

The logo for SpiralOS features a large, light gray circle with a vertical line passing through its center. A smaller spiral is centered on the vertical line, overlapping the larger circle. The word "SpiralOS" is written in a serif font across the middle of the circle. A registered trademark symbol (®) is positioned to the right of the word.

SpiralOS®

Das Betriebssystem für wertebasierten Wandel.

Framework für Führung in Spannungsfeldern – wenn Meetings nicht mehr reichen,  
Pläne nicht mehr greifen und Systeme sich gegenseitig bremsen.

# Definition, Resultate, Nutzen

## SpiralOS®

schafft eine **gemeinsame Sprache**, ein **klares Vorgehen** und **wiederholbare Bausteine**, damit Transformation nicht im Aktionismus endet – sondern in **messbarer Umsetzung**.



- Eindeutiger Fokus
- Schnelle Entscheidungen
- Messbares Lernen



- Transformation strukturieren – statt „Initiativen-Suppe“
- Operating Model & Zusammenarbeit klar beschreiben
- Teams schnell in Umsetzung bringen – Pilot → Skalierung

# SpiralOS®-Migration

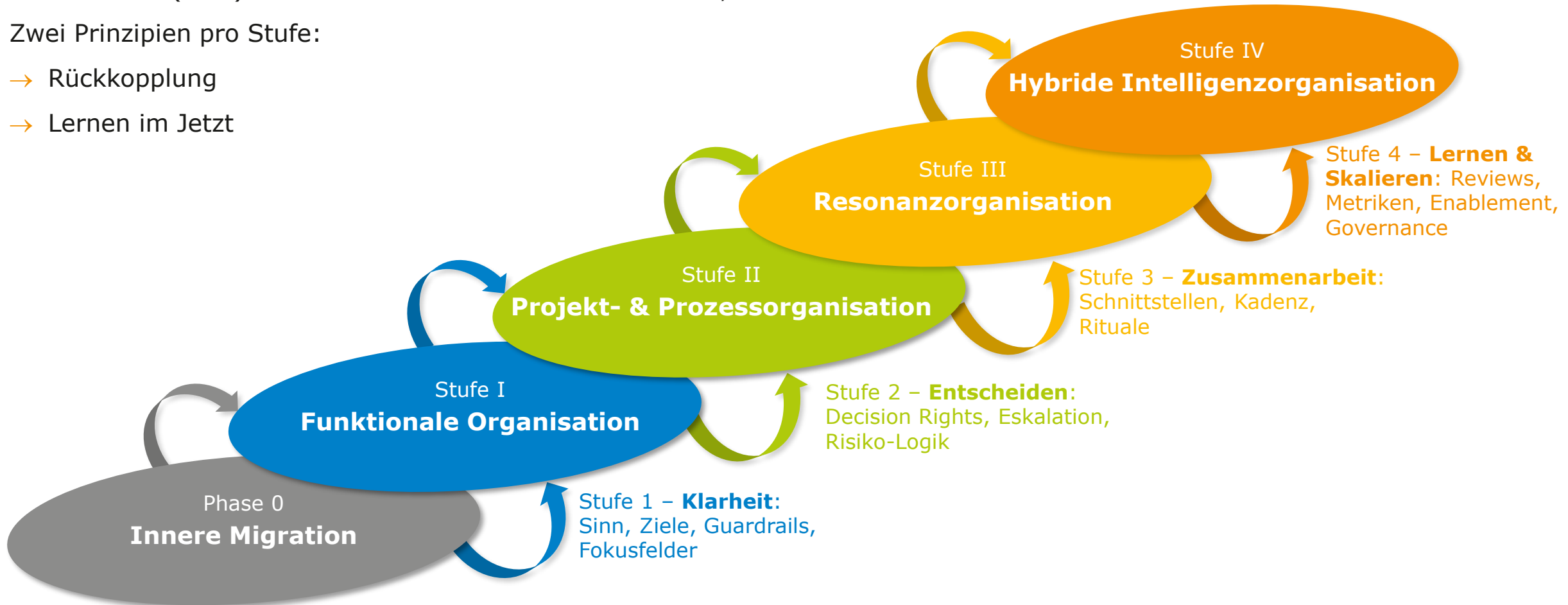
Von Abteilungen zu vernetzten Wertströmen und klaren Schnittstellen



Klein starten (Pilot) → messbar lernen → nur das skalieren, was wirkt.

