

Maßnahmen des Qualitätsmanagements

Qualitätsmanagement zielt darauf ab, die Qualität von Prozessen, Arbeitsergebnissen, Produkten und Dienstleistungen systematisch zu sichern und kontinuierlich zu verbessern. Der Kerngedanke: Messen → bewerten → verbessern – nicht nach Gefühl, sondern auf Basis definierter Kriterien und messbarer Metriken.

Vier zentrale Qualitätsbereiche

Prozessqualität

Wie effizient läuft der Ablauf?
Messkriterien: Durchlaufzeit, Fehlerquote bei Übergaben, Wartezeiten zwischen Prozessschritten

Arbeitsqualität

Wie sauber wird gearbeitet?
Messkriterien: Einhaltung von Standards, Nutzung von Checklisten, 4-Augen-Prinzip bei kritischen Arbeiten

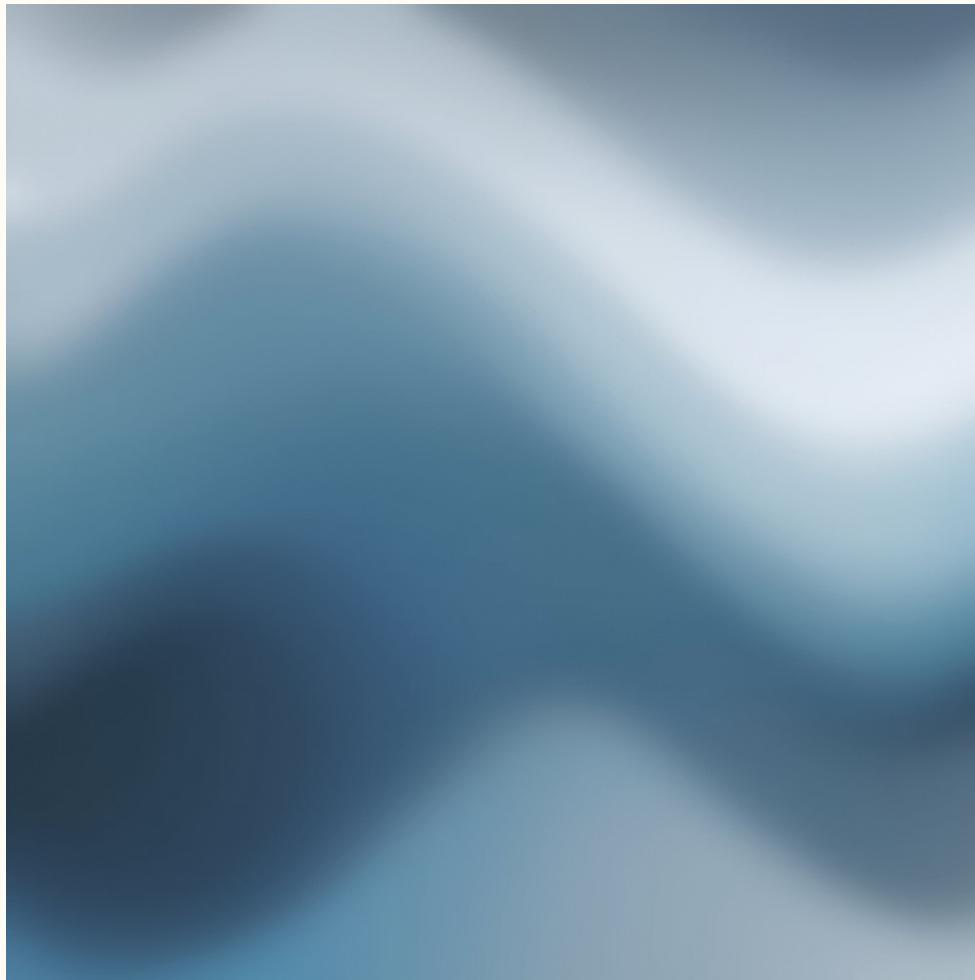
Produktqualität

Erfüllt das Ergebnis die Anforderungen?
Messkriterien: Funktionserfüllung, Performance-Werte, Sicherheitsstandards

Dienstleistungsqualität

Passt Service und Support?
Messkriterien: Reaktionszeit, Lösungsquote, Einhaltung von Service Level Agreements

