

H A N D L U N G S F E L D E R

12 HANDLUNGSFELDER FÜR ERFOLGREICHES SAP- TESTMANAGEMENT

*Die vollständige Landkarte, mit Substanz,
Selbstbewertung und konkreten Handlungsempfehlungen*

Vier Dimensionen. 12 Handlungsfelder. Ein Ziel: Testmanagement, das in SAP-Projekten wirklich funktioniert.

METHODIK	01 • 02 • 03 • 04 Fundament, Objekte, Testfälle, Umfang
PROZESSE	05 • 06 • 07 • 08 Planung, Durchführung, Daten, Dokumentation
MENSCHEN	09 • 11 Reporting, Rollen & Organisation
TECHNOLOGIE	10 • 12 Automatisierung, Tools & Toolchain

SO NUTZEN SIE DIESES DOKUMENT

- Jedes Handlungsfeld enthält: Warum das Thema wichtig ist, was in der Praxis funktioniert und klärende Fragen.
- Bewerten Sie pro Feld Ihre eigene Situation: ✓ **Gut aufgestellt**, ~ **Verbesserungspotenzial**, ✗ **Handlungsbedarf**.
- Nutzen Sie die Übersicht auf der nächsten Seite als Einstiegspunkt für Gespräche mit Projektleitung und IT.

IHRE PERSÖNLICHE STANDORTBESTIMMUNG

Bewerten Sie jedes Handlungsfeld mit einem der drei Symbole. Eine ehrliche Standortbestimmung zeigt, wo die größten Hebel liegen.



GUT AUFGESTELLT

Sauber definiert, dokumentiert
und gelebt.



VERBESSERUNGSPOTENZIAL

Im Ansatz vorhanden, aber nicht
konsequent.



HANDLUNGSBEDARF

Fehlt oder entsteht nur ad hoc.

HANDLUNGSFELD	✓ GUT	~ MITTEL	✗ BEDARF
01 • Testmethodik	☐	☐	☐
02 • Testobjekte	☐	☐	☐
03 • Testfälle	☐	☐	☐
04 • Testumfang	☐	☐	☐
05 • Testplanung	☐	☐	☐
06 • Testdurchführung	☐	☐	☐
07 • Testdaten	☐	☐	☐
08 • Testdokumentation	☐	☐	☐
09 • Testreporting	☐	☐	☐
10 • Testautomatisierung	☐	☐	☐
11 • Testorganisation & Rollen	☐	☐	☐
12 • Technologie & Test-Tools	☐	☐	☐

Viele ✗? Sprechen Sie uns an. Ein kostenloser Erstcheck zeigt, wo die größten Hebel liegen. synaworks.com/testmanagement

DIE 12 HANDLUNGSFELDER IM DETAIL

01 TESTMETHODIK

METHODIK

Das Fundament: Vorgehen, Prinzipien und Methoden für ein strukturiertes Testmanagement. Ohne klare Methodik entscheidet der Zufall, was getestet wird.

WARUM WICHTIG?

- › Einheitliches Vorgehen verhindert chaotische Testphasen
- › Methodik verbindet Teststrategie mit Projektmethodik (SAP Activate / BP 2.0)
- › Klare Prinzipien ermöglichen Qualitätsentscheidungen unter Zeitdruck

SO GELINGT ES IN DER PRAXIS

- Teststrategie zu Beginn der Explore-Phase definieren
- Abstraktionsniveaus festlegen: generisch (Prototyp) und spezifisch (Integration)
- Methodik schriftlich im Testkonzept verankern, 20 bis 30 Seiten, pragmatisch

KLÄRENDE FRAGEN

- ? Haben wir eine schriftliche Teststrategie?
- ? Ist unsere Testmethodik mit der Projektmethodik abgestimmt?
- ? Wissen alle Beteiligten, wie wir testen, und warum?

✓ GUT AUFGESTELLT

Schriftliche Teststrategie, abgestimmt mit Projektleitung und Implementierungspartner.

~ VERBESSERUNGSPOTENZIAL

Strategie existiert, aber nicht konsequent kommuniziert.

✗ HANDLUNGSBEDARF

Kein explizites Vorgehen. Testphasen entstehen spontan.

