



Jürgen Hoffmann • Stefan Rook

Agile Unternehmen

Veränderungsprozesse gestalten,
agile Prinzipien verankern,
Selbstorganisation
und neue Führungsstile etablieren



dpunkt.verlag



Inhalt

Cover

Über den Autor

Titel

Impressum

Widmung

Inhaltsübersicht

Inhaltsverzeichnis

1 Einleitung

1.1 Echte Agilität

1.2 Agile Fluency

1.3 Fokus dieses Buches: echte Agilität oder auch »Optimize Value«

1.3.1 Eigenschaften von »Optimize Value«-Unternehmen

1.3.2 »Focus on Value« und »Deliver Value«: Literaturempfehlungen

1.4 An wen richtet sich das Buch?

1.5 Überblick über das Buch

1.6 Danksagung

2 Begeisterte Kunden

2.1 Definieren, was Wert bedeutet und schafft

2.1.1 Wert aus Kundensicht

2.1.2 Bedürfnisse identifizieren

2.2 Drei Horizonte für Wachstum und Innovation

2.2.1 Herausforderungen bei der Umsetzung des 3-Horizonte-Modells

2.2.2 Das 3-Horizonte-Modell und agile Entwicklung

2.2.3 Wert bedeutet in jedem Horizont etwas anderes

2.3 Wert in Horizont 1

2.3.1 Umsatz als Indikator für Wertschöpfung

2.3.2 Net Promoter System (NPS)

2.3.3 NPS: Bitte beachten

2.3.4 Produktreview/Sprint-Review

2.4 Wert in Horizont 2

2.4.1 Produktvision

2.4.2 Produktreview/Sprint-Review

2.4.3 Design Sprints

2.5 Wert in Horizont 3

2.5.1 Vorgehen in Horizont 3 zur Produkt-/Serviceentwicklung

2.5.2 Konkrete Techniken zum Einsatz in Horizont 3

2.6 Organisation für das 3-Horizonte-Modell

2.6.1 Freiraum in Horizont 1

2.6.2 Freiraum in Horizont 2

2.6.3 Freiraum in Horizont 3

2.6.4 Übergang von Horizont 3 nach Horizont 2

2.6.5 Übergang von Horizont 2 nach Horizont 1

2.6.6 Personalstrategien der Horizonte

2.6.7 Entwicklung in den drei Horizonten

2.6.8 Produkt-Roadmaps in den drei Horizonten

2.7 Das Kapitel in Stichpunkten

3 Wertschöpfung als Teamaufgabe

3.1 Eigenständige Teams

3.1.1 Manager-led Teams

3.1.2 Self-managing Teams

3.1.3 Self-designing Teams

3.1.4 Self-governing Teams

3.2 Funktionsübergreifende Teams

3.2.1 Zusammensetzung von Teams

3.2.2 Product-Owner-Rolle

3.2.3 Teambegleitung

3.2.4 Effizienz vs. Effektivität

3.3 Entscheidungen im Team

3.4 Das Kapitel in Stichpunkten

4 Unterstützende Organisation

4.1 Störungen durch das Unternehmen

4.2 Dezentrale Strukturen

4.2.1 Zellmodell in der Praxis der Softwareentwicklung

4.2.2 Mehr als ein Team pro Zelle

4.2.3 Alles Illusion?

4.3 Alignment bei dezentralen Strukturen

4.3.1 Management by Objectives (MbO)

4.3.2 Objectives and Key Results (OKR)

4.3.3 MbO-Beispiel – so bitte nicht

4.3.4 MbO-Beispiel – besser

4.3.5 Nutzen und Gefahren von Management by Objectives

4.3.6 Ziele ohne die MbO-Gefahren

4.4 Feedbackschleifen statt statischer Ziele

4.4.1 Feedbackschleife für den Umweltschutz

4.4.2 Feedbackschleifen bei Command & Control-Strukturen

4.4.3 Feedbackschleifen in einem agilen Unternehmen

4.4.4 Das Unternehmen als Organismus

4.5 Übergreifende Entscheidungsfindung bei dezentralen Strukturen

4.5.1 Konsent

4.5.2 Advice-Prozess

4.5.3 Das Unternehmen verstehen

4.5.4 Bewertung und Vergleich von Konsent und Advice-Prozess

4.6 Neue Rolle für Führungskräfte

4.6.1 Klassische Mitarbeiterführung

4.6.2 Probleme klassischer Führung in einer dynamischen Welt

4.6.3 Supporting Lines statt Reporting Lines

4.6.4 Verteilte Führung

4.6.5 Situative Führung

4.6.6 Ausbildung

4.7 Fallbeispiele zu moderner Mitarbeiterführung

4.7.1 ImmobilienScout24

4.7.2 siggate

4.7.3 it-agile

4.7.4 Zusammenfassung der Fallbeispiele für Mitarbeiterführung

4.8 Unternehmenskultur

4.8.1 Unternehmenskultur und agiles Arbeiten

4.9 Das Kapitel in Stichworten

5 Organisationsentwicklung

5.1 Organisationsentwicklung als komplexe Aufgabe

5.1.1 Satir Change Model

5.2 Erfolgsfaktoren für agile Organisationsentwicklung

5.2.1 Erfahrungen mit dem Kotter Change Model

5.3 Steuerung iterativer Organisationsentwicklung

5.3.1 Das agile Transitionsteam

5.3.2 Transition Backlog und Product Owner

5.3.3 Produktvision und Produktinkremente des Transitionsteams

5.3.4 Transitionsteam: Besetzung und Rollen

5.3.5 Sprints im Transitionsteam

5.3.6 Einbindung ins Unternehmen

5.3.7 Weitere Probleme im Transitionsteam

5.4 Organisationsentwicklung über Experimente

5.4.1 Der PDCA-Zyklus

5.4.2 PDCA in der Praxis

5.4.3 Organisationsentwicklung als Abfolge von Experimenten

5.4.4 Safe-to-Fail-Experimente

5.4.5 Experimente erleichtern die Veränderung

5.4.6 Organisation der Organisationsentwicklung

5.5 Kultur der kontinuierlichen Verbesserung

5.5.1 Transparenz in alle Richtungen

5.6 Orientierung mit einem Nordstern (True North)

5.6.1 Nordstern bei Toyota

5.6.2 Nordsterne für die Wissensarbeit

5.6.3 Eigenschaften eines guten Nordsterns

5.6.4 Arbeiten mit dem Nordstern

5.6.5 Nordstern und der PDCA-Zyklus

5.6.6 Die A3-Technik

5.6.7 Der Weg zum eigenen Nordstern

5.7 Das Kapitel in Stichworten

Anhang

A User Research

A.1 Design Thinking konkret

A.1.1 Team

A.1.2 Raum

A.1.3 Prozess

A.2 Design Sprints

A.3 Lean Startup

A.3.1 Die Historie und das Umfeld

A.3.2 Kundenbedürfnisse verstehen und Lösung validieren

A.3.3 Den Markt validieren

A.3.4 Minimum Viable Product (MVP)

A.3.5 Pivots

A.3.6 Skalierung

A.3.7 Fallbeispiel bei it-agile

A.3.8 Fazit zu Lean Startup

A.4 Das Kapitel in Stichworten

B Große Produkte mit dem LeSS-Framework entwickeln

B.1 Veränderung folgt Notwendigkeiten

B.2 Agile Skalierungsprinzipien nach LeSS

B.3 Durchstarten zur Skalierung

B.3.1 Schule alle Beteiligten

B.3.2 Definiere das »Produkt«

B.3.3 Definiere, wann es »fertig« ist

B.3.4 Baue angemessen strukturierte Teams auf

B.3.5 Nur der Product Owner versorgt die Teams mit Arbeit

B.4 Ein Produkt – mehrere Teams

B.5 Das Kapitel in Stichworten

Literaturverzeichnis

Index

4 Unterstützende Organisation

Bei der 2012 Pleite gegangenen Firma Kodak hatte schon 1974 ein 24-jähriger Ingenieur die erste Digitalkamera entwickelt. Die etablierte Organisation sah die Gefahr für ihr Filmgeschäft und verbannte das neue Gerät in den Giftschränk. Kodak verdiente an dem Patent Geld – aber den Wechsel hin zu einem überlebensfähigen Hersteller von Digitalkameras schaffte das Unternehmen trotz des Wissensvorsprungs nicht.¹

Etablierte Organisationen stehen ihren frechen Kindern extrem feindlich gegenüber. Insbesondere wenn die Horizont-3-Idee das Potenzial in sich trägt, die wirtschaftliche Basis in Horizont 1 anzugreifen, wie im Kodak-Beispiel. Die spezifischen Herausforderungen zur Organisation der drei Horizonte haben wir in Kapitel 2 bereits ausführlich dargestellt. Wie dort ausgeführt, gibt es gute Gründe, innerhalb desselben Unternehmens verschiedene Führungs- und Organisationskulturen zuzulassen. »Alles muss gleich sein« kann sich als fatale Schwäche erweisen.

Zusammengefasst darf die Organisation die Arbeit der Teams nicht stören, sondern muss sie geeignet unterstützen. Zielsysteme, Bonussysteme, Aufgaben und Verantwortungen von Führungskräften sowie Roadmap- und Portfolioplanung müssen sich der Wertschöpfung für den Kunden unterordnen.

Außerdem bekommen Führungskräfte in einer agilen Umgebung eine wichtige Rolle als Lehrer und Enabler. Als Führungskraft dient man dabei dem Unternehmen und den Mitarbeitern, indem man jeweils das Beste unternimmt, um den Erfolg – die Wertschöpfung – des Gesamtunternehmens zu unterstützen.

4.1 Störungen durch das Unternehmen

Wir hatten den agilen Kernzyklus identifiziert: ein autonomes, selbstorganisiertes Team entwickelt Lösungen für Endkunden (siehe Abb. 4-1).