Qualitätsbereiche in der Praxis



Praxisreflexion: Die vier Qualitätsbereiche

- *Wie würden Sie Prozess-, Arbeits-, Produkt- und Dienstleistungsqualität in Ihrem aktuellen IT-Projekt konkret messen?*
- *Woran erkennen Sie in der Praxis, ob ein Problem eher im Prozess oder im Produkt liegt?*
- *Welche QM-Maßnahmen können Sie ohne zusätzliche Tools sofort im Team einführen?*
- *Warum ist Qualitätsverbesserung ohne Messung fast immer Selbstbetrug?*

Qualitätsplanung: Standards vorher festlegen

Vorher festlegen, was „gut“ bedeutet

Qualitätsplanung definiert präzise Anforderungen, Messkriterien, Prüfpunkte und Verantwortlichkeiten, bevor die Umsetzung beginnt. Ohne klare Planung fehlt die Grundlage für objektive Bewertung.

Eine brauchbare Qualitätsplanung muss mindestens drei Elemente enthalten: messbare Qualitätskriterien, eine definierte Prüfstrategie und festgelegte Standards für die Umsetzung.

Qualitätskriterien

Konkrete Messwerte: „Login < 2s“, „0 kritische Bugs bei Abnahme“

Prüf-/Teststrategie

*Wer testet was wann?
Testarten, Abnahmekriterien, Testumgebungen*

Standards & Vorgehen

Coding-Standards, Definition of Done, Review-Regeln

Dokumentation

*Nachweisbare Ergebnisse:
Testprotokolle, Abnahmeprotokolle*

Risiken unzureichender Qualitätsplanung

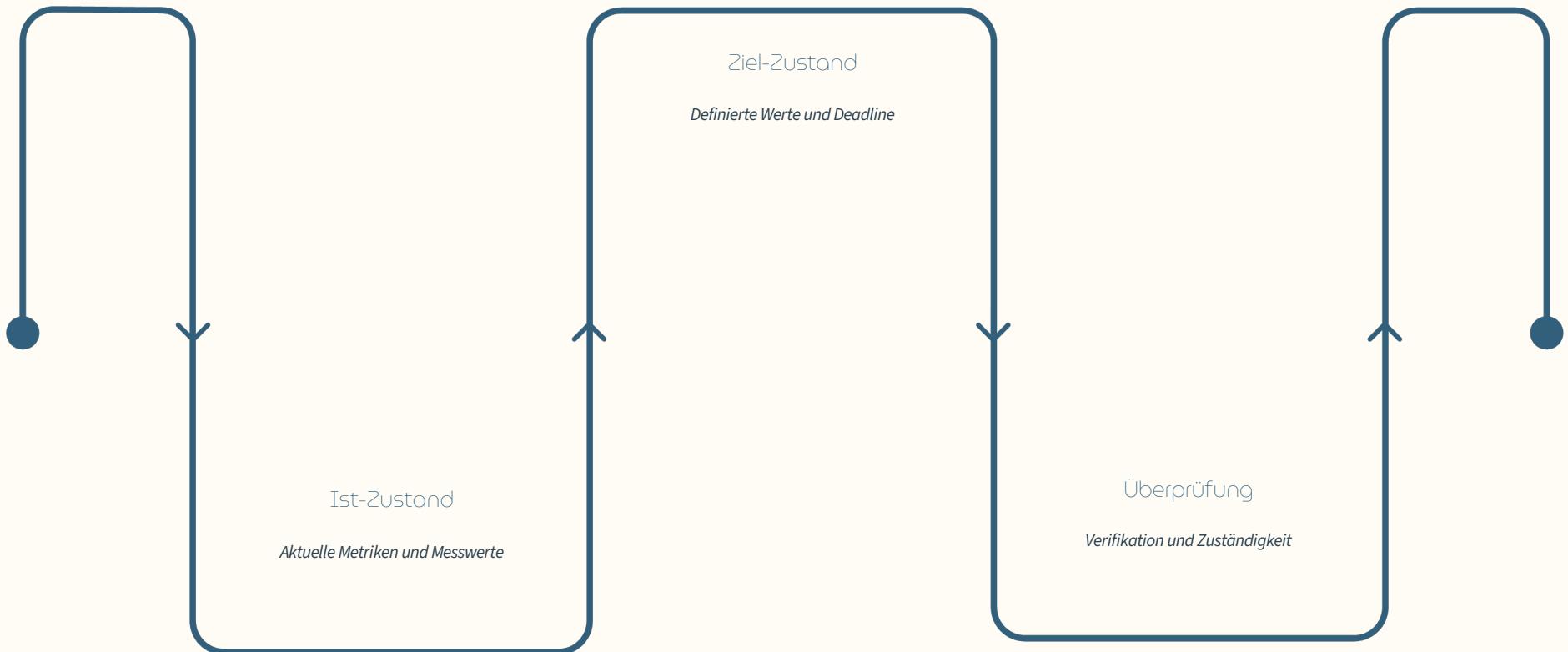
Was passiert typischerweise, wenn Qualitätsplanung erst kurz vor der Abnahme stattfindet? Die Folgen sind vorhersehbar und kostspielig: Anforderungen werden nachträglich interpretiert, Tests müssen unter Zeitdruck durchgeführt werden, und die Abnahmekriterien sind unklar oder umstritten.