Zwei Prinzipien pro Stufe:

- Rückkopplung
- Lernen im Jetzt



# Woran Organisationen scheitern, bevor sie scheitern

Problemraum: Typische Symptome in Transformationen

Wenn diese Signale häufiger auftreten, fehlt meist ein gemeinsames Betriebsmodell:

## Unklare Prioritäten

Alles ist wichtig. Portfolios wachsen – Fokus schrumpft.

## Entscheidungen dauern

Keine klare Zuständigkeit, zu viele Schleifen, zu viel Risiko-Angst.

## Silos & Doppelarbeit

Teams optimieren lokal, Gesamtwirkung bleibt unsichtbar.

## Meeting-Inflation

Koordination ersetzt Zusammenarbeit – ohne Ergebnisartefakte.

## Initiativen „versanden“

Pilot-Erfolge werden nicht skaliert, Lernschleifen fehlen.

## Kultur vs. Struktur

Kulturprogramme ohne Operating Model – oder umgekehrt.

**SpiralOS®** setzt hier an: Es macht sichtbar, welche Regeln, Routinen und Artefakte fehlen, damit Wandel anschlussfähig wird.

# In 60 Sek.: Das Problem – die Lösung – der Nutzen

Übersicht



## Problem

Viele Organisationen sind im Wandel zugleich zu schnell und zu langsam:  
Viele Initiativen, aber zu wenig Klarheit, Priorisierung und Anschlussfähigkeit in der täglichen Arbeit.

## SpiralOS®

Ein „Betriebssystem“ für Veränderung: Es verbindet Diagnose, Design und Umsetzung über wiederholbare Bausteine – mit klaren Artefakten, Rollen und Routinen.

## Nutzen

Mehr Fokus, bessere Entscheidungen und schnelleres Lernen.  
**SpiralOS®** reduziert Reibung und macht Fortschritt sichtbar – vom Pilot bis zur Skalierung.

Kernaussage: **SpiralOS®** übersetzt Veränderung in ein verständliches, messbares Betriebsmodell – damit Teams wissen, was „als Nächstes“ zu tun ist.

# Was ist SpiralOS®?

Ein Operating System: Prinzipien + Bausteine + Routinen



**SpiralOS®** ist ein Betriebsmodell für Veränderung. Wie ein Betriebssystem sorgt es dafür, dass unterschiedliche „Programme“ (Initiativen, Projekte, Teams) unter gemeinsamen Regeln funktionieren.

## Die 5 Ebenen eines „Operating Systems“

### Sinn & Richtung

Warum wir das tun: Ziele, Narrative, Guardrails

### Entscheiden

Rollen, Decision Rights, Eskalationspfade

### Zusammenarbeiten

Schnittstellen, Meeting-Kadenz, Rituale

### Liefern

Backlogs, Priorisierung, Portfolio-Steuerung

### Lernen

Metriken, Reviews, Feedbackschleifen

## Metapher: OS-Schichten

Initiativen & Produkte

Teams & Zusammenarbeit

**Entscheidungen & Portfolio**

Ziele & Guardrails

Sinn & Identität

„Spiral“ = iteratives Lernen

# Alleinstellungsmerkmal (USP)

Differenzierung: Warum **SpiralOS®** anders wirkt als klassische Transformationsprogramme



**SpiralOS®** kombiniert drei Dinge, die sonst oft getrennt laufen:



## Gemeinsame Sprache

Einheitliche Begriffe und Artefakte (z. B. Rollen, Regeln, Metriken), damit Teams dasselbe meinen – und Missverständnisse sinken.



## Wiederholbare Bausteine

Ein Toolkit aus Playbooks & Templates, das von Diagnose bis Umsetzung reicht – statt „Workshop-Folien ohne Anschluss“.



