ELEKTRONISCHES BEGLEITMATERIAL

Der Team-Katalog – Ein Nachschlagewerk für Teammodelle

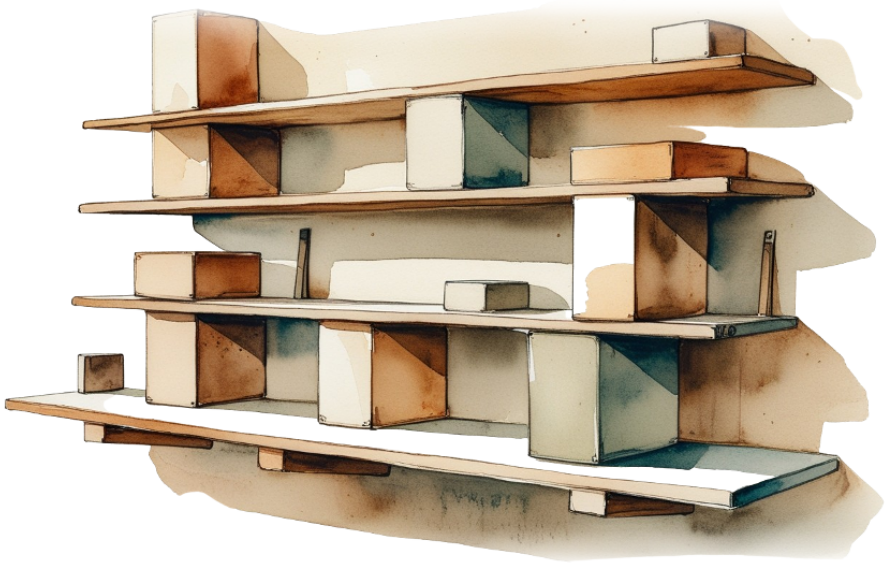
H.E.A.R. · Humanity · Empathy · Authenticity · Resilience

Begleitmaterial zum Buch
„Zeitlos Verwurzelt – Führung als Akt des Zuhörens“

KAVANARA
CONSULTING · TECHNICALS · SOLUTIONS

Version 1.0 · 2026
© Matthias Leo Wegner · kavanara.de

Der Team-Katalog – Ein Nachschlagewerk für Teammodelle



Dieses Kapitel ist anders. Es ist kein Text zum Durchlesen. Es ist ein Regal.

In Kapitel 12.2 hast du die Architektur-Logik kennengelernt: Die vier Funktionen im Ökosystem, die drei Dimensionen – Unsicherheit, Kopplung, Kritikalität –, die Interaktionsmodi zwischen Teams, die verschiedenen Organisationsarchitekturen.

Hier findest du den Katalog. Teammodelle im Detail. Nicht alle mit gleicher Tiefe – einige vollständig beschrieben, andere kompakter. Mit ihren Charakteristika, ihren Rollen, ihren Stärken und Herausforderungen.

Wie du diesen Katalog nutzt:

Du stehst vor einer konkreten Entscheidung. Du fragst dich: Welches Teammodell passt zu meiner Situation?

Dann tu das: Zuerst die Diagnose – beantworte die Diagnose-Fragen am Ende. Dann die Funktion – welche der vier Grundfunktionen soll das Team erfüllen? Dann die Dimension – wo steht es auf den drei Achsen? Dann lies die Detail-Beschreibungen der Modelle, die in Frage kommen. Und schließlich: Nutze die Entscheidungsmatrizen – aber nicht als Rezept, sondern als Kompass.

Wichtig: Dieser Katalog unterscheidet zwischen Teammodellen – wie ein einzelnes Team organisiert ist – und Skalierungsstrukturen – wie mehrere Teams koordiniert werden. Die Skalierungsstrukturen findest du in einem eigenen Abschnitt.

Schnellreferenz: Die vier Grundfunktionen

Jede Organisation braucht diese vier Funktionen. Die Frage ist nicht ob, sondern wie und wer:

Funktion	Architektur-Bild	Kernfrage	Erfolgsmaß
Wertschöpfer (Stream-Aligned)	Die tragenden Wände	Wofür ist das Gebäude da?	Kundenwert pro Zeiteinheit

Funktion	Architektur-Bild	Kernfrage	Erfolgsmaß
Fundamentleger (Platform)	Das Fundament	Worauf bauen andere?	Können andere selbstständig arbeiten?
Lehrmeister (Enabling)	Der Baumeister	Wer gibt Wissen weiter?	Werden wir nicht mehr gebraucht?
Spezialisten (Complicated-Subsystem)	Der Statiker	Wo braucht es Spezialwissen?	Ist die Komplexität gekapselt?