Ein testbares Qualitätskriterium erkennen Sie daran, dass es messbar, eindeutig und verifizierbar ist. Statt „Die Web-App soll gut sein“ benötigen Sie konkrete Werte: „Ladezeit der Startseite < 2 Sekunden bei 100 gleichzeitigen Nutzern“ oder „Alle Formulare validieren Eingaben mit max. 200ms Verzögerung“.

Praxis-Check

Formulieren Sie jetzt ein konkretes, messbares Qualitätskriterium für eine Web-App aus Ihrem Projekt.

Qualitätsziele: Vom Ist zum Soll



Qualitätsziele sind messbare Zielwerte, die direkt aus den Anforderungen abgeleitet werden. Sie folgen dem bewährten SMART-Prinzip und schaffen Klarheit über den angestrebten Zustand. Der Prozess beginnt mit der Ermittlung des Ist-Zustands: aktuelle Fehlerquote, Performance-Werte, Supporttickets, Ausfallzeiten, Kundenzufriedenheit. Daraus wird der Ziel-Zustand mit konkreten Werten, Terminen und Verantwortlichen definiert – etwa „Crashrate < 0,1% bis Release“.

Qualitätsziele messbar formulieren

Ist-Zustand erfassen

Pragmatische Methoden: Log-Analyse, Monitoring-Daten, Ticket-Statistiken, kurze Stakeholder-Befragung

Ziel-Zustand definieren

Beispiel Zuverlässigkeit: „Verfügbarkeit 99,5% in Q2“ – Beispiel Benutzbarkeit: „80% der Testnutzer schließen Checkout ohne Hilfe ab“

Unterschied: Ziel vs. Maßnahme

Ein Qualitätsziel beschreibt den gewünschten Endzustand („Fehlerrate < 2%“), eine Qualitätsmaßnahme beschreibt den Weg dorthin („Einführung automatisierter Tests“).

Typische Fallen

- *Ziele zu ambitioniert oder zu vage formuliert*
- *Keine realistische Baseline vorhanden*
- *Fehlende Verantwortlichkeiten*