02 TESTOBJEKTE

METHODIK

Was genau wird getestet? Prozesse, Funktionen, Schnittstellen, Systeme oder ganze Prozessketten? Wer das nicht klar definiert, testet zufällig.

WARUM WICHTIG?

- › Klare Testobjekte verhindern Lücken im Testumfang
- › Schnittstellen zu Drittsystemen werden systematisch mitgeplant
- › Differenzierung zwischen Customizing, RICEFW und Standard-SAP strukturiert die Testplanung

SO GELINGT ES IN DER PRAXIS

- Testobjekte aus der E2E-Prozesslandschaft ableiten (Level 0 bis 3)
- Schnittstellen als eigenständige Testobjekte definieren, nicht als Anhang
- Klare Abgrenzung: Was ist nicht im Testumfang?

KLÄRENDE FRAGEN

- ? Haben wir eine vollständige Liste unserer Testobjekte?
- ? Sind Schnittstellen zu Drittsystemen explizit als Testobjekte erfasst?
- ? Wissen wir, was wir bewusst nicht testen, und warum?

✓ GUT AUFGESTELLT

Vollständige Testobjekt-Liste mit klarer Scope-Abgrenzung. Schnittstellen explizit erfasst.

~ VERBESSERUNGSPOTENZIAL

Kernprozesse definiert. Schnittstellen und Sonderfälle noch lückenhaft.

✗ HANDLUNGSBEDARF

Keine strukturierte Erfassung. Was getestet wird, entsteht ad hoc.

03

TESTFÄLLE

PROZESSE

Die richtigen Tests zur richtigen Zeit. Nachvollziehbar, risikoorientiert und wiederverwendbar. Ein guter Testfall überlebt mehr als eine Testphase.

WARUM WICHTIG?

- › Testfälle auf Level-3-Aktivitätsebene sind das Optimum: aussagekräftig ohne übertriebene Detailtiefe
- › Wiederverwendbare Testfälle sparen in Folge-Releases erheblich Zeit
- › Klare Nomenklatur verhindert Chaos bei 300+ Testfällen

SO GELINGT ES IN DER PRAXIS

- Testfälle aus Level-3-Prozessmodellen ableiten
- Muster-Testfall gemeinsam erarbeiten, als Schablone für alle weiteren
- Nomenklatur festlegen: TF-[Prozess]-[Phase]-[Typ]-[Nr.]

KLÄRENDE FRAGEN

- ? Auf welcher Ebene schreiben wir Testfälle?
- ? Gibt es eine Schablone, an der sich alle orientieren?
- ? Sind unsere Testfälle nach dem Projekt noch nutzbar?

✓ GUT AUFGESTELLT

Testfälle auf Prozessschrittebene, klare Nomenklatur, wiederverwendbar.

~ VERBESSERUNGSPOTENZIAL

Testfälle existieren, aber inkonsistent im Detailgrad.

✗ HANDLUNGSBEDARF

Keine strukturierten Testfälle. Tests laufen aus dem Gedächtnis oder aus Excel-Listen.

04

TESTUMFANG

PROZESSE

Nicht alles testen. Das Richtige testen. Risiken erkennen und den Testaufwand sinnvoll steuern. Wer alles gleich priorisiert, priorisiert nichts.

WARUM WICHTIG?

- › Risikobasierter Ansatz schützt das Go-Live-Datum
- › Bewusste Priorisierung (HIGH/MEDIUM/LOW) schafft Flexibilität bei Zeitdruck
- › Klare Scope-Abgrenzung verhindert „endless testing“

SO GELINGT ES IN DER PRAXIS

- Testfälle nach Business Impact priorisieren: Kritikalität, Auswirkung, Nutzungshäufigkeit
- HIGH-Defekte sind Go-Live-Blocker ohne Ausnahme
- Scope-Änderungen formal managen, nicht still integrieren

KLÄRENDE FRAGEN

- ? Haben wir Go-Live-Kriterien definiert, bevor der erste Test läuft?
- ? Wissen wir, welche Testfälle wir notfalls weglassen dürfen?
- ? Wie gehen wir mit neuen Anforderungen in der Testphase um?

✓ GUT AUFGESTELLT

Explizite Priorisierung, klare Go-Live-Kriterien, definierter Umgang mit Scope-Änderungen.

~ VERBESSERUNGSPOTENZIAL

Kernprozesse sind High, aber die Priorisierung der Randfälle fehlt.