Schnellreferenz: Die drei Interaktionsmodi

Modus	Architektur-Bild	Wann passend	Wann wechseln
Collaboration	Das Baugerüst	Intensive gemeinsame Arbeit nötig	Wenn das “Gebäude” steht
X-as-a-Service	Der Anschluss	Klare, stabile Schnittstelle möglich	Wenn Anpassung nötig wird
Facilitating	Das Lehrgerüst	Ein Team befähigt ein anderes	Wenn das Team selbst trägt

Die Evolution: Was als Collaboration beginnt, sollte irgendwann zu X-as-a-Service werden. Das ist nicht Scheitern – das ist Reife.

Die vier Kern-Teammodelle

Diese vier Modelle bilden das Fundament moderner Teamorganisation. Sie stammen aus der Team-Topologies-Forschung von Skelton und Pais[126] und decken die wesentlichen Funktionen ab, die jede Organisation braucht.

Die Wertschöpfer (Stream-Aligned Teams)

Ein Stream-Aligned Team trägt ein Produkt von der ersten Idee bis zum Kunden. Alles drin: Code, Design, Testing, Deployment. End-to-End. Das ist ihr Stolz – und ihre Last.

In der Architektur-Metapher: Sie bauen die tragenden Wände. Ohne sie gibt es keinen Grund für das Gebäude.

Charakteristika: Cross-funktional – alle Fähigkeiten im Team, die für den Wertstrom nötig sind. Langlebig – produkt-basiert, nicht projekt-basiert. Autonom über das “Wie”, klare Erwartungen über das “Was”. Fokus: Kundennutzen liefern.

Kernrollen: Product Owner / Product Manager, Developers (Backend, Frontend), QA / Test Engineer, UX Designer.

Disziplinarische Führung: Meist keine im Team. Fachlich via Chapters oder Matrix.

In den Dimensionen: Unsicherheit mittel bis hoch – das Team muss im Nebel navigieren können. Kopplung lose – möglichst unabhängig, wenige Abhängigkeiten zu anderen Teams. Kritikalität niedrig bis mittel – Fehler sind korrigierbar. Wissen sitzt beim Kunden – das Team ist nah dran.

Wann die richtige Wahl: Digitale Produkte mit kontinuierlicher Weiterentwicklung. Klare Kundensegmente oder Produktbereiche. Mittlere bis niedrige Kritikalität.

Stärken: Schnelle Iteration, kurze Feedbackschleifen, hohe Motivation durch Eigenverantwortung, direkte Kundenbeziehung.

Herausforderungen: Die kognitive Last darf nicht zu groß werden. Wenn der Wertstrom zu breit ist, bricht das Team zusammen.

Beispiel: Ein Team, das für die “Checkout-Experience” in einem E-Commerce verantwortlich ist – vom Warenkorb bis zur Bezahlbestätigung.

Verwandt mit: Product Team (Cagan) – im Kern ähnlich, aber Stream-Aligned betont die Struktur, Product Team die Kultur.

Die Fundamentleger (Platform Teams)

Stell dir vor, du baust ein Gewächshaus. Das Platform Team ist der Boden. Nicht sichtbar für den Kunden. Nicht glamourös. Aber ohne es kann nichts wachsen.

In der Architektur-Metaphor: Sie bauen das Fundament – was niemand sieht, aber ohne das nichts steht.

Charakteristika: Baut und betreibt eine interne Plattform. Stream-Aligned Teams sind seine “Kunden”. Self-Service – Teams können die Plattform ohne Wartezeiten nutzen. Fokus: Kognitive Last für andere Teams reduzieren.

Kernrollen: Platform Lead, Site Reliability Engineers (SRE), DevOps Engineers, Infrastructure Engineers, Security Engineers.

Disziplinarische Führung: Platform Lead kann disziplinarisch sein, oft aber Matrix.

In den Dimensionen: Unsicherheit niedrig bis mittel – das Terrain ist bekannt. Kopplung lose durch Self-Service – andere Teams nutzen die Plattform, ohne zu warten. Kritikalität mittel bis hoch – wenn die Plattform fällt, fallen alle. Wissen sitzt im Fach (Infrastruktur, DevOps, Security).

Wann die richtige Wahl: Mehrere Stream-Aligned Teams mit ähnlichen technischen Bedürfnissen. Wiederkehrende Infrastruktur-Aufgaben. Skalierung ist wichtig.

Stärken: Multiplikator für Produktivität. Ermöglicht Autonomie und Geschwindigkeit für viele Teams gleichzeitig.

Herausforderungen: Muss wirklich “Platform-Thinking” haben – die anderen Teams als Kunden sehen, nicht als Nutzer, denen man Befehle gibt.