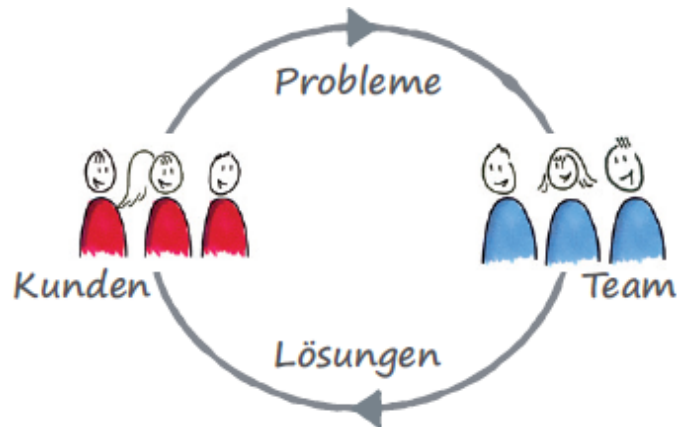


Abb. 4-1 Agiler Kernzyklus

So einfach und trivial dieser Kernzyklus aussieht, so schwierig ist er in der Praxis häufig umzusetzen. Faktisch sind die meisten Unternehmen sehr weit von diesem Ideal entfernt. Das liegt keineswegs an Naturgesetzen, die diesen Zyklus schwer umsetzbar machen. Vielmehr haben sich die meisten Unternehmen Strukturen und Prozesse gegeben, die den agilen Kernzyklus stören. Oder mit anderen Worten: Unternehmen haben sich so strukturiert, dass sie die eigene Wertschöpfung behindern!

Im agilen Kernzyklus finden sich diese Störungen in allen drei relevanten Bereichen (siehe Abb. 4-2):

- **Störungen bei Problemen**

Das Team darf nicht direkt mit den Endkunden interagieren und wird so gar nicht mit den kundenrelevanten Problemen konfrontiert. Stattdessen versuchen »Experten« die Probleme der Endkunden zu verstehen und daraus Anforderungen für das Team zu entwickeln, das diese »nur« noch umsetzen muss.

- **Störungen bei Lösungen**

Das Team darf oder kann seine Lösungen nicht direkt an Endkunden ausliefern. Es steht nicht die erforderliche Infrastruktur zur Verfügung, damit das Team direkt alle notwendigen qualitätssichernden Maßnahmen (vor allem: Tests) durchführen kann. Und selbst wenn die Infrastruktur vorhanden wäre, dürfte das Team nicht ohne vorherige Prüfung durch andere an die Endkunden ausliefern. Außerdem darf das Team den Kunden keine Vorabversionen oder Prototypen zeigen, um frühzeitig Feedback einzuholen.

- **Störungen im Team**

Und schließlich haben es viele Unternehmen zur echten Meisterschaft gebracht, ihre Mitarbeiter von wertschöpfender Arbeit abzuhalten. Stattdessen verbringen die Mitarbeiter viel Zeit in Meetings, warten auf Entscheidungen von Vorgesetzten, folgen ihren individuellen Zielen statt den Teamzielen etc. Diese Behinderung der wertschöpfenden Arbeit auf individueller Ebene potenziert sich bei Teamarbeit. Durch die gegenseitige Abhängigkeit im Team können die Störungen der individuellen Arbeit zu einem vollständigen Erliegen der Teamarbeit führen.

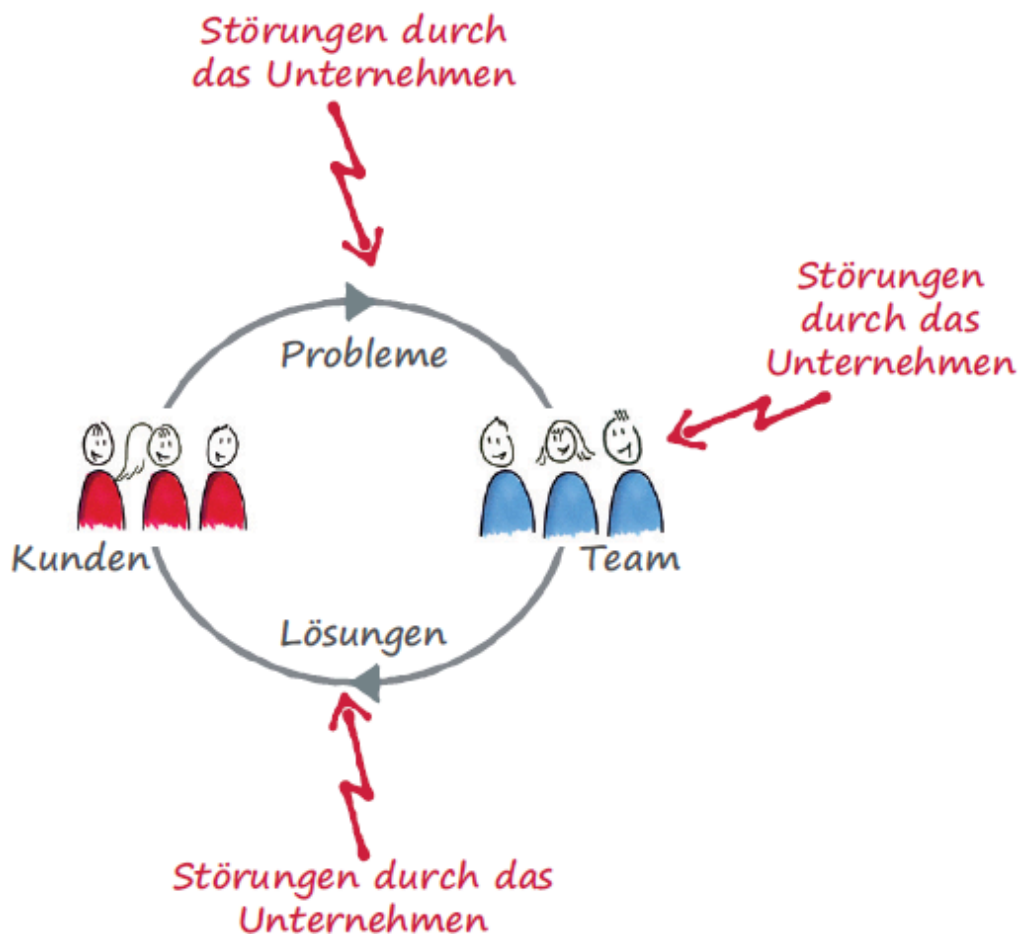


Abb. 4-2 Unternehmen stören den agilen Kernzyklus.

Scrum nennt diese Störungen Hindernisse (engl. *Impediments*). Werden diese Störungen systematisch durch Strukturen des Unternehmens erzeugt, spricht man von *organisatorischen Dysfunktionen*.

Für effektives agiles Arbeiten ist es essenziell, diese Störungen so weit wie möglich zu beseitigen oder zumindest zu dämpfen.

4.2 Dezentrale Strukturen

Autonome, selbstorganisierte Teams zielen offensichtlich auf eine dezentrale Struktur ab: Möglichst viele Entscheidungen sollen dezentral in Teams getroffen werden. Diese Idee ist keineswegs neu; auch nicht auf der Ebene von Unternehmensstrukturen.

Peter Drucker beschrieb bereits 1954 dezentrale Entscheidungen als essenziell für jedes Unternehmen (siehe [Drucker 1954]).

»[...] functional centralization is impossible. They require the closest co-operation of people from all functions at every stage. They require that design, production, marketing and the organization of the work be tackled simultaneously by a team representing all functions. They require that every member of the team both know his own functional work and see the impact on the whole business all the time. And decisions affecting the business as a whole have to be taken at a decentralized level [...]« (Position 1847, Kindle [Drucker 1954]).

Damit Entscheidungen wirklich dezentral getroffen werden können, müssen Unternehmen sich so strukturieren, dass ihre Einheiten möglichst unabhängig voneinander sind. Sonst müssen ständig Entscheidungen der einen Einheit mit denen anderer Einheiten abgestimmt werden.

Zuerst sollten laut Drucker Unternehmen nach Märkten bzw. Produkten organisiert sein. Wer Lkws und Pkws herstellt, sollte auf oberster Ebene die Unternehmenseinheiten Lkw und Pkw haben und nicht Einkauf, Verkauf, Produktion etc. Drucker räumt ein, dass mit dieser Art der Strukturierung mitunter nicht das ganze Unternehmen organisiert werden kann (die entstehenden Einheiten sind mitunter zu groß, um sie ohne weitere Struktur organisieren zu können). Daher schlägt er vor, unterhalb der Märkte/Produkte nach Phasen (engl. stages) zu organisieren. Phasen sind Schritte der Wertschöpfung, die zwei Kriterien erfüllen sollen:

1. Es gibt wenige Abhängigkeiten zwischen den Phasen.
2. Das Ergebnis einer Phase kann sehr lange liegen bleiben, bis die Folgephase mit dem Ergebnis weitermacht und durch diese Liegezeiten tritt kein Wertverlust ein. (Was allerdings nicht bedeutet, dass lange Liegezeiten wünschenswert wären.)
3. Das Ergebnis einer Phase kann qualitätsgesichert werden und stellt einen validierten Fortschritt in Richtung des Gesamtergebnisses dar.

Die Produktion eines Pkw-Motors ist demnach eine valide Phase. Der Motor kann qualitätsgesichert produziert werden, ist offensichtlich ein wichtiger Schritt in Richtung eines kompletten Autos, kann längere Zeit im Lager liegen und die Motorproduktion kann mit wenig Abhängigkeiten zur Produktion von Kotflügeln organisiert werden.

Die heute in Unternehmen häufig vorzufindende Organisation nach funktionalen Silos (Vertrieb, Einkauf, Produktion, Qualitätssicherung etc.) mag phasenartig aussehen, ist aber nicht das, was Drucker als Phase verstand. Hat der Vertrieb beispielsweise einen Verkaufsabschluss erreicht, verliert dieser mit zunehmender Liegezeit an Wert (irgendwann wird der Kunde ungeduldig und storniert den Auftrag). Bei der Qualitätssicherung besteht das Problem, dass diese nicht am Ende die Qualität in das Produkt hineintesten kann. Qualitätssicherung muss die ganze Zeit über stattfinden. In Phasen gedacht würde man vielleicht Motor, Kotflügel etc. als Organisationseinheiten definieren und die Qualitätssicherung wäre jeweils integriert.

Niels Pfläging schlägt mit dem Zellmodell ein konkretes Organisationsmodell vor, das Druckers Ideen aufgreift (siehe [Pfläging 2014]). Er baut Unternehmensstrukturen ausgehend vom Markt auf. Zuerst werden Kundengruppen identifiziert, deren Probleme gelöst werden sollen. Passend zu diesen Kundengruppen werden Organisationseinheiten definiert, die die Probleme ihrer Kunden autonom lösen. Pfläging nennt diese Organisationseinheiten *Peripheriezellen*. An der Benennung lässt sich erkennen, dass Pfläging Organisationsstrukturen nicht als »in Stein gemeißelt« ansieht, sondern Unternehmen als Organismen begreift. Es bilden sich dort Zellen, wo es sinnvoll ist, und wo eine Zelle nicht mehr benötigt wird, stirbt sie ab.