## Spiralförmiger Lernzyklus

Kurze Loops mit klaren Reviews: Fortschritt wird sichtbar, Hypothesen werden überprüft, das OS wird iterativ besser.

USP in einem Satz

**SpiralOS® macht Wandel operational: Ambitionen werden zu Artefakten, Routinen und Messpunkten, die Teams sofort anwenden.**

# So läuft SpiralOS® typischerweise ab

Vorgehen: Von Diagnose zu Umsetzung – in kurzen, planbaren Loops

Ein bewährtes Vorgehen, das an Umfang und Reifegrad angepasst wird:

## 1. Diagnose

Reifegrad-Check, Schmerzpunkte, Zielbild

## 2. Design

Operating Model + Bausteine definieren

## 3. Pilot

Routinen & Artefakte im Alltag testen

## 4. Skalierung

Rollout, Enablement, Governance

### Ergebnisse pro Phase (Beispiele)

- Klarer Problem- und Zielrahmen (inkl. messbarer Indikatoren)
- Operating-Model-Skizze: Rollen, Entscheidungspfade, Kadenz
- Pilot-Playbook: Ritual-Set, Artefakte, Verantwortlichkeiten
- Skalierungsplan: Enablement, Governance, Messroutine

### Wichtig in der Praxis

- Co-Creation statt „Berater übergeben Folien“
- Artefakte sind so leicht wie möglich, so klar wie nötig
- Jeder Loop endet mit einem Review + konkretem Next Step
- Messung folgt dem Zweck: wenige, relevante Signale

# Der SpiralOS® -Loop

Mechanik: Die kleinste Einheit → beobachten → ausrichten → umsetzen → lernen



## Sense

Signale sammeln  
& Reibung sichtbar machen

## Align

Ziele, Regeln  
& Prioritäten klären

## Act

Umsetzen mit  
Routinen & Artefakten

## Learn

Review, Metriken  
& nächste Anpassung

## Was der Loop liefert

- Ein klares „Next best step“-Backlog für Organisation & Teams
- Konkrete Artefakte (z. B. Entscheidungsregel, Kadenzplan)
- Ein Review-Rhythmus, der Lernen messbar macht
- Transparenz: Was hat geholfen – was war Overhead?

# SpiralOS® Toolkit

Bausteine, die je nach Bedarf kombiniert werden



SpiralOS ist modular: Sie wählen, was im aktuellen Reifegrad wirklich hilft.

## ■ Reifegrad-Scan

Kurzer Check: Klarheit, Entscheiden, Zusammenarbeit, Lernen

## ■ Operating-Model-Canvas

Rollen, Schnittstellen, Steuerung – auf einer Seite

## ■ Decision Rights

Wer entscheidet was – und wie wird eskaliert?

## ■ Portfolio-Kadenz

Priorisierung, Reviews, Kapazitäts- und Impact-Blick

## ■ Team-Playbooks

Rituale, Artefakte, Schnittstellen – minimal & wirksam

## ■ Metriken & Reviews

Wenige, relevante Signale + ein Routine-Rhythmus

**Lieferobjekte sind editierbar und bleiben bei Ihnen (Templates/ Artefakte) – damit SpiralOS nach dem Projekt weiterlebt.**

# Wo SpiralOS® typischerweise eingesetzt wird

Use Cases mit typischen Ergebnissen



## Wachstum & Skalierung

Mehr Teams, mehr Schnittstellen → klare Regeln, Kadenz und Ownership.

### Typische Ergebnisse:

- Fokus-Portfolio
- Schnittstellenklarheit
- Kürzere Entscheidungsschleifen

## Transformation / Re-Organisation

Strukturen ändern sich → der Alltag muss trotzdem liefern.

### Typische Ergebnisse:

- Operating Model
- Rollen & Decision Rights
- Rituale, die Stabilität geben

## Post-Merger / Integration

Zwei Kulturen, zwei Prozesse → ein gemeinsames Mindest-OS.

### Typische Ergebnisse:

- Gemeinsame Sprache
- Gemeinsame Governance
- Gemeinsame Lernroutine

## Produktivität & Fokus

Meeting-Overload, unklare Prioritäten, Reibung zwischen Teams.