Beispiel: Ein Team, das eine Kubernetes-Plattform betreibt mit CI/CD-Pipelines, Monitoring, Logging, Secret-Management – alles Self-Service.

Die Lehrmeister (Enabling Teams)

Das Enabling Team befähigt andere Teams. Es hilft ihnen, besser zu werden. Es teilt Wissen. Es gibt Impulse. Es macht sich selbst überflüssig – und das ist sein Erfolg.

In der Architektur-Metapher: Sie sind wie die Baumeister, die ihr Wissen weitergeben. Oder wie das Lehrgerüst beim Bau eines Steinbogens – es gibt Form und Halt, bis der Bogen sich selbst trägt.

Charakteristika: Spezialisiert auf eine bestimmte Fähigkeit oder Technologie. Arbeitet zeitlich begrenzt mit Stream-Aligned Teams. Ziel: Andere befähigen, nicht dauerhaft unterstützen. Erfolg wird gemessen an: Haben die anderen Teams gelernt?

Kernrollen: Coaches (DevOps Coach, Agile Coach, Security Coach), Technische Spezialisten, Trainer.

Disziplinarische Führung: Keine – sie sind Gäste in anderen Teams.

In den Dimensionen: Unsicherheit variiert, oft hoch bei neuen Technologien – sie bringen Licht ins Unbekannte. Kopplung temporär eng, dann entkoppelt – sie gehen wieder, wenn das Wissen transferiert ist. Kritikalität niedrig – es geht um Lernen, nicht um kritische Systeme. Wissen sitzt im Fach (das weitergegeben wird).

Wann die richtige Wahl: Neue Technologien oder Praktiken sollen eingeführt werden. Teams brauchen Hilfe, aber keine dauerhafte Abhängigkeit. Wissen muss transferiert werden.

Stärken: Verbreitet Wissen schneller als jede Dokumentation. Katalysator für organisationales Lernen.

Herausforderungen: Muss wissen, wann es gehen muss. Wenn es zu lange bleibt, werden die anderen Teams abhängig.

Beispiel: Ein DevOps-Enabling-Team, das für 6-8 Wochen mit einem Stream-Aligned Team arbeitet, um CI/CD-Praktiken zu etablieren – dann geht es weiter zum nächsten Team.

Die Spezialisten (Complicated-Subsystem Teams)

Manche Teile eines Systems sind so komplex, dass sie spezielle Expertise brauchen. Machine Learning. Payment-Systeme. Kryptographie. Das Complicated-Subsystem Team kapselt diese Komplexität – und stellt sie anderen als einfachen Service bereit.

In der Architektur-Metapher: Wie der Statiker für ein freitragendes Dach oder der Akustiker für einen Konzertsaal. Spezialwissen, das andere entlastet.

Charakteristika: Verantwortlich für ein komplexes Subsystem. Tiefe Spezialisierung notwendig. Stellt “Service” für Stream-Aligned Teams bereit. Langlebig. Fokus: Komplexität kapseln.

Kernrollen: Tech Lead, Spezialisten (ML Engineers, Payment Experts, Kryptographie-Experten, etc.).

Disziplinarische Führung: Tech Lead oft disziplinarisch.

In den Dimensionen: Unsicherheit mittel – das Gebiet ist speziell, nicht unbekannt. Kopplung lose via API – andere Teams konsumieren, ohne die Komplexität zu kennen. Kritikalität variiert – hängt vom Subsystem ab. Wissen sitzt im Fach (hochspezialisiert).

Wann die richtige Wahl: Es gibt Bereiche, die zu speziell für Generalisten sind. Die Komplexität würde Stream-Aligned Teams überlasten. Tiefes Expertenwissen nötig.

Stärken: Kapselung. Ein komplexes Problem wird einmal gelöst und dann von allen genutzt.

Herausforderungen: Muss den “Service”-Gedanken leben. Darf nicht zum Elfenbeinturm werden.

Beispiel: Ein Team, das die Empfehlungs-Engine baut. Andere Teams nutzen sie über eine API. Sie müssen nicht verstehen, wie die Algorithmen funktionieren.

Weitere Teammodelle

Die folgenden Modelle sind keine direkten Team-Topologies-Kategorien, aber wichtige Strukturformen, die du kennen solltest.

Integrated Product Teams (IPT)

Ursprünglich aus der Luft- und Raumfahrt. Ein IPT vereint alle Disziplinen, die für ein Produkt nötig sind – unter einem Dach, mit gemeinsamer Verantwortung.

Charakteristika: Alle Disziplinen integriert – Hardware, Software, Test, Qualität, Produktion. Gemeinsame Verantwortung für das Gesamtprodukt. Langlebig, oft über Jahre. Fokus: Komplexe, integrierte Systeme.