Diese Sichtweise passt hervorragend zu unserem agilen Kernzyklus. Ein auf dieser Basis organisiertes Unternehmen sähe wie in Abbildung 4-3 dargestellt aus.

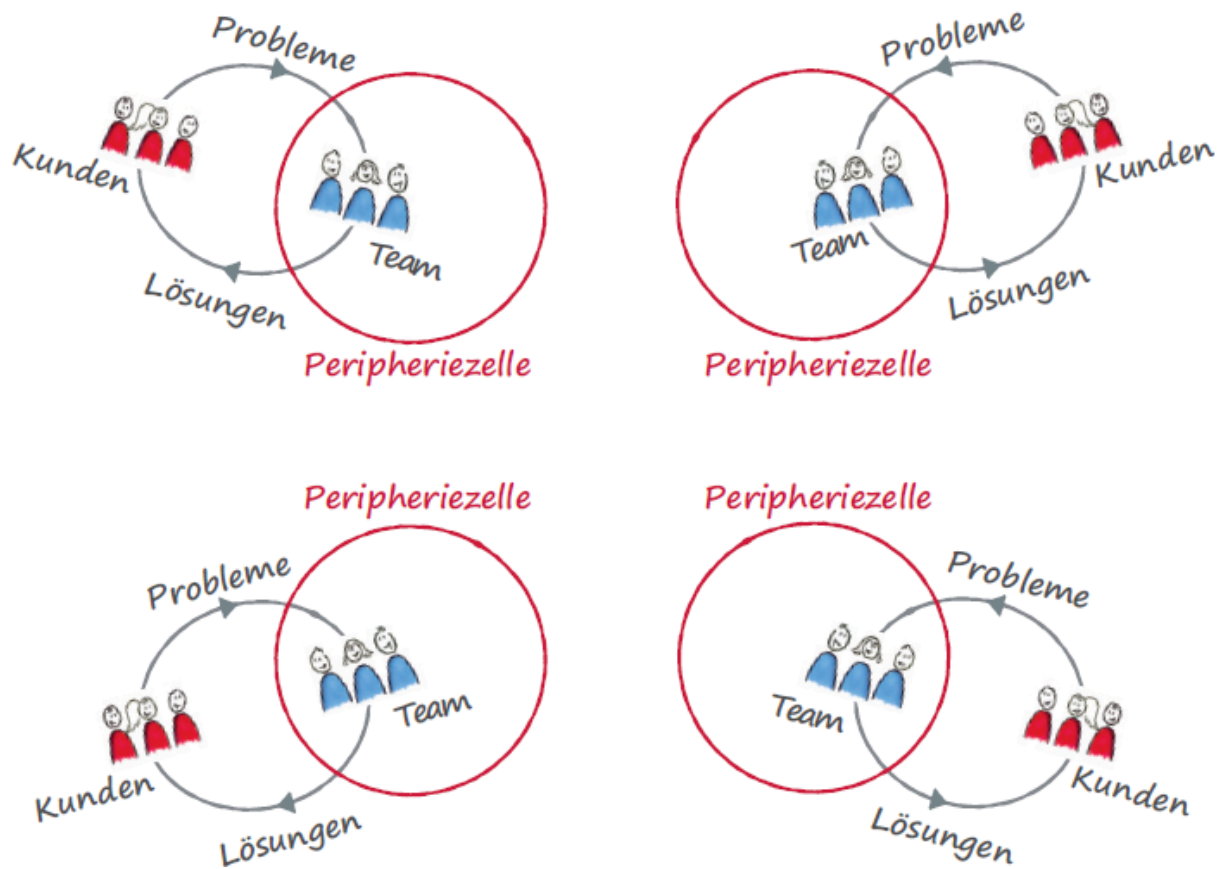


Abb. 4-3 Teambasierte Unternehmensstruktur

Offensichtlich reicht diese Struktur aber noch nicht aus. Ohne weitere Verbindungen gäbe es keinen Grund, warum die Peripheriezellen überhaupt ein gemeinsames Unternehmen sein sollten. Es wäre sinnvoller, wenn jede Zelle ein eigenes Unternehmen wäre. Allerdings würden die Zellen dann Synergieeffekte nicht nutzen können und das Lernen im Unternehmen wäre deutlich eingeschränkter.

Pfläging sieht daher zwei weitere Strukturelemente vor. Zunächst dürfen und sollen die Peripheriezellen sich gegenseitig unterstützen. Kann eine Peripheriezelle einen Bedarf ihrer Kunden nicht decken, bittet sie bei anderen Peripheriezellen um Hilfe. Die Peripheriezellen bilden also ein Netzwerk (siehe Abb. 4-4).

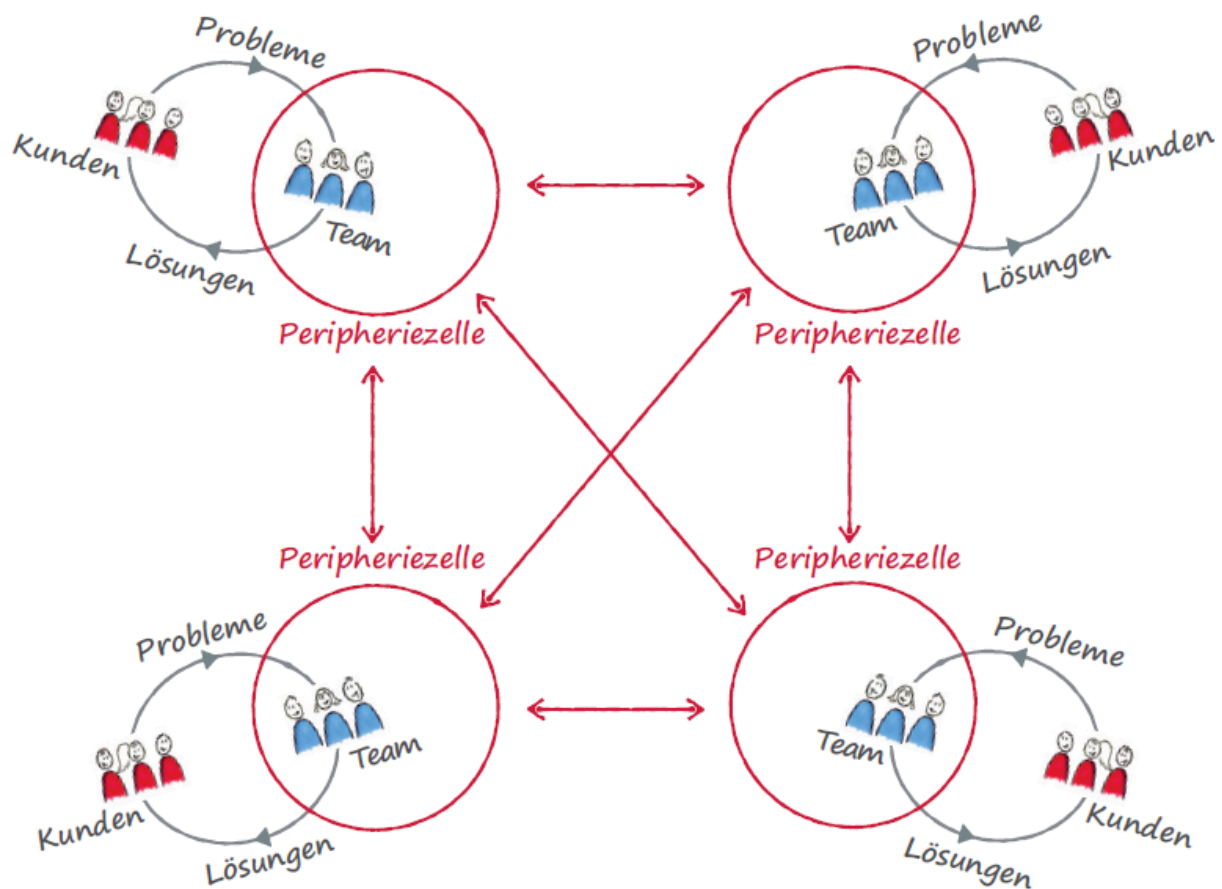


Abb. 4-4 Peripheriezellen bilden ein Netzwerk.

Zusätzlich können die Peripheriezellen sogenannte Zentrumszellen herausbilden (z. B. für Buchhaltung oder auch für innovative Entwicklungen). Diese erbringen zentral Dienste für die Peripheriezellen (siehe Abb. 4-5). Wichtig ist dabei, dass sich diese Zentrumszellen tatsächlich als Dienstleister verstehen und den Peripheriezellen keine Vorgaben machen. Denn dann würden diese Vorgaben als Störungen bei der Wertschöpfung der Peripheriezellen wirken.

