

Qualitätsbegriff nach ISO 9000 (Kernidee)



Prüfungswissen



Prüfungsfallen & Tipps

- Qualität = Grad, in dem Anforderungen/Erwartungen erfüllt werden
- Kundenzufriedenheit & Prozessfähigkeit im Fokus
- Qualitätsmerkmale: definierte Anforderungen & messbare Kriterien

- Qualität bedeutet nicht „edel“ oder „teuer“, sondern „Fit for Purpose“
- Ohne klare Anforderungen kann Qualität nicht bewertet werden

Merker: Keine Anforderung, keine Qualität.

ISO 9000: Begriffe, die in Prüfungen auftauchen



Prüfungswissen



Prüfungsfallen & Tipps

- Prozess: zusammenhängende Tätigkeiten mit Input/Output
- Dokumentierte Information: Nachweis & Lenkung (Versionen, Freigaben)
- Korrektur vs. Korrekturmaßnahme: Fehler beheben vs. Ursache verhindern
- Korrekturmaßnahme wird oft mit „Fix“ verwechselt – Ursache zählt!
- Dokumentation soll steuern, nicht nur abgeheftet werden

Beispiel: Bugfix = Korrektur; Root-Cause + Test ergänzen = Korrekturmaßnahme

Betriebliche QM-Systeme (Wozu?)



Prüfungswissen



Prüfungsfallen & Tipps

- QM-System = Regeln, Prozesse & Verantwortungen für planbare Qualität
- Ziele: weniger Fehler, stabile Abläufe, bessere Nachvollziehbarkeit, Compliance
- Bausteine: Prozesse, Rollen, Schulungen, Kennzahlen, Dokumentation, Reviews

- QM „auf Papier“ ohne gelebte Praxis führt zu Bürokratie
- Kennzahlen ohne Maßnahmen sind wertlos

Merker: QM = System, nicht Einzelaktion.

QM-Systeme im IT-Alltag (konkret)



Prüfungswissen

- Prozessdokumentation: Tickets, Changes, Releases, Abnahmen
- Qualitätsnachweise: Testprotokolle, Review-Logs, Checklisten
- Kontinuierliche Verbesserung: Retros, Lessons Learned, PDCA



Prüfungsfallen & Tipps

- Ohne Versionierung/Freigaben ist Doku nicht verlässlich
- “Wir testen am Ende” ist ein Qualitätsrisiko

Beispiel: Change nur mit Risikoanalyse, Test & Rollback

QS-Normen (Was ist das überhaupt?)



Prüfungswissen



Prüfungsfallen & Tipps

- Norm = anerkannter Standard (Begriffe, Anforderungen, Vorgehen)
- Nutzen: Vergleichbarkeit, Mindestniveau, Audit-/Zertifizierbarkeit
- Beispiele: ISO 9001 (QM-Anforderungen), ISO 25010 (Softwarequalität)

- Norm \neq Gesetz – kann vertraglich verpflichtend werden
- „Zertifiziert“ heißt nicht automatisch fehlerfrei

Merker: Norm = Referenzrahmen.

Audit (Definition & Ziel)



Prüfungswissen



Prüfungsfallen & Tipps

- Audit = systematische, unabhängige Prüfung gegen Kriterien (Norm/Prozess)
- Auditarten: intern (Selbstprüfung), extern (Kunde/Zertifizierer)
- Ergebnis: Abweichungen, Beobachtungen & Verbesserungen
- Audit prüft Nachweise, nicht Gefühle
- „Wir machen das so“ ohne Doku/Logs zählt selten

Beispiel: Nachweis = Ticket, Testprotokoll, Freigabe

Zertifizierung (Was passiert da?)



Prüfungswissen



Prüfungsfallen & Tipps

- Zertifizierung = Bestätigung durch unabhängige Stelle, dass Anforderungen erfüllt sind
- Ablauf: Audit planen → prüfen → Abweichungen beheben → Zertifikat
- Nutzen: Vertrauen, Marktanforderung, Teilnahme an Ausschreibungen

- Zertifikat ist zeitlich begrenzt (Überwachungsaudits)
- Papier ist nicht gleich gelebte Qualität

Merker: Zertifizierung = Audit + bestätigtes QM.

Qualitätsmanagement: Maßnahmen zur Verbesserung (Überblick)



Prüfungswissen



Prüfungsfallen & Tipps

- Prozessqualität: klare Abläufe, Standards, Checklisten, Übergaben
- Arbeitsqualität: Schulungen, 4-Augen-Prinzip, klare Verantwortungen
- Produktqualität: Tests, Reviews, Definition of Done, Dokumentation
- Dienstleistungsqualität: SLAs, Supportprozesse, Feedback, Incident-Analyse

- Maßnahmen müssen messbar sein – sonst bleibt es Aktionismus
- Nur ein Tool einführen ohne Prozess ist ein Scheinerfolg

Beispiel: Code Review Pflicht + Mindesttestabdeckung

Qualitätsplanung (Planen statt hoffen)



Prüfungswissen



Prüfungsfallen & Tipps

- Qualitätsplanung definiert Standards, Tests, Reviews, Metriken & Verantwortliche
- Inhalte: Kriterien, Prüfmethoden, Tools, Zeitpunkte, Abnahmekriterien
- Ergebnis: Qualitätsplan – wer prüft was, wann und womit

- „QA macht das am Ende“ ist kein Plan
- Abnahmekriterien müssen früh feststehen

Merker: Quality is designed in.