Kernrollen: IPT Lead, Systems Engineer, Vertreter jeder Disziplin (HW, SW, Test, etc.), Quality Assurance.

Disziplinarische Führung: Fachlich oft Matrix (Disziplin-Leads), operativ beim IPT Lead.

In den Dimensionen: Unsicherheit niedrig bis mittel – typisch für etablierte Branchen. Kopplung eng – alle Disziplinen müssen zusammenspielen. Kritikalität hoch – Fehler können Menschenleben kosten. Wissen sitzt in den Disziplinen (integriert).

Wann die richtige Wahl: Hardware-lastige Produkte mit vielen Disziplinen. Sicherheitskritische Branchen – Aerospace, Automotive, Medizintechnik. Lange Produktlebenszyklen.

Stärken: Tiefe Integration, frühes Erkennen von Schnittstellenproblemen.

Herausforderungen: Komplexe Koordination, kann schwerfällig werden.

Beispiel: Ein Team, das ein Flugzeug-Subsystem entwickelt – von der Spezifikation bis zur Zertifizierung.

Concurrent Engineering Teams

Ähnlich wie IPT, aber mit stärkerem Fokus auf parallele Entwicklung. Statt sequentiell – erst Design, dann Test – arbeiten alle Disziplinen gleichzeitig.

Charakteristika: Parallele Arbeit statt sequentielle Phasen. Intensive Kommunikation zwischen Disziplinen. Frühe Integration von Test und Produktion. Fokus: Time-to-Market reduzieren.

Kernrollen: Projektleiter, Disziplin-Experten (parallel arbeitend).

Disziplinarische Führung: Projektleiter oft disziplinarisch, manchmal Matrix.

In den Dimensionen: Unsicherheit mittel – der Weg ist bekannt, aber parallel. Kopplung eng – alle arbeiten gleichzeitig am selben Objekt. Kritikalität mittel bis hoch – Fehler kosten Zeit und Geld. Wissen sitzt in den Disziplinen (koordiniert).

Wann die richtige Wahl: Produktentwicklung mit Zeitdruck. Hardware-Software-Integration. Wenn sequentielle Phasen zu langsam sind.

Stärken: Schnellere Entwicklung, frühe Problemerkennung.

Herausforderungen: Erfordert hohe Abstimmungsdisziplin, kann chaotisch werden.

Agile Hardware Teams

Die Übertragung agiler Prinzipien auf Hardware-Entwicklung. Kurze Iterationen, frühe Prototypen, schnelles Feedback.

Charakteristika: Kurze Sprints auch bei Hardware. Frühe und häufige Prototypen. Cross-funktional mit HW und SW. Fokus: Lernen durch Bauen.

Kernrollen: HW Engineers, SW Engineers, Prototyping-Spezialisten, Testingenieure.

Disziplinarische Führung: Variiert, oft Matrix.

In den Dimensionen: Unsicherheit hoch – Iteration ist der Weg. Kopplung mittel – zwischen HW und SW. Kritikalität niedrig bis mittel – Prototypen dürfen scheitern. Wissen sitzt im Team (integriert).

Wann die richtige Wahl: Neue Hardware-Produkte mit unklaren Anforderungen. Startups und Innovation Labs. Wenn traditionelle Hardware-Entwicklung zu langsam ist.

Stärken: Schnelles Lernen, frühe Validierung.

Herausforderungen: Hardware-Iterationen kosten Geld, nicht alles ist “agil” machbar.

Value Stream Teams

Ein Team, das einen kompletten Wertstrom von Anfang bis Ende besitzt. Keine Übergaben, keine Silos.

Charakteristika: End-to-End-Verantwortung für einen Wertstrom. Alle Funktionen integriert, die für den Strom nötig sind. Fokus auf Flow und Durchlaufzeit. Fokus: Verschwendung eliminieren.

Kernrollen: Value Stream Manager, Prozess-Spezialisten, alle für den Strom nötigen Funktionen.

Disziplinarische Führung: Value Stream Manager oft disziplinarisch.

In den Dimensionen: Unsicherheit niedrig – der Strom ist bekannt. Kopplung lose nach außen – der Strom ist autark. Kritikalität variiert. Wissen sitzt im Prozess.

Wann die richtige Wahl: Klare, abgrenzbare Wertströme. Lean-orientierte Organisationen. Wenn Silos den Flow blockieren.

Stärken: Schneller Flow, klare Verantwortung.

Herausforderungen: Abgrenzung der Ströme kann schwierig sein.

Pods

Kleine, autonome Einheiten, die jeweils einen Kunden oder ein Kundensegment betreuen. Wie kleine Inseln.

Charakteristika: Klein – 5-10 Personen. Ein Kundensegment oder eine Kundengruppe. Alle Funktionen im Pod: Entwicklung, Support, Success. Fokus: Maximale Kundennähe.