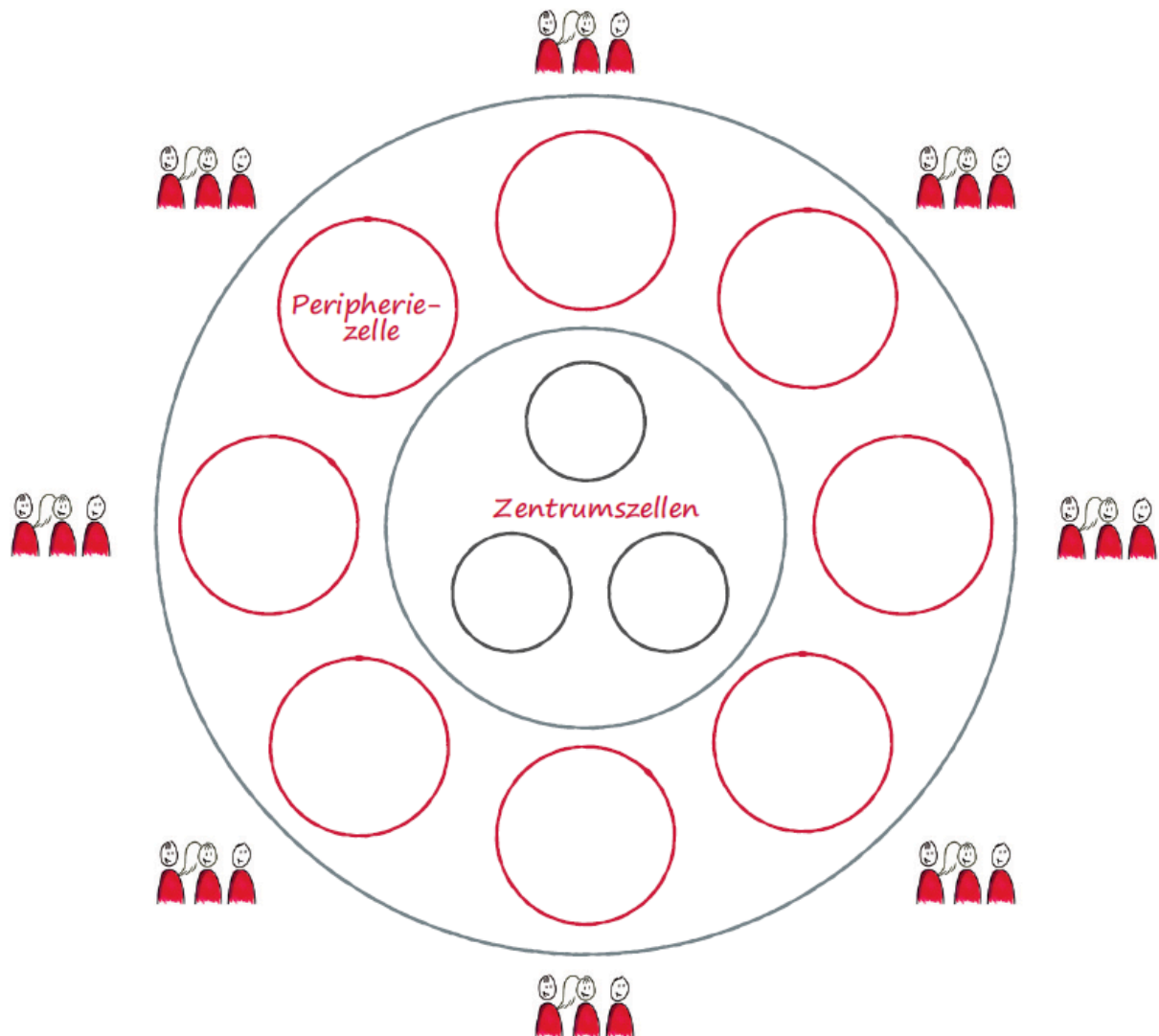


Abb. 4-5 Peripherie und Zentrum

Dieses ohnehin schon radikale Organisationsmodell verschärft Pfläging nochmals durch zwei zusätzliche Forderungen:

1. Die Anzahl der in den Zellen tätigen Menschen soll so klein sein, dass ein echtes Team entsteht.
2. Kein Mitarbeiter soll exklusiv in Zentrumszellen arbeiten. Entweder sind Mitarbeiter gleichzeitig in Peripherie- und Zentrumszellen beschäftigt oder sie rotieren. Dadurch wird der Wissensaustausch zwischen Peripherie und Zentrum sichergestellt und das Lernen im Unternehmen befördert².

4.2.1 Zellmodell in der Praxis der Softwareentwicklung

Pfläging's Zellmodell mutet so radikal an, dass man sich fragen muss, ob es in der Praxis wirklich funktioniert. Und falls sich diese Frage bejahen lässt, stellt sich natürlich auch noch die Frage, ob und wie das Modell für das eigene Unternehmen angewendet werden kann.

Zunächst einmal ist das Modell relativ leicht auf Unternehmen anwendbar, die mit einer ausgeprägten Filialstruktur arbeiten (z. B. Handel). Dort wäre jede Filiale eine Peripheriezelle. Je mehr Entscheidungskompetenzen den Filialen zugestanden wird, desto kundenorientierter können sie entscheiden und desto zufriedener werden die Kunden sein. Pfläging führt Svenska Handelsbanken und dm drogerie markt als Beispiele für filialbasierte Unternehmen an, die hohe Entscheidungskompetenz in den Filialen haben.

Da wir uns in diesem Buch nicht mit Handel beschäftigen, sondern mit Softwareentwicklung, stellt sich natürlich die Frage, wie dieser Bereich in das Modell passt. Prinzipiell kann Softwareentwicklung auf zwei Arten integriert werden. Man kann Softwareentwicklung als eine oder mehrere Zentrumszellen begreifen, die Dienste für die Peripheriezellen erbringen (siehe Abb. 4–6).

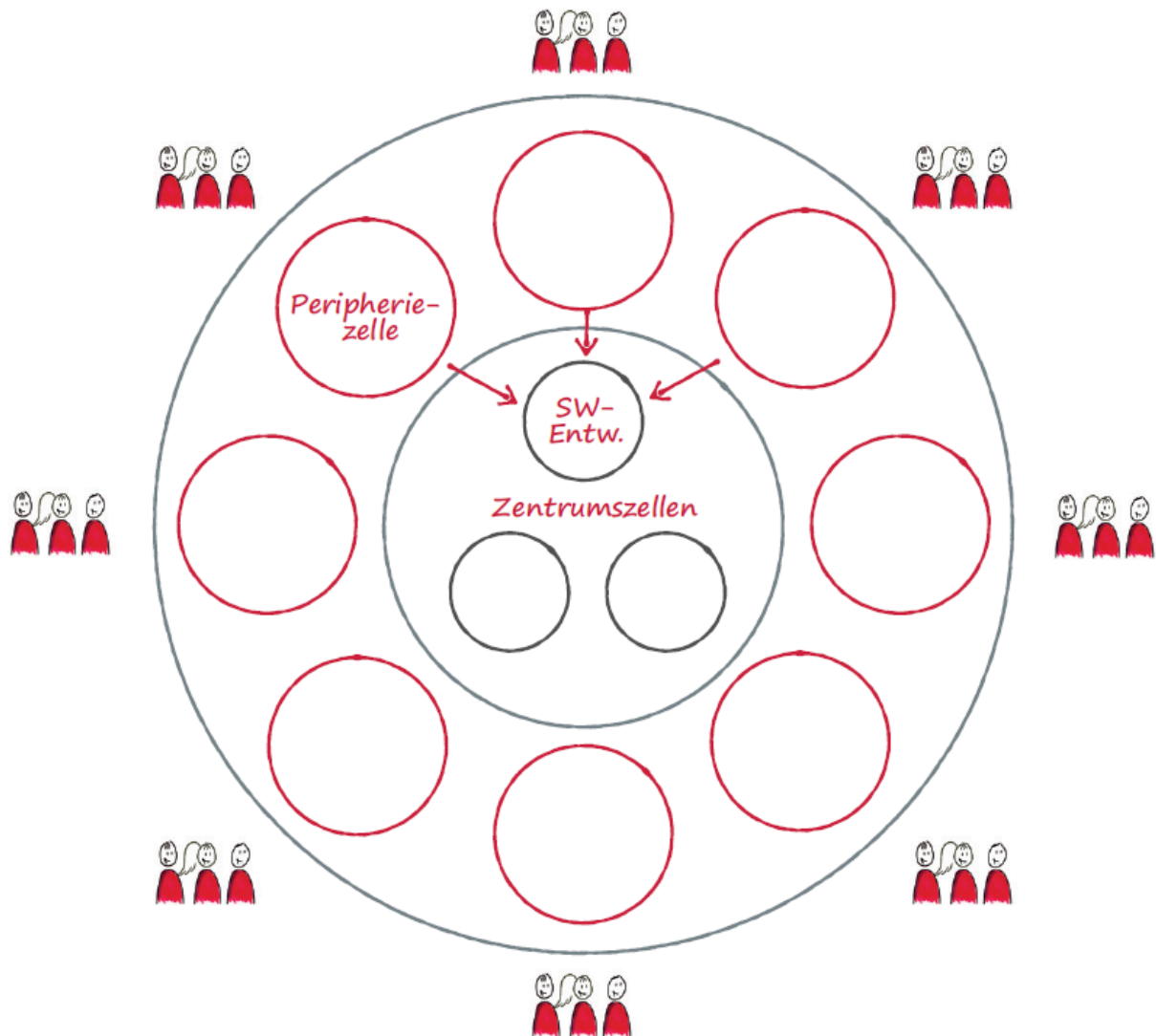


Abb. 4-6 Softwareentwicklung im Zentrum

Alternativ kann die Softwareentwicklung auch in der Peripherie stattfinden. Daher kann entweder jede Peripheriezelle ihre eigene Softwareentwicklung haben oder einige Peripheriezellen können eigene Softwareentwicklung betreiben und diese den anderen Zellen als Dienstleistung anbieten (siehe Abb. 4-7).