✗ HANDLUNGSBEDARF

Keine Priorisierung. Go-Live-Entscheidung basiert auf Bauchgefühl.

05

TESTPLANUNG

PROZESSE

Wer macht was, wann und mit welchen Ressourcen? Gute Planung schafft Verbindlichkeit und schützt vor zu wenig Zeit kurz vor dem entscheidenden Test.

WARUM WICHTIG?

- › Testplanung drei Wochen vor IT1 ist Krisenmanagement, keine Planung
- › Ressourcen für Tests müssen von der Linienorganisation freigegeben werden
- › Zeitpuffer für Re-Tests nach Defekt-Behandlung wird oft nicht eingeplant

SO GELINGT ES IN DER PRAXIS

- Testplanung beginnt in der Explore-Phase als Meilenstein
- Für jede Testphase: Zeitplan, Ressourcen, Systemverfügbarkeit, Testdaten
- Bugfix-Puffer explizit einplanen (Faustregel: 30 bis 40 % der Testlaufzeit)

KLÄRENDE FRAGEN

- ? Sind Testtermine bereits im Gesamtprojektplan verankert?
- ? Haben die Key-User ihre Testkapazität bestätigt?
- ? Was passiert, wenn ein Testtermin nicht gehalten werden kann?

✓ GUT AUFGESTELLT

Testtermine im Projektplan. Ressourcen bestätigt. Bugfix-Puffer eingeplant.

~ VERBESSERUNGSPOTENZIAL

Plan existiert, aber Ressourcen sind nicht formell bestätigt.

✗ HANDLUNGSBEDARF

Kein expliziter Testplan. Die Testphase beginnt, wenn die Implementierung „fertig“ ist.

06

TESTDURCHFÜHRUNG

PROZESSE

Tests effizient organisieren, steuern und begleiten und dabei den Überblick behalten. Kein Testergebnis darf verloren gehen.

WARUM WICHTIG?

- › Undisziplinierte Durchführung produziert keine verwertbaren Ergebnisse
- › Tagesaktuelle Steuerung ist entscheidend, wenn Defekte die Ausführung blockieren
- › Jeder fehlgeschlagene Testfall muss mit einem Defekt verlinkt sein

SO GELINGT ES IN DER PRAXIS

- Täglicher Defect Stand-up (15 Min.) während aktiver Testphasen
- Cloud ALM als single source of truth für alle Testergebnisse
- Kein Defekt ohne Verlinkung zum Testfall, Priorität nach Business Impact

KLÄRENDE FRAGEN

- ? Wie stellen wir sicher, dass alle Testergebnisse dokumentiert werden?
- ? Wie lange dauert es vom Defekt bis zur Fehlerbehebung?
- ? Haben wir einen klaren Defect-Workflow, den alle kennen?

✓ GUT AUFGESTELLT

Strukturierte Durchführung, Testergebnisse lückenlos in Cloud ALM, täglicher Defect-Austausch.

~ VERBESSERUNGSPOTENZIAL

Durchführung läuft, aber die Dokumentation ist inkonsistent.

✗ HANDLUNGSBEDARF

Tests werden durchgeführt, aber Ergebnisse nicht systematisch erfasst.

07

TESTDATEN

PROZESSE

Ohne passende Testdaten kein aussagekräftiger Test. Verfügbarkeit, Qualität und Vorbereitung entscheiden und werden systematisch unterschätzt.

WARUM WICHTIG?

- › Testfälle ohne passende Testdaten sind wertlos, beide gemeinsam planen
- › Datenmigration und Testdaten sind dasselbe Thema, oft in zwei Teams aufgeteilt
- › Offshore-Anteile der Migration erfordern einen expliziten QS-Mechanismus

SO GELINGT ES IN DER PRAXIS

- Testdaten ausgehend von den Testfällen anlegen, nicht umgekehrt
- Migrations-Test dreidimensional: technisch, fachlich, prozessual
- Mandanten-Konzept früh klären: Wer stellt Stammdaten für welches System bereit?

KLÄRENDE FRAGEN

- ? Haben wir für jeden Testfall die benötigten Testdaten definiert?
- ? Wer ist verantwortlich für die Stammdaten-Bereitstellung je Testsystem?
- ? Wie testen wir die migrierten Daten vor dem Integrationstest?