Kernrollen: Pod Lead, Developer, Customer Success / Support, Product Manager (oft geteilt).

Disziplinarische Führung: Pod Lead oft disziplinarisch.

In den Dimensionen: Unsicherheit mittel – nah am Kunden, schnelles Feedback. Kopplung lose – Pods arbeiten unabhängig. Kritikalität variiert. Wissen sitzt beim Kunden (direkt).

Wann die richtige Wahl: B2B mit unterschiedlichen Kundenbedürfnissen. Wenn Kundennähe der Schlüssel ist. Agentur- oder Beratungsmodelle.

Stärken: Extreme Kundennähe, schnelle Reaktion.

Herausforderungen: Synergien zwischen Pods schwierig, kann zu Redundanz führen.

Cell-Based Teams

Kleine, selbstorganisierende Zellen mit hoher Autonomie. Inspiriert von biologischen Systemen.

Charakteristika: Sehr klein – 3-7 Personen. Hohe Autonomie. Selbstorganisierend. Fokus: Schnelle Anpassung.

Kernrollen: Cell Lead (oft rotierend), Spezialisten je nach Aufgabe.

Disziplinarische Führung: Cell Lead kann disziplinarisch sein, oft aber selbstorganisiert.

In den Dimensionen: Unsicherheit hoch – Zellen passen sich an. Kopplung lose – Zellen arbeiten unabhängig. Kritikalität niedrig – Experimente sind erlaubt. Wissen sitzt im Team (verteilt).

Wann die richtige Wahl: Hohe Unsicherheit, schnelle Veränderung. Teams mit hoher Reife und Selbstorganisation. Innovation und Exploration.

Stärken: Maximale Anpassungsfähigkeit, hohe Motivation.

Herausforderungen: Braucht reife Menschen, kann ohne Struktur chaotisch werden.

Project-based Squads

Teams, die für ein spezifisches Projekt zusammengestellt werden. Zeitlich begrenzt, zielgerichtet.

Charakteristika: Für ein Projekt gebildet. Cross-funktional zusammengestellt. Zeitlich begrenzt. Fokus: Projektziel erreichen.

Kernrollen: Projektleiter, Spezialisten (je nach Projektbedarf).

Disziplinarische Führung: Funktional – Teammitglieder bleiben in ihrer Linie.

In den Dimensionen: Unsicherheit variiert. Kopplung eng innerhalb, lose nach außen. Kritikalität variiert. Wissen sitzt im Projekt.

Wann die richtige Wahl: Klar definierte Projekte mit Ende. Wenn dauerhafte Teams nicht sinnvoll sind. Transformationen, Migrationen.

Stärken: Fokussiert, zielgerichtet.

Herausforderungen: Wissen geht nach Projektende verloren, Team-Building für kurze Zeit.

Holacracy / Soziokratie Teams

Selbstorganisation als Prinzip. Keine klassischen Hierarchien, sondern Rollen und Kreise.

Charakteristika: Rollen statt Positionen. Kreise statt Abteilungen. Governance-Prozesse für Entscheidungen. Fokus: Spannungen produktiv nutzen.

Kernrollen: Rollen (definiert durch die Arbeit, nicht die Person), Facilitator (moderiert Meetings), Secretary (dokumentiert).

Disziplinarische Führung: Keine klassische. Verantwortung liegt in den Rollen.

In den Dimensionen: Unsicherheit hoch – das System ist darauf ausgelegt. Kopplung variiert – Kreise können eng oder lose sein. Kritikalität niedrig bis mittel – braucht psychologische Sicherheit. Wissen sitzt in den Rollen (verteilt).

Wann die richtige Wahl: Organisationen mit hoher Reife. Wenn klassische Hierarchien hindern. Wenn Selbstorganisation gelebt werden soll.

Stärken: Maximale Anpassungsfähigkeit, verteilte Verantwortung.

Herausforderungen: Steile Lernkurve, kann ohne Reife scheitern.

Mission Command Teams

Inspiriert vom Militär. Der Führende gibt das “Was” vor, die Teams entscheiden über das “Wie”.

Charakteristika: Klare Absicht (Commanders Intent). Dezentrale Ausführung. Teams haben Autonomie im “Wie”. Fokus: Schnelle Entscheidungen vor Ort.

Kernrollen: Commander (gibt Absicht vor), Team Leads (setzen um), Operateure (handeln).

Disziplinarische Führung: Commander disziplinarisch.

In den Dimensionen: Unsicherheit hoch – das Terrain ist unbekannt. Kopplung lose – Teams agieren autonom. Kritikalität hoch – Fehler können schwerwiegend sein. Wissen sitzt vor Ort (dezentral).