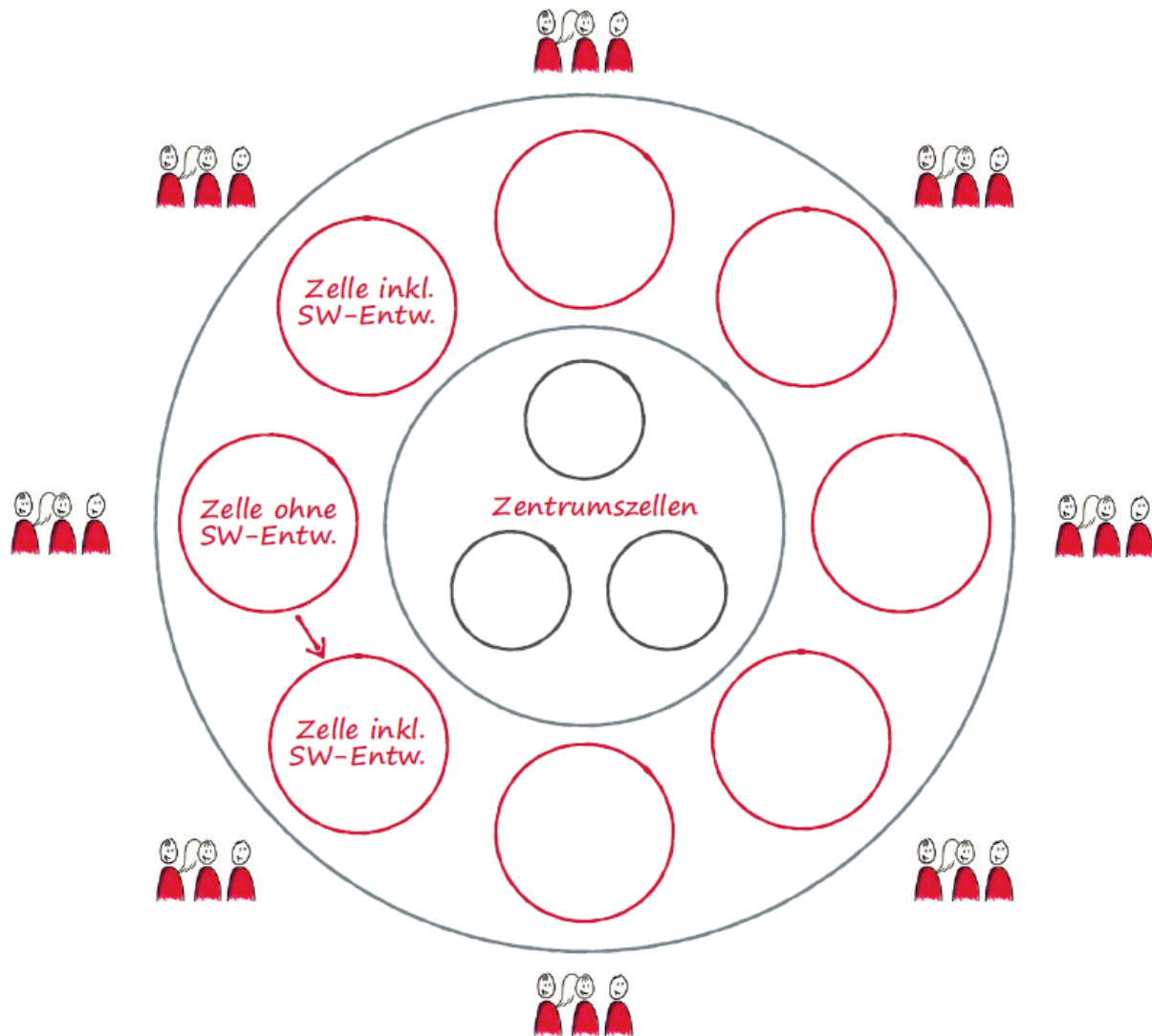


Abb. 4-7 Softwareentwicklung in der Peripherie

Nach dem 3-Horizonte-Modell ist in den meisten Fällen ein Mischmodell sinnvoll. Die Optimierung existierender Produkte und Dienstleistungen (Horizont 1) sollte direkt in den Peripheriezellen stattfinden. Der Umweg über Zentrumszellen ist unnötig schwerfällig.

Die Entwicklung neuer Produkte oder Dienstleistungen sollte hingegen in Innovationszellen im Zentrum erfolgen. Für die Entwicklung wird eine temporäre Zentrumszelle gebildet, die sich wieder auflöst, wenn das Produkt bzw. die Dienstleistung in die Peripherie übergeht. Nonaka und Takeuchi weisen in [Nonaka & Takeuchi 1995] bereits darauf hin, dass Mitarbeiterrotation ein sehr mächtiges Instrument zum Wissenstransfer insbesondere zwischen Geschäfts- und Innovationssystem darstellt. Das Team in der Innovationszelle sollte aus Mitarbeitern der Peripherie zusammengestellt werden. So wird sichergestellt, dass direkte Erfahrungen mit dem Markt und den Bedürfnissen der Endkunden im

Team vorhanden sind. Nach Abschluss der Entwicklung sollten die Teammitglieder wieder in die Peripherie integriert werden. So wird sichergestellt, dass das Wissen um das neue Produkt bzw. die neue Dienstleistung reibungsfrei in der Peripherie ankommt und dort für die Endkunden der maximale Nutzen generiert werden kann.

4.2.2 Mehr als ein Team pro Zelle

Natürlich ist es denkbar, dass in einer Zelle mehr als neun Personen mitarbeiten müssen und damit die empfohlene Obergrenze für ein agiles Team überschritten wird. In diesem Fall spricht man von »agiler Skalierung«: Mehrere Teams sind stark voneinander abhängig, z. B. weil sie am selben Produkt arbeiten.

Für agile Skalierung gibt es eine Reihe von Frameworks, die Lösungsmuster anbieten – die bekanntesten sind LeSS und SAFe. Während agile Skalierung für viele Unternehmen eine relevante Herausforderung darstellt, ist das Befolgen eines Skalierungsframeworks ein zweischneidiges Schwert. Natürlich sollte man bei der agilen Skalierung nicht mit Scheuklappen unterwegs sein und die Erfahrungen zu dem Thema ignorieren. Daher lohnt sich eine Auseinandersetzung mit den Skalierungsframeworks. Auf der anderen Seite darf man nicht einfach einem der Skalierungsframeworks folgen und dieses als Blaupause für seine eigene Struktur verwenden. Agiles Verständnis muss organisch wachsen und kann nicht in einem Big-Bang-Ansatz verordnet werden.

Ken Schwaber schreibt dazu: »Values and principles scale, but practices are context sensitive« [Schwaber 2013].

Werte und Prinzipien skalieren, Praktiken nicht. Praktiken sind kontextabhängig und was in einem Kontext funktioniert, kann in einem anderen sogar hochgradig schädlich sein.

Wir empfehlen daher, schrittweise die eigene Skalierungsstruktur zu suchen und mit konkreten Techniken aus Skalierungsframeworks zu experimentieren, um real existierende Probleme zu lösen. Bei schrittweiser Entwicklung der eigenen Skalierungsstruktur lassen wir uns von den agilen Werten und Prinzipien leiten. Wir suchen also nach Strukturen, in denen Menschen ihr Potenzial ausschöpfen können, und nicht nach Strukturen, die Menschen beschränken. Wir suchen nach wenigen leichtgewichtigen Strukturen, die sich leicht anpassen lassen, und nicht nach schwergewichtigen Strukturen für die Ewigkeit. Wir suchen nach Strukturen, die die Wertschöpfung für Endkunden in den Vordergrund stellen, und nicht nach Strukturen, die primär das Einhalten von

Regeln sicherstellen. Konkrete Ansätze zur schrittweisen Entwicklung der eigenen Strukturen diskutieren wir im nächsten Kapitel.

Das LeSS-Framework kommt von seinen Grundannahmen kundenwertoptimierenden Teams am nächsten. Daher skizzieren wir es in Anhang B.

Fallbeispiel: Eigene Skalierungsstruktur finden (von Jürgen Hoffmann)

Für einen »Proof of Concept« wurde ein Scrum-Team gebildet, das sich ein paar grundlegenden technischen und architektonischen Herausforderungen eines neuen Produktes stellte. Nach einigen Wochen sollte die Liefergeschwindigkeit erhöht werden. Es wurden weitere Experten hinzugezogen und auf einmal war das Produktentwicklungsteam mehr als 20 Personen groß. Diese Gruppe teilte sich mithilfe eines Moderators auf eigenen Wunsch in zunächst 3 Feature-Teams auf. Diese Teilung folgte der Notwendigkeit, Retrospektiven und Backlog-Verfeinerungstermine mit kleineren Gruppen von Mitarbeitern effektiver zu machen.

Viele solcher Schritte führen nach einiger Zeit zu einer Organisation, die in Struktur- und Prozessbildern festgehalten werden kann. Die Stärke des Vorgehens liegt darin, dass jedes Struktur- und Prozesselement aus einer inneren Notwendigkeit der aktuellen Produktentwicklung entsteht. Das ist nachvollziehbar für die Mitarbeiter und wird von ihnen selbst gestaltet.

4.2.3 Alles Illusion?

In vielen Unternehmen mag das vorgestellte Zellmodell zwar als plausibel, aber völlig illusorisch erscheinen. Man kann sich vielleicht nicht vorstellen, dass sich das eigene Unternehmen jemals so umgestalten wird. Es mag sein, dass im existierenden Unternehmen diese Struktur nicht vollständig umgesetzt werden kann oder dass ein vollständiges Umsetzen gar nicht sinnvoll ist. Es ist aber sehr sinnvoll, die existierenden Strukturen vor dem Hintergrund des Zellmodells infrage zu stellen und sich diesem schrittweise anzunähern. Wir sehen uns dazu zwei Beispiele an.

Fallbeispiel: Struktur der Bank-IT spiegelt Fachstruktur (von Stefan Roock)

Eine Bank wollte agiles Arbeiten in ihre IT einführen und entschied sich letztlich zu einer kompletten Reorganisation der IT. Im Ergebnis wurden Organisationseinheiten aufseiten der IT geschaffen, die die existierende

Struktur der Geschäftsfelder auf Fachseite abbildeten (siehe Abb. 4–8). Die IT-Zellen bestehen dabei jeweils aus mehreren Teams.