Qualitätsziele (Ist ermitteln → Ziel festlegen)



Prüfungswissen



Prüfungsfallen & Tipps

- Ist-Zustand: Messung & Analyse (Fehlerquote, Durchlaufzeit, Reklamationen)
- Ziel-Zustand: konkret, messbar, terminiert (SMART)
- Beispiele: weniger Incidents, kürzere Bearbeitungszeit, weniger Reopens
- Ziele ohne Baseline sind Fantasie
- Zu viele Ziele führen zu fehlender Priorität

Beispiel: Reopen-Rate von 12 % auf < 5 % bis Q3

Qualitätslenkung (Steuern im Betrieb)



Prüfungswissen



Prüungsfallen & Tipps

- Lenkung = laufend überwachen & korrigieren (Soll-Ist-Vergleich)
- Mittel: Reviews, Test-Gates, Freigaben, Change-Prozess, Monitoring
- Eskalation erfolgt bei Überschreitung von Grenzwerten

- Ohne klare Grenzwerte kann nicht gesteuert werden
- „Ausnahmen“ werden schnell zur Regel → der Prozess bricht

Merker: Lenkung = im Prozess eingreifen, bevor's knallt.

PDCA (Plan–Do–Check–Act) als Verbesserungsmaschine



Prüfungswissen



Prüfungsfallen & Tipps

- Plan: Ziel/Änderung planen und Messmethode festlegen
 - Do: Umsetzung starten (klein beginnen, z.B. Pilot)
 - Check: Ergebnisse messen & auswerten – hat es geholfen?
 - Act: Standardisieren oder nachbessern
- Check wird oft vergessen → keine echte Verbesserung
 - Act fehlt → Lernen versandet und es bleibt beim Aktionismus

Beispiel: Neue Review-Checkliste → Pilot → Metrik → Standard

Softwarequalität: Überblick (6 Kriterien)



Prüfungswissen



Prüfungsfallen & Tipps

- Funktionalität: tut, was gefordert ist
- Zuverlässigkeit: stabil, wenige Ausfälle
- Benutzbarkeit: leicht bedienbar & verständlich
- Effizienz: angemessene Ressourcen/Performance
- Änderbarkeit: wartbar & erweiterbar
- Übertragbarkeit: portierbar auf andere Systeme/Umgebungen

- Qualität ist mehr als Bugs: umfasst Usability, Wartbarkeit & Portabilität
- Kriterien können sich widersprechen (z.B. Effizienz vs. Änderbarkeit)

Merker: 6er-Set als AP1-Standardliste merken.

Funktionalität & Zuverlässigkeit (konkretisieren)



Prüfungswissen



Prüfungsfallen & Tipps

- Funktionalität: korrekte Ergebnisse, passende Funktionen, Sicherheit als Qualitätsaspekt
- Zuverlässigkeit: Verfügbarkeit, Fehlertoleranz & Wiederherstellbarkeit
- Nachweise: Tests, Monitoring, Incident-Statistik, SLAs

- „Läuft bei mir“ ohne Tests/Monitoring ist kein Nachweis
- Zuverlässigkeit erfordert auch Betrieb: Backup & Alarmierung

Beispiel: 99,5 % Verfügbarkeit + Alarmierung + Restore-Test

Benutzbarkeit & Effizienz (konkretisieren)



Prüfungswissen



Prüfungsfallen & Tipps

- Benutzbarkeit: Verständlichkeit, Lernbarkeit, Barrierefreiheit & Fehlervermeidung
- Effizienz: Antwortzeiten, Durchsatz & Ressourcenverbrauch (CPU/RAM/IO)
- Methoden: Usability-Tests, Performance-Tests & Profiling
- Effizienz optimieren ohne Messung ist blindes Raten
- Usability-Feedback ignorieren → Supportkosten steigen

Merker: Schnell + einfach spart Betriebskosten.

Änderbarkeit & Übertragbarkeit (konkretisieren)



Prüfungswissen



Prüfungsfallen & Tipps

- Änderbarkeit: modularer Code, klare Schnittstellen, Tests & Dokumentation
- Übertragbarkeit: läuft in anderer Umgebung (OS, DB, Cloud) & geringe Abhängigkeiten
- Praktiken: Clean Code, Continuous Integration, Containerisierung, Konfiguration statt Hardcoding

- Hardcoding oder fehlende Tests machen jede Änderung zum Risiko
- Vendor-Lock-in ohne Exit-Plan wird schnell teuer

Beispiel: Konfiguration per Environment-Variable + automatisierte Tests erhöht Portabilität.