Wann die richtige Wahl: Krisensituationen. Schnelle Entscheidungen nötig. Wenn zentrale Steuerung zu langsam ist.

Stärken: Schnell, anpassungsfähig, nah an der Realität.

Herausforderungen: Braucht Vertrauen und Kompetenz auf allen Ebenen.

Beispiel: General McChrystals Transformation der Joint Special Operations Task Force im Irak – von strikter Hierarchie zu vernetzten, autonomen Teams.[254]

Multi-Disciplinary Care Teams

Teams im Gesundheitswesen, die einen Patienten ganzheitlich betreuen. Alle relevanten Disziplinen arbeiten zusammen.

Charakteristika: Alle Disziplinen um den Patienten herum. Gemeinsame Fallbesprechungen. Ganzheitliche Verantwortung. Fokus: Der Mensch im Zentrum.

Kernrollen: Ärzte (verschiedene Fachrichtungen), Pflege, Therapeuten, Sozialarbeit, Case Management.

Disziplinarische Führung: In der Fach-Linie (Ärzte zu Ärzten, Pflege zu Pflege).

In den Dimensionen: Unsicherheit mittel – Medizin hat Standards, aber jeder Patient ist anders. Kopplung eng – alle Disziplinen müssen zusammenarbeiten. Kritikalität hoch – es geht um Menschenleben. Wissen sitzt beim Patienten und in den Disziplinen.

Wann die richtige Wahl: Komplexe Patientenfälle. Chronische Erkrankungen. Wenn Silo-Medizin schadet.

Stärken: Ganzheitliche Betreuung, weniger Lücken.

Herausforderungen: Koordination kostet Zeit, unterschiedliche Kulturen der Disziplinen.

Funktionale Teams

Die klassische Struktur: Teams, die nach Fachgebiet gruppiert sind. Marketing, Entwicklung, Vertrieb.

Charakteristika: Gruppierung nach Expertise. Klare Spezialisierung. Stabile Strukturen. Fokus: Fachliche Tiefe.

Kernrollen: Abteilungsleiter, Fachspezialisten.

Disziplinarische Führung: Abteilungsleiter.

In den Dimensionen: Unsicherheit niedrig – das Fachgebiet ist bekannt. Kopplung eng zwischen Abteilungen – Projekte erfordern Koordination. Kritikalität variiert. Wissen sitzt im Fachgebiet.

Wann die richtige Wahl: Junge Organisationen, die Struktur brauchen. Wenn fachliche Tiefe wichtiger ist als Geschwindigkeit. Stabile Umfelder.

Stärken: Klare Expertise, klare Karrierewege, Stabilität.

Herausforderungen: Silos, lange Wege bei Querschnittsthemen, langsame Reaktion auf Markt.

Skalierungsstrukturen

Skalierungsstrukturen koordinieren mehrere Teams. Sie sind keine Teammodelle, sondern Architekturen für das Zusammenspiel.

Das Spotify-Modell

Squads, Tribes, Chapters, Guilds. Eine Matrix, die Autonomie und Alignment verbindet.

Elemente: Squads – kleine, autonome Teams (wie Stream-Aligned Teams). Tribes – mehrere Squads, die an verwandten Bereichen arbeiten. Chapters – fachliche Heimat (z.B. alle Backend-Developer). Guilds – informelle Communities of Practice.

Charakteristika: Matrix aus Produkt (Squad) und Fach (Chapter). Autonomie auf Squad-Ebene. Alignment durch Tribes und Guilds. Fokus: Skalierte Agilität.

Kernrollen: Product Owner (im Squad), Chapter Lead (für Disziplin), Tribe Lead (für Alignment).

Disziplinarische Führung: Chapter Lead für die Disziplin.

In den Dimensionen: Unsicherheit mittel bis hoch – Squads können experimentieren. Kopplung mittel – Squads arbeiten unabhängig, aber im Tribe abgestimmt. Kritikalität mittel – Autonomie mit Leitplanken. Wissen sitzt im Squad (nah am Produkt) und im Chapter (fachlich).

Wann die richtige Wahl: Mittlere bis große Tech-Organisationen. Mehrere Produkte oder Produktbereiche. Wenn Agilität skaliert werden soll.

Stärken: Balance aus Autonomie und Alignment, fachliche Entwicklung.

Herausforderungen: Kann zur Bürokratie werden, Tribes können zu Silos werden.

Network of Teams

Inspiriert von McChrystal. Autonome Teams, verbunden durch “Shared Consciousness” – gemeinsames Situationsbewusstsein.

Elemente: Autonome Teams – dezentrale Entscheidungen. Shared Consciousness – alle haben das gleiche Lagebild. Extreme Transparenz – Information fließt frei. Liaisons – Verbindungspersonen zwischen Teams.