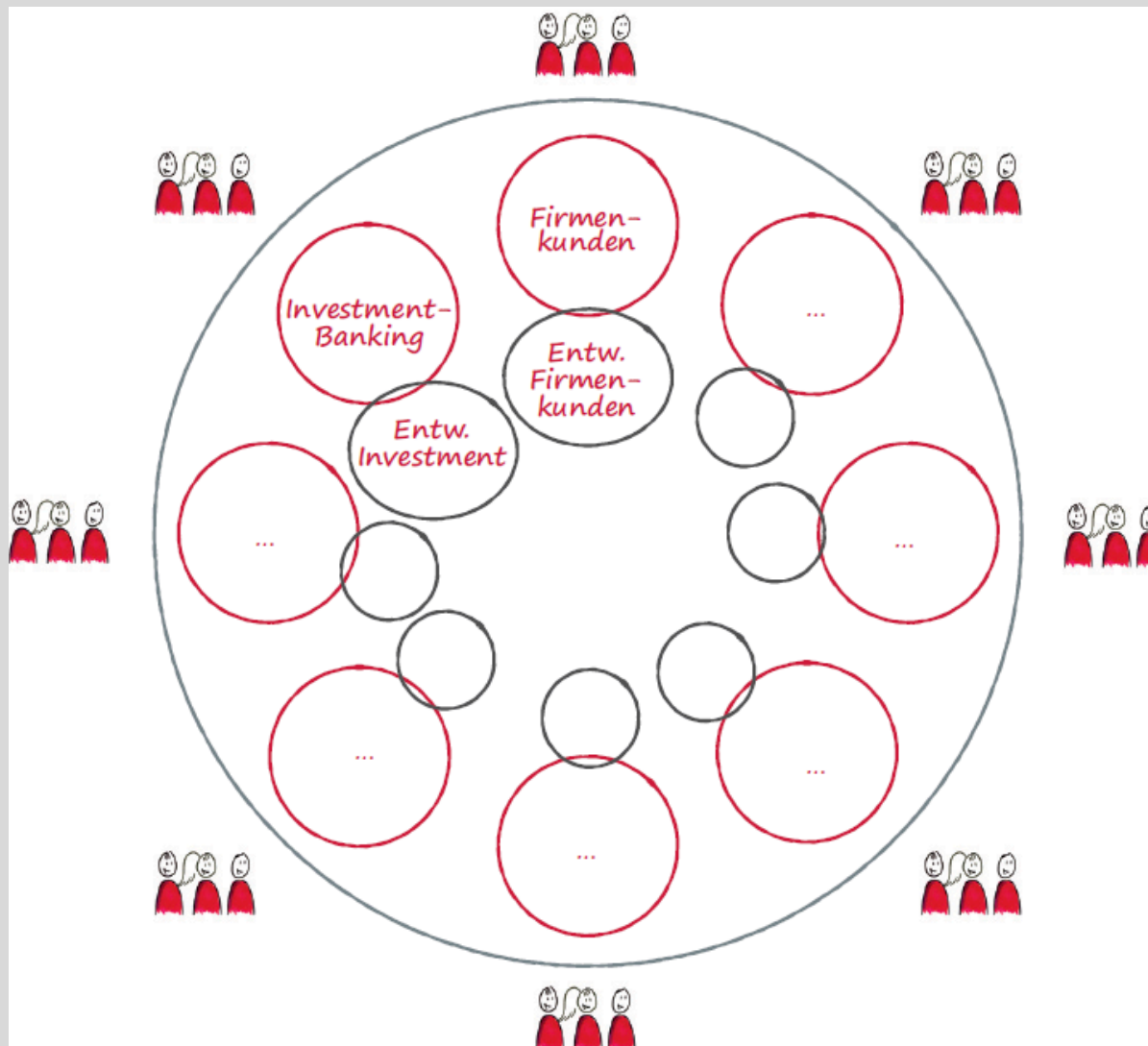


Abb. 4–8 Entwicklung organisiert nach Geschäftsfeldern

In diesem Fall wurde die *Wertschöpfungsstruktur* nach dem Zellmodell organisiert, aber noch nicht die Aufbauorganisation. Die Entwickler sind nach wie vor Führungskräften in der IT unterstellt. Die so entstehende Abweichung zwischen Wertschöpfungsstruktur und Aufbauorganisation kann Störungen erzeugen (wenn z. B. eine IT-Führungskraft die Weiterentwicklung der Mitarbeiter unpassend zu den Zielen des Fachbereiches steuert). Trotzdem ist mit der Reorganisation der Wertschöpfungsstruktur ein erster wichtiger Schritt in Richtung marktorientiertem Zellmodell gegangen erfolgt.

Fallbeispiel: Wooga (von Stefan Roock)

Ein zweites Beispiel ist die Berliner Firma Wooga, die Onlinespiele entwickelt. Eine neue Spieleentwicklung beginnt stets mit einem sehr kleinen Team aus zwei oder drei Personen. Je weiter die Entwicklung fortschreitet und je sicherer man sich wird, dass das Spiel gute Erfolgsaussichten hat, desto größer wird das Team. Nach dem Zellmodell handelt es sich bei dem Team um eine Zentrumszelle. Diese nimmt die Dienstleistung einer anderen Zentrumszelle in Anspruch, die Entwickler in der passenden Qualifikation liefert. Wird das Spiel schließlich veröffentlicht, wird aus der Zentrumszelle eine Peripheriezelle, die das Spiel betreibt und weiterentwickelt.

Wooga wird häufig als Vorzeigebeispiel für agile Organisationsstrukturen verwendet. Natürlich muss man dieses Beispiel (wie jedes andere auch) im entsprechenden Kontext betrachten. Die Strukturen von Wooga sind auf das spezielle Wooga-Geschäft abgestellt, in dem überschaubare Spiele mit kleinen Teams entwickelt und betrieben werden können. Dadurch hat Wooga es deutlich einfacher, vollständig autonome Teams zu bilden, als Unternehmen mit sehr komplexen Wertschöpfungsketten. Jedes Unternehmen muss die Strukturen finden, die zu dem eigenen Geschäft passen. Beispiele wie Wooga können »lediglich« zur Inspiration dienen, gedanklich neue Wege zu beschreiten.

4.3 Alignment bei dezentralen Strukturen

»Aber dann macht doch jeder, was er will.« Das hört man häufig, wenn es um dezentrale Entscheidungen geht. Tatsächlich ist diese Gefahr nicht von der Hand zu weisen und viele Unternehmen kennen das (zumindest gefühlte) Chaos, das durch autonome Teams entstehen kann. Mitunter entsteht der Eindruck, Alignment (alle ziehen an einem Strang) und Autonomie stünden im Widerspruch zueinander: Mehr Autonomie bedeutet weniger Alignment und ein größeres Alignment lässt sich nur durch Aufgabe von Autonomie erreichen. Bungay argumentiert hingegen, dass dem nicht so ist (siehe [Bungay 2010] und Abb. 4–9). Stattdessen kann man Alignment und Autonomie als unabhängige Dimensionen betrachten (siehe Abb. 4–10). Alignment bei geringer Autonomie stellt man durch Anweisung und Kontrolle her. Beim Militär spricht man von *Command Control*. Das von vielen Unternehmen erlebte Chaos beim Übergang zu dezentralen Strukturen ist der Übergang von links oben nach rechts unten im Diagramm. Autonomie nimmt zu, Alignment nimmt ab. Die Ursache liegt darin, dass zwar *Command Control* als Steuerungsinstrument abgeschafft, aber kein geeigneteres

Steuerungsinstrument installiert wurde. Ein solches Steuerungsinstrument muss für alle Beteiligten das Was und Warum klarstellen. Dann werden die Mitarbeiter und Teams ihre dezentralen Entscheidungen an diesem Was und Warum ausrichten. Beim Militär ist von *Mission Command* die Rede: Die Einheit hat eine Mission zu erfüllen und es ist ihr vollkommen freigestellt, wie sie das macht.

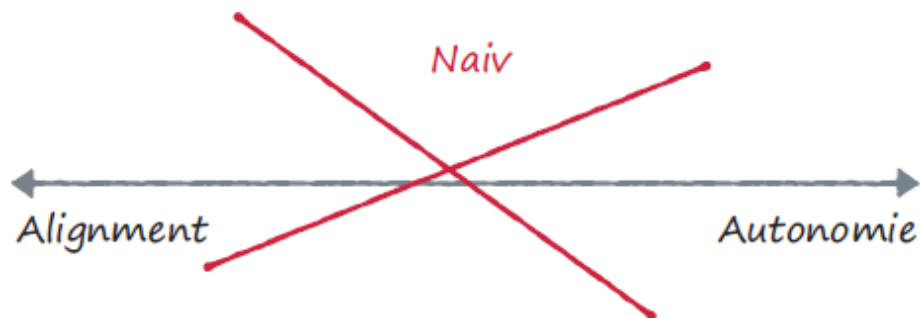


Abb. 4-9 Alignment und Autonomie stehen nicht im Widerspruch zueinander.

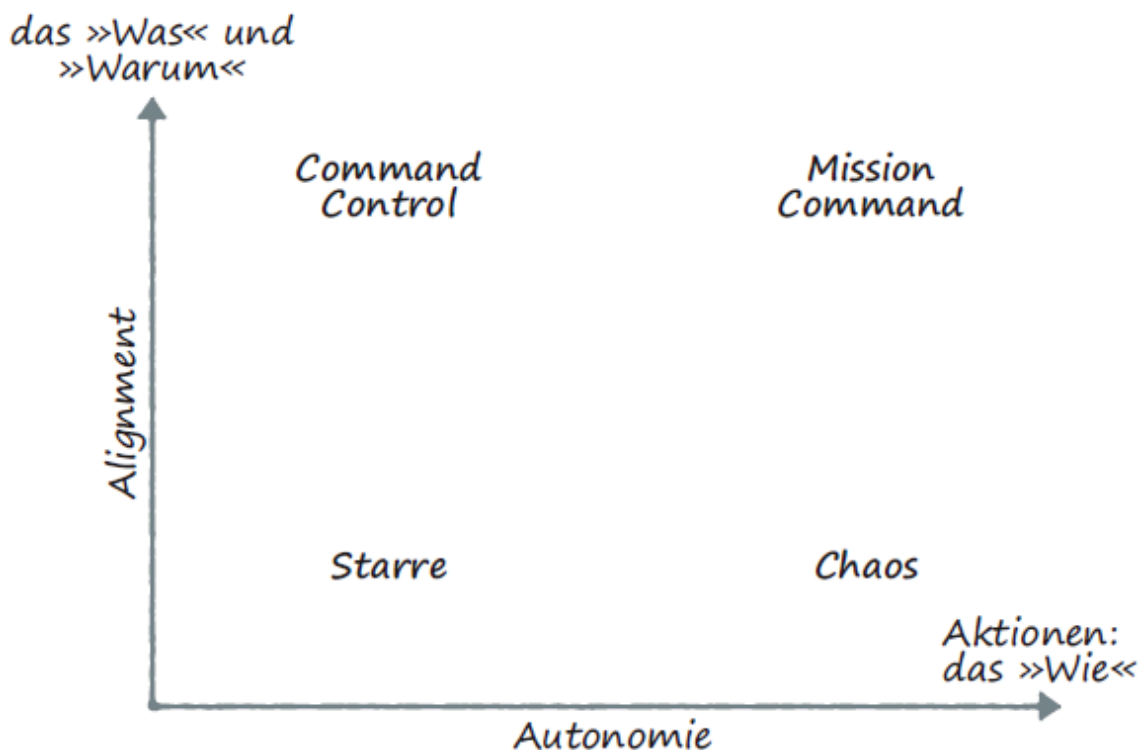


Abb. 4-10 Autonomie und Alignment als unabhängige Dimensionen

4.3.1 Management by Objectives (MbO)

Interessanterweise hat Drucker bereits 1954 diesen Sachverhalt beschrieben, allerdings ohne die Visualisierungen (siehe [Drucker 1954]). Drucker schlug vor, das Was und Warum über Ziele (engl. *Objectives*) klarzustellen, um dezentrale Entscheidungen (Autonomie) zu ermöglichen.