### Typische Ergebnisse:

- Weniger, bessere Meetings
- Klarer „Next step“
- Messbare Reibungsreduktion

# Schnell genug für Momentum, robust genug für Alltag



Umsetzung: Ein leichtgewichtiges Setup – mit klaren Rollen

Typischer Rahmen (anpassbar):

## Beispiel-Zeitplan

**Woche 1–2** Diagnose: Interviews, Quick Scan, Hypothesen

**Woche 3–4** Design: Operating Model, Artefakte, Kadenz

**Woche 5–8** Pilot: Routinen einführen, Feedback, Anpassung

**Woche 9–12** Skalierung: Enablement, Governance, Messroutine

## Rollen im Setup

- Sponsor: schafft Priorität & Schutzraum
- Core Team: co-designt Artefakte + betreibt Kadenz
- Teams: testen im Alltag, liefern Feedback
- Enablement: Training, Communities, Onboarding

## Erfolgsfaktor

**Nicht mehr „Change“ – sondern Betrieb:**  
**SpiralOS® wird Teil der Routine (Reviews, Entscheidungen, Lernen).**

# Wie Erfolg sichtbar wird

Messung, die Verhalten verbessert – statt Reporting zu produzieren

SpiralOS® nutzt wenige, aber klare Signale – kombiniert aus Output-, Outcome- und „Friction“-Metriken.



## Output (Liefen)

- Durchlaufzeiten
- Plan vs. Ist (Trend, nicht Punkt)
- Abschlussrate von Initiativen



## Outcome (Wirkung)

- Kundennutzen-/ Qualitätsindikatoren
- Wertbeitrag pro Fokusfeld
- Risiko-/Compliance-Signale



## Friction (Reibung)

- Entscheidungsdauer
- Meeting-Zeit pro Woche
- Schnittstellen-„Ping-Pong“

Mess-Rhythmus (Beispiel)

**Wöchentlich: Team-Review · 2-wöchentlich: Pilot-Retro · monatlich: Portfolio/OS-Review ·  
quartalsweise: Reifegrad-Check**

The logo for SpiralOS features a large, light gray circle with a vertical line passing through its center. A smaller spiral is centered on the vertical line, overlapping the letter 'O' in the word 'SpiralOS'. The word 'SpiralOS' is written in a serif font, with the 'O' being significantly larger than the other letters. A registered trademark symbol (®) is positioned to the right of the word.

SpiralOS®

Das Betriebssystem für wertebasierten Wandel.

**Nächste Schritte: Von Interesse zu Momentum in 2 Wochen**

# SpiralOS® Discovery + Pilot

Call to Action: Termin für Discovery vereinbaren



## 1) Discovery

**60–90 min**

Scope + Hypothesen +  
nächster Schritt

## 2) Pilot

**2–4 Wochen, 2–4 Teams**

Loop + Artefakte + Metriken  
+ Entscheidung „Scale/Stop“

## 3) Scale

**6–12 Wochen**

Enablement + Governance +  
Review-Rhythmus

**Bringen Sie 1–2 Beispiele für aktuelle Initiativen mit. Wir liefern am Ende ein klares Pilot-Design + nächste Schritte.**

# Anhang: FAQ (kurz)

Häufige Fragen im Sales-Gespräch



## **Ist SpiralOS® „Agile“?**

**SpiralOS®** ist methodenoffen. Es ergänzt vorhandene Vorgehensweisen durch klare Regeln, Artefakte und Lernroutinen.

## **Brauchen wir ein großes Programm?**

Nein. **SpiralOS®** startet klein (Pilot) und skaliert über wiederholbare Bausteine.

## **Wie verhindert ihr Overhead?**

Durch Modularität, kurze Loops und die Regel:

jedes Artefakt muss im Alltag genutzt werden – sonst wird es gestrichen.

## **Was bleibt nach dem Projekt?**

Templates, Playbooks und ein Mess-/ Review-Rhythmus, der vom Core Team weiterbetrieben wird.

# CREARO

THE VALUE CREATOR



**xxx**

xxx

mobil: +49 xxx

email: xxx@spiralos.de