Charakteristika: Netzwerk statt Hierarchie. Information als Bindeglied. Schnelle Anpassung. Fokus: Komplexe, volatile Umfelder.

Kernrollen: Team Lead (im Team), Liaison (zwischen Teams).

Disziplinarische Führung: Team Lead disziplinarisch.

In den Dimensionen: Unsicherheit hoch – das Netzwerk ist darauf ausgelegt. Kopplung lose strukturell, eng informationell – Teams sind unabhängig, aber informiert. Kritikalität hoch – oft in Krisensituationen eingesetzt. Wissen sitzt überall (geteilt).

Wann die richtige Wahl: Krisensituationen. Schnell wechselnde Umfelder. Wenn klassische Hierarchien zu langsam sind.

Stärken: Extreme Anpassungsfähigkeit, schnelle Reaktion.

Herausforderungen: Braucht Vertrauen und Reife, hoher Kommunikationsaufwand.

SAFe / LeSS / Nexus

Frameworks für skalierte Agilität. Mehrere Teams arbeiten an einem gemeinsamen Produkt.

Elemente (am Beispiel SAFe): Agile Release Train (ART) – mehrere Teams mit gemeinsamer Kadenz. Program Increment (PI) – Planungszyklus, meist 8-12 Wochen. Solution Train – Koordination mehrerer ARTs für große Lösungen.

Charakteristika: Portfolio-Management. Gemeinsame Kadenz (z.B. Program Increments). Synchronisierung vieler Teams. Klare Rollen und Artefakte.

Kernrollen (SAFe): Release Train Engineer, Product Management, System Architect, Scrum Masters / Team Leads.

Disziplinarische Führung: Variiert – oft in der Linie.

In den Dimensionen: Unsicherheit niedrig bis mittel – Strukturen geben Halt. Kopplung eng – viele Teams, ein Produkt, intensive Koordination nötig. Kritikalität mittel – Fehler kosten Zeit und Geld. Wissen sitzt im Prozess und Framework.

Wann die richtige Wahl: Viele Teams müssen koordiniert werden. Gemeinsame Releases nötig. Mittlere kulturelle Reife. Struktur fehlt.

Stärken: Gibt Struktur, wo Chaos droht. Klare Verantwortlichkeiten.

Herausforderungen: Kann Agilität ersticken, wenn mechanisch angewendet. Oft zu bürokratisch.

Beispiel: Eine Bank, die 40 Teams in einem SAFe Release Train bündelt.

Rollen und Führung – Gesamtübersicht

Diese Tabelle dient als schnelle Referenz. Sie zeigt, welche Kernrollen die verschiedenen Teammodelle haben und wie die disziplinarische Führung geregelt ist.

Teammodelle

Team	Kernrollen	Disziplinarische Führung
Stream-Aligned	PO, Developers, QA, UX	Keine im Team – Matrix/Chapters
Platform	Platform Lead, SRE, DevOps, Infra, Security	Platform Lead (oft Matrix)
Enabling	Coaches, Spezialisten	Keine – sie sind Gäste
Complicated-Subsystem	Tech Lead, Spezialisten	Tech Lead disziplinarisch
IPT	IPT Lead, Systems Engineer, Disziplin-Vertreter	Disziplin-Leads disziplinarisch
Concurrent Engineering	Projektleiter, Disziplin-Experten	Projektleiter oft disziplinarisch
Agile Hardware	HW/SW Engineers, Prototyping	Variiert (Matrix)
Value Stream	Value Stream Manager, Prozess-Spezialisten	VSM oft disziplinarisch
Pods	Pod Lead, Developer, PM, Support	Pod Lead disziplinarisch
Cell-Based	Cell Lead, Spezialisten	Cell Lead disziplinarisch
Project Squads	Projektleiter, Spezialisten	Funktional, disziplinarisch in Linie
Holacracy	Rollen, Facilitator, Secretary	Keine klassische
Mission Command	Commander, Team Leads	Commander disziplinarisch
Care Teams	Ärzte, Pflege, Therapeuten, Sozialarbeit	In der Fach-Linie
Funktional	Abteilungsleiter, Fachspezialisten	Abteilungsleiter

Skalierungsstrukturen

Struktur	Kernrollen	Disziplinarische Führung
Spotify-Modell	PO, Developers, Designer (im Squad)	Chapter Lead für Disziplin
Network of Teams	Team Lead, Operateure (im Team)	Team Lead disziplinarisch
SAFe/LeSS	RTE, Product Management, System Architect	In der Linie

Entscheidungsmatrizen – Schnelle Orientierung

Diese Tabellen sind nicht zum Durchlesen gedacht. Sie sind Orientierungshilfen für den Moment der Entscheidung.