Vor den Zielen kommt bei Drucker allerdings eine geeignete Unternehmensstruktur. Unternehmen sollten zunächst nach Märkten oder Produkten organisiert sein. Wie eine solche Struktur aussehen kann, haben wir bereits oben gesehen: Das von Niels Pfläging beschriebene Zellmodell folgt diesem Organisationsprinzip. Zu den Zeiten Druckers mit großen Produktionsunternehmen wie z. B. Ford waren die resultierenden Organisationseinheiten häufig noch zu groß, um ohne weitere Struktur zu existieren; man stelle sich Hunderte von Produktionsmitarbeitern für den Bau von Pkws vor. Wie oben beschrieben, schlägt Drucker als Substruktur *Phasen* vor.

Mögliche Phasen beim Hausbau könnten Fundament, Rohbau, Innenausbau etc. sein. Das Ausheben der Baugruppe ist möglicherweise keine sinnvolle Phase, weil die Baugruppe bei langer »Liegezeit« abrutschen kann. Stützt man die Baugrube allerdings ausreichend ab, entsteht so doch wieder eine valide Phase. An dem Baugruben-Beispiel kann man bereits erkennen, dass Phasen auf den ersten Blick manchmal wie funktionale Silos aussehen, diesen aber selten entsprechen. Für eine ausreichend abgestützte Baugrube brauchen wir nicht nur Baggerfahrer, sondern auch Zimmermänner. Und nicht zuletzt forderte Drucker, dass die Phasenergebnisse qualitätsgesichert werden. Die entsprechenden Fähigkeiten müssen in der Phase ebenfalls vorhanden sein. Phasen-Organisationseinheiten müssen also auch interdisziplinär besetzt sein.

In der Softwareentwicklung kennen wir sequenzielle Phasenmodelle seit Jahrzehnten (Wasserfall, V-Modell etc.). Diese Modelle erfüllen allerdings Druckers Anspruch an Phasen nicht. Wir haben gelernt, dass die Phasenergebnisse in diesen Modellen schnell an Wert verlieren (heute aufgenommene Anforderungen sind meist schon nach wenigen Monaten überholt) und schlecht qualitätsgesichert werden können (ob die formulierten Anforderungen wirklich das Problem der Anwender lösen, wissen wir meistens erst, wenn die Software entwickelt wurde). Produktinkremente à la Scrum hingegen erfüllen die Kriterien, die Drucker an Phasen anlegt.

Basierend auf diesen Organisationsprinzipien und der Idee der »Führung über Ziele« definierte Drucker ein komplettes Managementsystem namens *Management by Objectives (MbO)*: Auf jeder Unternehmensebene werden Ziele definiert, an denen sich die jeweilige Einheit ausrichtet. Eine Top-down-Zerlegung der Ziele ist naheliegend: Unternehmensziele werden in Bereichsziele gegliedert, diese dann wieder in Abteilungsziele etc.

Eine strikte Top-down-Zerlegung ist bei MbO allerdings gar nicht vorgesehen. Stattdessen sollen die obersten Unternehmensziele auf *jeder* Ebene klar sein. Die

einzelnen Ebenen sollen selbst dazu passende Ziele für sich definieren – dazu führen sie einen kontinuierlichen Dialog mit den Einheiten über, unter und neben sich. Mit diesem Mechanismus werden Ziele von der Unternehmensspitze bis hinunter zu den Mitarbeitern verwendet (siehe Abb. 4–11).

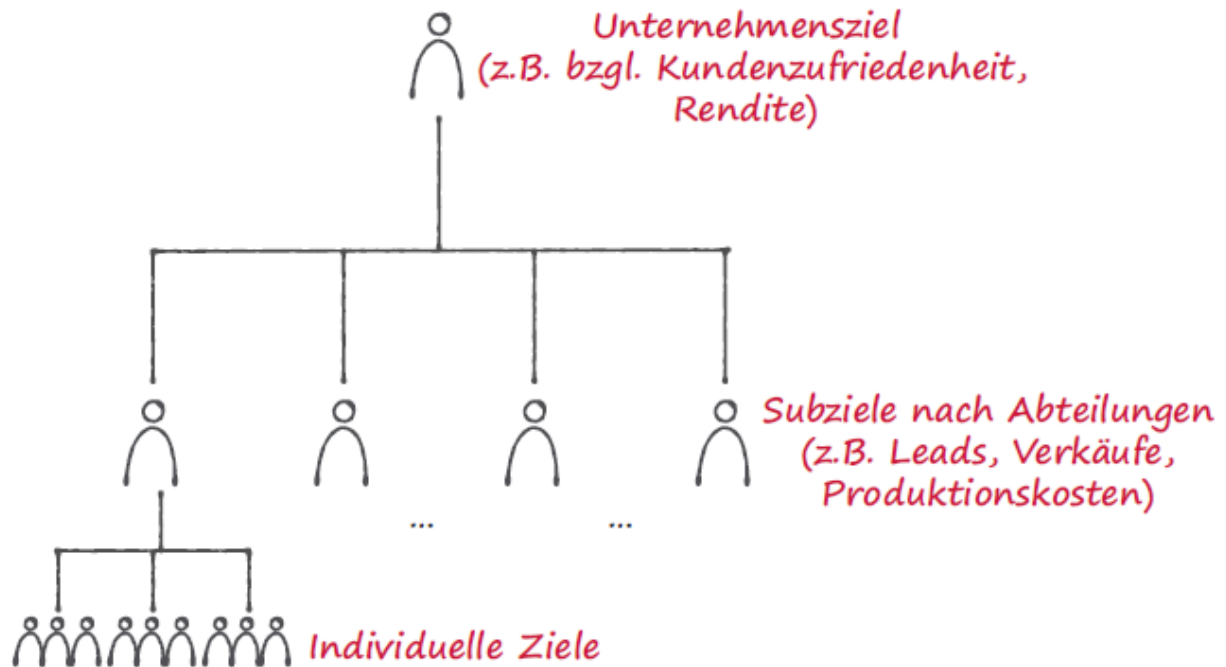


Abb. 4-11 Ziele (Objectives) auf allen Unternehmensebenen

Besonders herausfordernd ist, die Ziele der unteren Ebenen so zu formulieren, dass sie am Ende nicht zu Mikromanagement führen, sondern tatsächlich dezentrale Entscheidungen ermöglichen.

Messbarkeit von Zielen und Anreizsysteme

Laut Peter Drucker sollten Ziele messbar sein – allerdings nur zur Selbstkontrolle. Jede Einheit soll selbst kontrollieren können, ob sie auf dem richtigen Weg ist. Drucker betont, dass die Messung nicht zur Kontrolle durch übergeordnete Einheiten verwendet werden soll. Dazu passt, dass Drucker davon abrät, Boni an die Zielerreichung zu knüpfen. Dadurch gäbe man einen starken Anreiz, auf lokale Ziele zu optimieren und dabei das globale Ganze zu beschädigen.

4.3.2 Objectives and Key Results (OKR)

Insbesondere in Start-ups wird in den letzten Jahren häufig mit *Objectives and Key Results* (abgekürzt OKR) gearbeitet. Das OKR-System wird häufig Google zugeschrieben, stammt allerdings aus den 1970er-Jahren von Intel³. Bei OKR werden die Ziele (Objectives) mit Key Results unterlegt, die anzeigen, inwieweit

das Ziel erreicht wurde. Die Key Results sollen dabei anspruchsvoll sein – es ist keine 100 %ige Erreichung vorgesehen, sondern es genügen 60–70%. Typischerweise werden Quartalszeiträume mit OKR geplant. OKR sind eine mögliche MbO-Implementierung, die viele Gemeinsamkeiten mit »klassischem« MbO aufweist (siehe Tab. 4–1). Die Unterschiede betreffen die Planungszeiträume, den angestrebten Zielerreichungsgrad und die Kopplung von Zielerreichung und Gehalt:

▪ **Planungshorizont**

Mit MbO werden kurze und lange Zeiträume geplant. Es sollen sowohl monatliche wie auch mehrjährige Objectives definiert werden. Demgegenüber adressiert OKR nur Planungshorizonte von einem bis drei Monate.

▪ **Zielerreichungsgrad**

Während MbO eine vollständige Zielerreichung oder sogar ein Übertreffen des Ziels anstrebt, definiert man bei OKR die Ziele so, dass eine Zielerreichung von 60 – 70 % angestrebt wird.

▪ **Zielerreichung und Gehalt**

Bei der Gehaltsfrage sieht MbO vor, dass die Fähigkeit, Objectives zu definieren und zu erreichen, einen Einfluss auf Beförderungen bzw. Gehaltserhöhungen hat. Das OKR-Konzept sieht keine Kopplung der Zieldefinition und der Zielerreichung an das Gehalt vor. Beide Ansätze warnen davor, Boni an die Zielerreichung zu koppeln.

	MbO	OKR
Art der Ziele	Ergebnis (einmalig)	
Zielfindung	Schrittweise Zerlegung entlang der Hierarchie. Gemeinsame Erarbeitung gewünscht (top-down und bottom-up kombiniert), in der Praxis bei OKR häufiger anzutreffen als bei MbO.	
Unternehmensziel vs. Einzelziele	Summe der Einzelziele ergibt das Unternehmensziel	
Planungshorizont	Monate und Jahre	Monat bis Quartal
Angestrebte Zielerreichung	100 % oder mehr	60–70 %

Messung	Nur zur Selbstkontrolle während der Umsetzung	Während der Umsetzung zur Selbstkontrolle und am Ende der Planungsperiode, um festzustellen, inwieweit das Ziel erreicht wurde.
Gehaltserhöhung	Gehaltserhöhung für Manager setzt voraus, dass sie sinnvolle Ziele definieren und erreichen können.	Keine Abhängigkeit zur Zielerreichung
Bonus	Unabhängig von Zielerreichung	

Tab. 4-1 Gemeinsamkeiten und Unterschiede zwischen MbO und OKR

Neben diesen prinzipiellen Unterschieden der Theorie unterscheiden sich MbO und OKR in der Praxis durchaus. Die Unternehmen, die OKR verwenden, tun dies nach unserer Erfahrung in der Praxis transparenter und kooperativer. Häufig wird das Was »von oben« vorgegeben und das Wie »von unten« gefunden. Dass dies von Drucker bei MbO auch so intendiert war, ändert leider wenig darin, dass die MbO-Praxis dann doch häufig anders aussieht. (Und natürlich haben wir auch gesehen, wie OKR eingeführt wurde, um straff top-down durchzuregieren.)