Qualitätslenkung: Steuerung im laufenden Betrieb

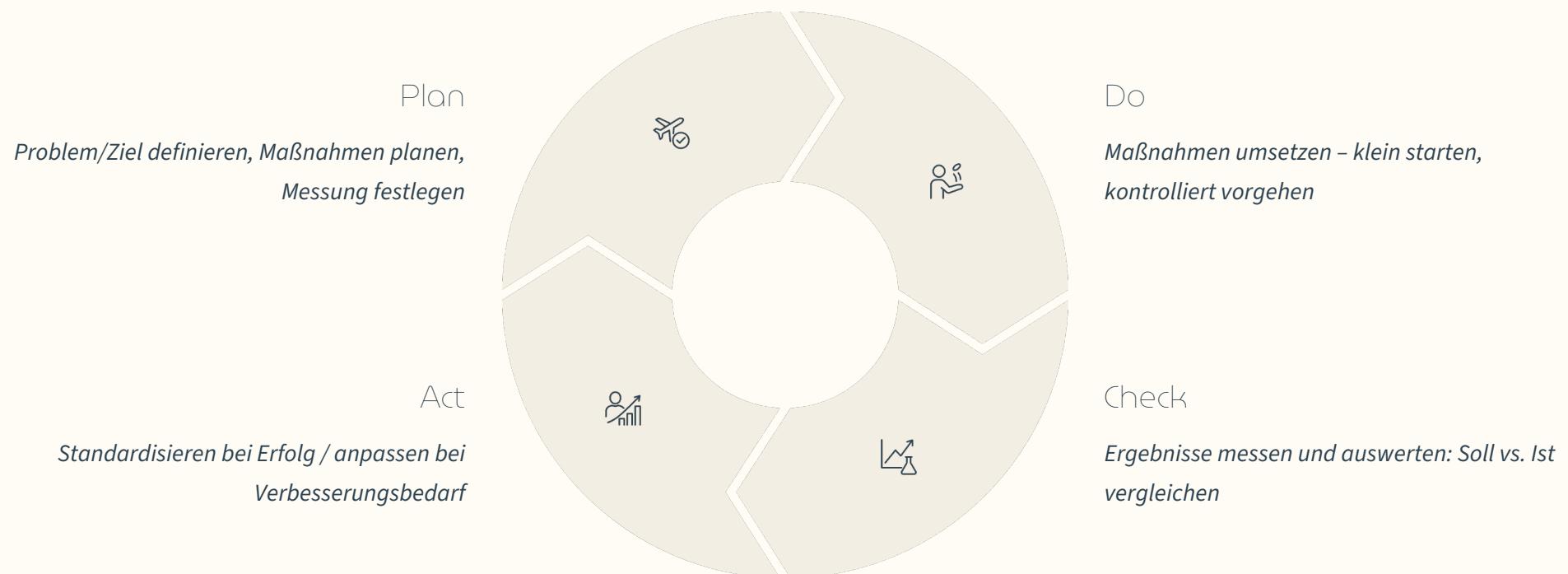


Qualitätslenkung bedeutet Umsetzung und aktive Steuerung im laufenden Projekt, damit die definierten Ziele tatsächlich erreicht werden. Ziel ist es, Abweichungen früh zu erkennen und sauber nachzusteuern.

Zentrale Lenkungsmaßnahmen umfassen Reviews und Inspections von Code, Anforderungen und Tests, gestaffelte Teststufen (Unit, Integration, System), definierte Freigabe-Gates (z. B. Merge nur mit Review und grünen Tests) sowie systematische Korrekturmaßnahmen bei Abweichungen.

Wenn Qualitätsziele regelmäßig verfehlt werden, reicht „mehr testen“ nicht aus. Analysieren Sie die Ursachen: Sind Prozesse unklar? Fehlen Skills? Ist die Architektur problematisch? Reviews spielen eine komplementäre Rolle zu Tests: Sie finden konzeptionelle Probleme, während Tests technische Fehler aufdecken.

PDCA-Zyklus: Kontinuierliche Verbesserung



PDCA ist keine einmalige Aktion, sondern ein kontinuierlicher Verbesserungszyklus. Der Zyklus wird immer wieder durchlaufen, um nachhaltige Qualitätssteigerungen zu erreichen.

PDCA in der IT-Praxis anwenden

Beispiel: Zu viele Bugs im Sprint

Plan: Ziel definieren „Bug-Rate um 30% senken“, Maßnahme „Pair Programming bei komplexen Features“, Metrik „Bugs pro Story Point“

Do: Pair Programming in 3 Sprints pilotieren, nur bei Features mit Komplexität > 5

Check: Bug-Rate messen: Vorher 2,4 Bugs/SP, nachher 1,6 Bugs/SP → 33% Reduktion

Act: Pair Programming als Standard für komplexe Features etablieren, im Team-Agreement verankern

Rennzahl für Codequalität

Sinnvoll: Code Coverage (mind. 80%), kombiniert mit Zyklomatischer Komplexität (< 10 pro Methode). Beide Metriken ergänzen sich und sind automatisiert messbar.

Wenn Maßnahmen nicht funktionieren

„Act“ bedeutet dann: Ursachenanalyse vertiefen, Hypothese anpassen, neue Maßnahme planen, PDCA-Zyklus neu starten – nicht die Maßnahme erzwingen.

Warum PDCA schwer durchzuhalten ist

Zeitdruck, fehlende Disziplin, mangelnde Messbarkeit. Hilft: PDCA in Retrospektiven verankern, kleine Zyklen wählen, Erfolge sichtbar machen.