✓ GUT AUFGESTELLT

Testdaten aus Testfällen abgeleitet. Migrationstest-Konzept steht. Verantwortlichkeiten klar.

~ VERBESSERUNGSPOTENZIAL

Stammdaten vorhanden, aber ad hoc bereitgestellt.

✗ HANDLUNGSBEDARF

Testdaten werden kurz vor der Testphase zusammengesucht.

08

TESTDOKUMENTATION

PROZESSE

Ergebnisse, Nachweise und Entscheidungen nachvollziehbar dokumentieren, ohne unnötige Bürokratie. Wenn es nicht dokumentiert ist, ist es nicht passiert.

WARUM WICHTIG?

- › Go-Live-Entscheidungen brauchen eine nachvollziehbare Grundlage
- › Revisionssichere Dokumentation schützt alle Beteiligten
- › Nach dem Projekt nutzbare Test-Asets ersparen beim nächsten Release erheblich Aufwand

SO GELINGT ES IN DER PRAXIS

- Cloud ALM als führendes Dokumentationssystem, kein Excel parallel
- Screenshots und Logs bei fehlgeschlagenen Tests sind Pflicht
- Testphasen-Abschlussbericht mit Qualitätsaussage als Pflicht-Lieferobjekt

KLÄRENDE FRAGEN

- ? Können wir nachweisen, was wann getestet wurde?
- ? Sind unsere Testdokumente als Revisionsbasis nutzbar?
- ? Wer ist verantwortlich für die Vollständigkeit der Dokumentation?

✓ GUT AUFGESTELLT

Vollständige Dokumentation in Cloud ALM. Abschlussberichte je Phase. Test-Asets nach Go-Live nutzbar.

~ VERBESSERUNGSPOTENZIAL

Kern-Dokumentation vorhanden. Abschlussberichte fehlen oder sind lückenhaft.

✗ HANDLUNGSBEDARF

Dokumentation über Excel-Sheets verteilt. Kein zentrales System.

Aus Testergebnissen werden Entscheidungen: Transparenz schaffen, Risiken und Fortschritt sichtbar machen. Gutes Reporting entsteht rückwärts, vom Empfänger her.

WARUM WICHTIG?

- › Drei Empfängerebenen brauchen drei unterschiedliche Formate
- › Go-Live-Entscheidungen auf Bauchgefühl sind vermeidbar, wenn Kriterien früh definiert wurden
- › Testkoordinatoren müssen Dashboards selbständig lesen können

SO GELINGT ES IN DER PRAXIS

- Empfänger zuerst: Wer bekommt welchen Report für welche Entscheidung?
- Drei Ebenen: Tages-Report, Wochenstatus, Abschlussbericht
- Ampel-Kriterien vorab definieren: Was ist Grün, was ist Rot?

KLÄRENDE FRAGEN

- ? Wer bekommt welchen Report, zu welchem Zeitpunkt?
- ? Welche Information braucht der Lenkungsausschuss für die Go-Live-Entscheidung?
- ? Sind unsere Ampel-Kriterien schriftlich festgelegt?

✓ GUT AUFGESTELLT

Drei Berichtsebenen etabliert. Ampel-Kriterien definiert. Lenkungsausschuss-Report strukturiert.

~ VERBESSERUNGSPOTENZIAL

Status wird berichtet, aber ohne festes Format.

✗ HANDLUNGSBEDARF

Kein Reporting. Status wird auf Nachfrage mündlich kommuniziert.

Automatisieren, wo es sinnvoll ist, und dort, wo Automatisierung langfristig echten Mehrwert bringt. Automatisierung ist ein Mittel, kein Ziel.

WARUM WICHTIG?

- › Im SAP-Transformationsprojekt oft zu früh priorisiert
- › Nach Go-Live für Regressionstests erheblicher ROI möglich
- › Gute manuelle Testfälle sind Voraussetzung für sinnvolle Automatisierung

SO GELINGT ES IN DER PRAXIS

- Im Transformationsprojekt bewusst zurückstellen, Fokus auf Testfall-Qualität
- Bereits bei der Testfallerstellung die Automatisierbarkeit mitdenken
- Nach Go-Live mit HIGH-priorisierten Regressions-Szenarien beginnen

KLÄRENDE FRAGEN

- ? Sind unsere manuellen Testfälle als Basis für Automatisierung geeignet?
- ? Für welche Szenarien wäre der ROI am größten?
- ? Wer pflegt nach Go-Live die automatisierten Tests?

