

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.

OffenIn ArbeitErfülltNicht 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.

OffenIn ArbeitErfülltNicht 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.

OffenIn ArbeitErfülltNicht 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.

OffenIn ArbeitErfülltNicht 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.

OffenIn ArbeitErfülltNicht 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 berechnigte 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 Berechnigung 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 Benutzererkennung 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.					
OffenIn ArbeitErfülltNicht 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.					
OffenIn ArbeitErfülltNicht 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.					
OffenIn ArbeitErfülltNicht 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.					
OffenIn ArbeitErfülltNicht 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.					
OffenIn ArbeitErfülltNicht 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.					
OffenIn ArbeitErfülltNicht 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.					
OffenIn ArbeitErfülltNicht 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