Nutze sie so: Frag dich zuerst, was deine konkrete Situation ist. Schau dann in die Tabelle – welche Modelle passen grob? Lies die Detail-Beschreibung der passenden Modelle oben. Und entscheide mit Kontext – die Matrizen sind Kompass, keine Landkarte.

Nach Situation

Deine Situation	Passende Teammodelle	Passende Skalierungsstruktur
Digitales Produkt bauen	Stream-Aligned + Platform Teams	—
Viele Teams koordinieren	—	SAFe / LeSS / Nexus, Spotify-Modell
Sicherheitskritische Branche	IPT, Concurrent Engineering	—
Radikal kundenorientiert sein	Pods, Cell-Based Teams	—
Schnell auf Marktveränderungen reagieren	Agile Hardware, Mission Command	Network of Teams, Spotify-Modell
Zeitlich begrenztes Vorhaben	Project-based Squads	—
Teams technisch weiterentwickeln	Enabling Teams im Facilitating-Modus	—
Struktur in junger Organisation	Funktionale Teams	—
Maximale Selbstorganisation	Holacracy / Soziokratie Teams	—
Krisensituationen	Mission Command Teams	Network of Teams
Hardware mit kurzen Zyklen	Agile Hardware Teams	—
Ganzheitliche Betreuung von Menschen	Multi-Disciplinary Care Teams	—

Nach den drei Dimensionen

Dimension	Ausprägung	Passende Teammodelle	Passende Skalierungsstruktur
Unsicherheit	Hoch	Agile HW, Mission Command, Holacracy	Spotify, Network of Teams
	Niedrig	Value Stream, IPT, Funktional	SAFe
Kopplung	Eng	IPT, Concurrent Engineering	Collaboration-Modus

Dimension	Ausprägung	Passende Teammodelle	Passende Skalierungs-struktur
	Lose	Stream-Aligned, Pods, Cell-Based	X-as-a-Service-Modus
Kritikalität	Hoch	IPT, Value Stream, Care Teams	Network of Teams
	Niedrig	Agile HW, Cell-Based, Ho-lacracy	Spotify

Nach kultureller Reife

Reife	Passende Teammodelle	Passende Skalierungsstruktur
Nied-rig	Funktionale Teams, klare Hierarchien	SAFe
Mittel	Stream-Aligned + Platform, Pods, Project-based Squads	LeSS
Hoch	Holacracy, Cell-Based, Mission Command	Spotify-Modell, Network of Teams

Die Diagnose-Fragen

Wenn du vor einer Strukturentscheidung stehst:

Wo sitzt das Wissen? Beim Kunden → Stream-Aligned / Divisional / Pods. Im Fachgebiet → Funktional / Complicated-Subsystem. Im Prozess → Platform / Value Stream.

Wie hoch ist die Unsicherheit? Hoch → Autonome Teams, kurze Zyklen (Cell-Based, Mission Command, Holacracy). Niedrig → Definierte Prozesse, klare Handoffs (IPT, Value Stream, Funktional).

Wie eng die Kopplung? Lose → Teams können parallel arbeiten (X-as-a-Service-Modus). Eng → Investiere in Kommunikation oder entkopple (Collaboration-Modus, IPT).

Wie kritisch die Fehler? Hoch → Mehr Absicherung, mehr Review (IPT, Care Teams). Niedrig → Schneller iterieren, aus Fehlern lernen (Agile HW, Cell-Based).

Wie reif ist die Organisation kulturell? Niedrig → Strukturierte Frameworks, klare Rollen (SAFe, Funktional). Hoch → Mehr Selbstorganisation möglich (Holacracy, Network of Teams).

Schlussgedanke

Die Frage ist nie: Welches Team ist das beste?

Die Frage ist immer: Welchen Baustil braucht unser Gebäude?

Dieser Katalog ist dein Kompass. Nutze ihn, wenn du vor einer Entscheidung stehst. Aber vergiss nie: Teammodelle sind Werkzeuge. Sie sind keine Identität. Keine Ideologie. Kein Selbstzweck.

Die Reife einer Organisation erkennt man daran, dass sie verschiedene Formen erkennt – und beherzt die wählt, die für ihre Aufgabe jetzt am besten ist.

Menschen sind zum Geliebtwerden geschaffen, nicht zum Funktionieren.
Und wenn wir Organisationen als Gewächshäuser gestalten
– Räume, die schützen, nähren, ermöglichen –
dann kann etwas wachsen, das größer ist als wir selbst.

KAVANARA
CONSULTING · TECHNICALS · SOLUTIONS

Mehr erfahren: **kavanara.de**