✓ GUT AUFGESTELLT

Klare Post-Go-Live-Strategie für Automatisierung. Testfälle automatisierbar aufgebaut.

~ VERBESSERUNGSPOTENZIAL

Automatisierung ist ein Thema, aber es gibt keinen Plan.

✗ HANDLUNGSBEDARF

Automatisierung wird ad hoc diskutiert, ohne Strategie.

Klare Verantwortlichkeiten, funktionierende Zusammenarbeit und die richtige Einbindung von IT und Fachbereichen. Was keine klare Verantwortung hat, wird nicht gemacht.

WARUM WICHTIG?

- › Testrollen auf bestehende Projektrollen überführen, keine neuen Ressourcen
- › Unabhängiger Testmanager strukturell nötig, wenn der Implementierungspartner alles steuert
- › Test-Koordinatoren nach Prozessen organisieren, nicht nach Modulen

SO GELINGT ES IN DER PRAXIS

- RACI-Matrix: Testmanager, Test-Koordinator, Testfall-Ersteller, Tester
- Rollenbeschreibungen vor der ersten Schulung veröffentlichen
- Kapazität der Key-User mit der Linienorganisation formell bestätigen

KLÄRENDE FRAGEN

- ? Wer ist Test-Koordinator für welchen E2E-Prozess?
- ? Sind Rollenbeschreibungen schriftlich und allen bekannt?
- ? Hat jeder Key-User seine Testkapazität formell bestätigt?

✓ GUT AUFGESTELLT

RACI vorhanden, Rollen bekannt, Kapazitäten bestätigt.

~ VERBESSERUNGSPOTENZIAL

Rollen grundsätzlich klar, aber nicht schriftlich.

✗ HANDLUNGSBEDARF

Unklar, wer was testet. Testverantwortung liegt implizit beim Implementierungspartner.

Die passende Toolchain unterstützt den Prozess. Sie ersetzt ihn nicht. Ein schlecht konfiguriertes Tool macht schlechtes Testmanagement nicht besser.

WARUM WICHTIG?

- › Cloud ALM bietet End-to-End-Traceability: Anforderung → Feature → Testfall → Defekt
- › Tool-Setup in der Explore-Phase ist eine Architekturentscheidung, keine Admin-Aufgabe kurz vor dem Test
- › Deployment-Management und Testmanagement müssen im selben Tool abgebildet sein

SO GELINGT ES IN DER PRAXIS

- Cloud ALM in der Explore-Phase konfigurieren: Prozesshierarchie, Templates, Governance
- Verlinkungskette aufbauen: Von einem Defekt rückwärts zur Anforderung navigierbar?
- Governance definieren: Wer legt an? Wer gibt frei? Wer schließt Defekte?

KLÄRENDE FRAGEN

- ? Ist Cloud ALM so konfiguriert, dass die Traceability lückenlos ist?
- ? Wer ist verantwortlich für Governance und QS im Tool?
- ? Können Testkoordinatoren Cloud ALM Dashboards selbständig interpretieren?

✓ GUT AUFGESTELLT

Cloud ALM als single source of truth. Traceability lückenlos. Governance definiert.

~ VERBESSERUNGSPOTENZIAL

Cloud ALM im Einsatz, aber die Traceability ist lückenhaft.

✗ HANDLUNGSBEDARF

Excel und E-Mail als Testdokumentation. Cloud ALM nicht oder nur rudimentär genutzt.

LUST AUF EINEN REALITÄTSCHECK?

Nehmen Sie die 12 Handlungsfelder als Spiegel für Ihr Projekt. In einem kostenfreien 30-minütigen Erstgespräch zeigen wir Ihnen, wo Ihr größter Hebel liegt.

- ✓ Unverbindlich ✓ Ohne Vorbereitung
- ✓ Direkte Einschätzung aus 20+ Jahren SAP-Praxis

WEB synaworks.com/testmanagement

E-MAIL consulting@synaworks.com

© 2026 Synaworks GmbH · Alle Rechte vorbehalten · Dieses Dokument darf mit Quellenangabe frei geteilt werden.