	Für das Produkt und seine Bausteine lassen sich wirksame Prüfkriterien und Tests aufstellen und ausführen (Testability).				
Woran man es festmacht: Messbar über Testabdeckung in Prozent (Zeilen und Zweige) aus dem Coverage-Report sowie über die Laufzeit der Testsuite. Automatisierter Testlauf bei jedem Commit ist der Beleg der Wirksamkeit.					
	Offen	In Arbeit	Erfüllt	Nicht anwendbar	
Nachweis / Begründung					
8 Flexibilität (Flexibility)					4
8.1	Das Produkt lässt sich an andere oder veränderte Umgebungen anpassen, ohne dass es umgebaut werden muss (Adaptability).				
Woran man es festmacht: Nachweis über Konfiguration außerhalb des Codes (Umgebungsvariablen, Konfigurationsdateien) und über Tests auf mehreren Zielumgebungen, Browsern oder Betriebssystemen in einer Testmatrix. Ergebnis als Kompatibilitätstabelle ablegen.					
	Offen	In Arbeit	Erfüllt	Nicht anwendbar	
Nachweis / Begründung					
8.2	Das Produkt bewältigt wachsende oder schrumpfende Mengen an Nutzern und Daten, ohne dass die Leistung unzulässig einbricht (Scalability).				
Woran man es festmacht: Beleg über Lasttests mit steigender Nutzerzahl und dem Verlauf von Antwortzeit und Fehlerrate, ergänzt um die Skalierungsstrategie (horizontal oder vertikal). Kennzahl: Antwortzeit bei ein-, zwei- und fünffacher Last.					
	Offen	In Arbeit	Erfüllt	Nicht anwendbar	
Nachweis / Begründung					

8.3 Das Produkt lässt sich in der Zielumgebung zuverlässig installieren, aktualisieren und wieder entfernen (Installability).

Woran man es festmacht: Nachweis über eine reproduzierbare Installationsanleitung und einen automatisierten Installationstest auf einem frischen System, gemessen als Erfolgsquote und benötigte Zeit. Deinstallation und Rollback ebenfalls einmal durchspielen und protokollieren.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

8.4 Das Produkt kann ein anderes Produkt mit gleichem Zweck ersetzen, Daten und Abläufe bleiben dabei erhalten (Replaceability).

Woran man es festmacht: Beleg über Import- und Exportfunktionen in offenen Formaten sowie einen Migrationstest mit echten Altdaten, geprüft auf Vollständigkeit und Datentreue. Datenmodell und Migrationsskripte versioniert ablegen.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

9 Gefahrenfreiheit (Safety)

5

9.1 Der sichere Betriebsbereich ist festgelegt, und das Produkt erkennt und behandelt das Verlassen dieses Bereichs (Operational Constraint).

Woran man es festmacht: Nachweis über dokumentierte Betriebsgrenzen (Grenzwerte, zulässige Zustände) und Tests, die das Verhalten außerhalb dieser Grenzen prüfen. Ist die Software ohne Gefahrenpotenzial für Menschen oder Sachwerte, gilt der Punkt typisch als nicht anwendbar; diese Einstufung muss aber begründet und dokumentiert werden, nicht stillschweigend weggelassen.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

9.2 Gefährdungen, die vom Produkt oder seinem Einsatz ausgehen können, sind systematisch ermittelt und bewertet (Risk Identification).

[Dokument](#)

Woran man es festmacht: Beleg über eine Gefährdungs- und Risikobeurteilung mit Eintrittswahrscheinlichkeit, Schwere und Maßnahme je Gefährdung, praktikabel als Tabelle im Qualitätsordner. Nicht anwendbar nur mit schriftlicher Begründung, dass kein Schadenspotenzial für Menschen oder Sachwerte besteht.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

9.3 Bei Störungen geht das Produkt in einen sicheren Zustand über (Fail Safe).

Woran man es festmacht: Nachweis über Testfälle mit erzwungenem Fehler und dem dokumentierten Übergang in den sicheren Zustand, etwa Abschalten, Sperren oder Rückfall auf einen geprüften Standardwert. Bei Software ohne Gefahrenpotenzial ist der Punkt typisch nicht anwendbar, die Begründung gehört trotzdem in die Bewertung.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

9.4 Das Produkt warnt rechtzeitig vor drohenden gefährlichen Zuständen, sodass eine Reaktion möglich ist (Hazard Warning).

Woran man es festmacht: Beleg über Alarm- und Warnkonzept mit Schwellwerten, Vorwarnzeit und Empfängern sowie ein Testprotokoll ausgelöster Warnungen. Ohne Gefahrenpotenzial ist der Punkt typisch nicht anwendbar; die Begründung ist zu dokumentieren.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung

9.5 Beim Zusammenspiel mit anderen Systemen entstehen keine neuen Gefährdungen (Safe Integration).

Woran man es festmacht: Nachweis über Schnittstellen-Risikobetrachtung und Integrationstests mit den angeschlossenen Systemen, inklusive Verhalten bei Ausfall oder Fehldaten des Partnersystems. Bei Software ohne Gefahrenpotenzial gilt der Punkt typisch als nicht anwendbar, das ist begründet festzuhalten.

Offen In Arbeit Erfüllt Nicht anwendbar

Nachweis / Begründung