In den für dieses Buch wesentlichen Punkten stimmen MbO und OKR allerdings überein, sodass wir in der Folge nur noch von MbO sprechen und dabei jeweils auch OKR meinen.

4.3.3 MbO-Beispiel – so bitte nicht

Das folgende (sehr einfache) Beispiel illustriert, wie MbO mitunter eingesetzt wird (siehe Abb. 4-12). Wir werden unten diskutieren, warum das Beispiel problematisch ist und auch nicht zu Druckers Intention passt.

Wir gehen von einem Dienstleistungsunternehmen im Projektgeschäft aus, das eine Umsatzrendite von 20 % anstrebt. Auf Basis von Erfahrungen aus der Vergangenheit definiert das Management eine Reihe von Annahmen:

- Das Unternehmen hat jährliche Kosten in Höhe von 5 Mio. Euro/Jahr. Der Zielumsatz für 20 % Umsatzrendite liegt also bei etwa 6 Mio. Euro.
- Das durchschnittliche Projekt hat einen Umsatz von 250.000 Euro.
- Die Konversionsrate von Leads in Aufträge liegt bei 10 %.
- Pro Monat sollen zwei neue Projekte akquiriert werden – eines mit einem neuen und eins mit einem existierenden Kunden.

- Die Aufwandsschätzungen der Projekte können im Großen und Ganzen eingehalten werden (Unter- und Überschreitungen der geschätzten Aufwände halten sich die Waage).

Auf dieser Basis können Ziele für die Abteilungen Marketing, Vertrieb und Entwicklung definiert werden. Das Marketing müsste 20 Leads monatlich generieren, auf deren Basis der Vertrieb Abschlüsse in Höhe von 500.000 Euro (durch neue und existierende Kunden) erzielen müsste. Die Entwicklung darf nicht zu weit von der initialen Schätzung abweichen. Wir definieren als Ziel, dass 90 % der Projekte *in time and budget* abgewickelt werden.

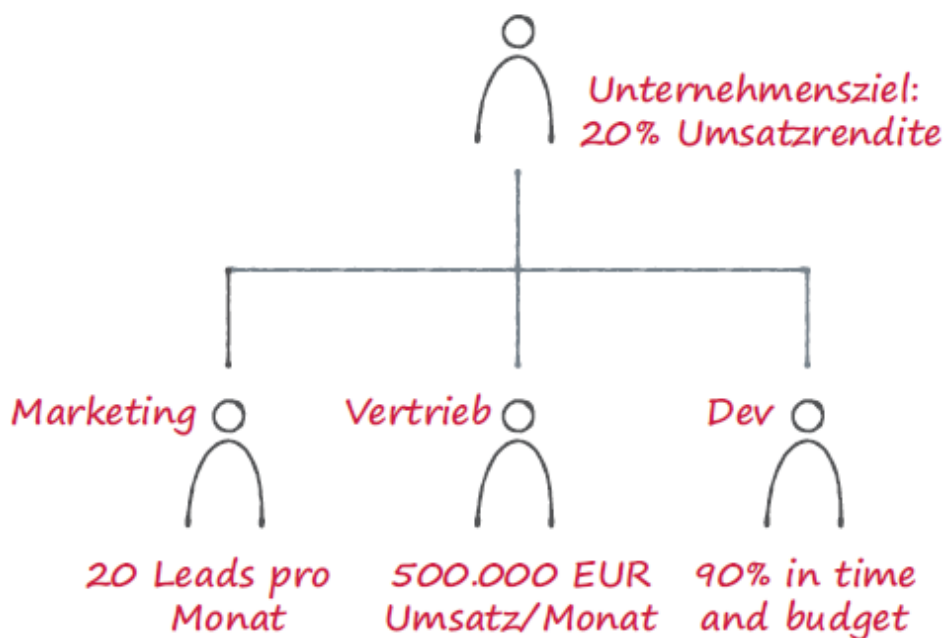


Abb. 4-12 Beispiel für eine falsche MbO-Anwendung

Dieses Beispiel könnte in der Praxis so oder so ähnlich aussehen. Allerdings enthält es gleich zwei Aspekte, von denen Drucker dringend abrät: Fokussierung auf Geld/Umsatz statt auf Kundenzufriedenheit sowie eine Abteilungsstruktur, die nach funktionalen Silos geschnitten ist. Wie oben beschrieben, fordert Drucker eine Unternehmensstruktur, die entlang von Märkten oder Produkten organisiert ist und nicht nach funktionalen Silos. Wir schauen uns im folgenden Abschnitt an einem Beispiel an, wie eine bessere MbO-Anwendung aussieht.

4.3.4 MbO-Beispiel – besser

Wir gehen wieder von unserem Dienstleistungsunternehmen im Softwareprojektgeschäft aus. Jetzt ist es aber nicht nach funktionalen Silos (Marketing, Vertrieb, Entwicklung) organisiert, sondern nach Märkten (Konzerne,

Mittelstand, Frankreich). Das Unternehmen möchte die Kundenzufriedenheit steigern, den Mittelstandsmarkt vergrößern und in Frankreich einen neuen regionalen Markt aufbauen. Durch Aushandlung zwischen Unternehmensführung und den einzelnen Abteilungen entstehen konkretisierte Ziele: Sowohl bei Konzernen wie auch bei Mittelständlern soll die Kundenzufriedenheit um 5 Prozentpunkte steigen. Bei den Mittelständlern soll es außerdem 10 % Wachstum geben. Für Frankreich wird genauer definiert, was unter erfolgreichem Markteintritt zu verstehen ist: keine Verluste und Aufbau von zwei neuen Teams. Abbildung 4-13 zeigt das Gesamtbild.

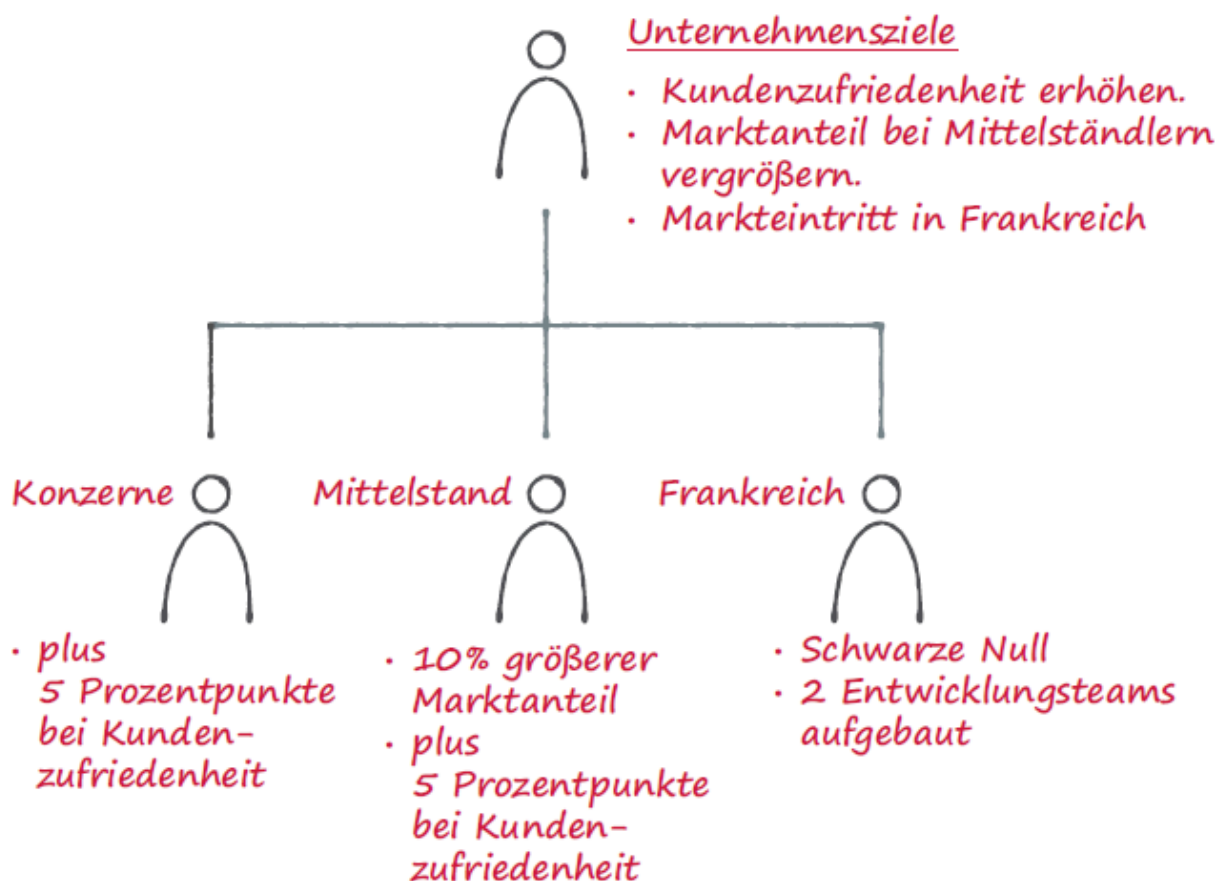


Abb. 4-13 Besseres MbO-Beispiel

Unterhalb der drei gezeigten Abteilungen findet sich natürlich eine Substruktur (z. B. könnten Konzerne differenziert werden nach Branchen wie Telekommunikation, Banken & Versicherungen, Energieversorger). Die Folge ist, dass viele Funktionen aus Gesamtunternehmenssicht redundant vorhanden sind. Jede der drei Abteilungen wird z. B. mindestens einen Vertrieb haben.

4.3.5 Nutzen und Gefahren von Management by Objectives

Der Hauptnutzen von MbO besteht darin, dass klare Ziele bei dezentralen Entscheidungen Alignment herstellen. Allerdings geht mit einem MbO-System auch eine Reihe von Gefahren einher, die zu ernsthaften Störungen agilen Arbeitens führen können.

- **Unpassende Organisationsstruktur**

Nach funktionalen Silos organisierte Unternehmen erzeugen starke Abhängigkeiten zwischen Zielen, sodass die Ziele kontinuierlich mit viel Aufwand angepasst werden müssten. Viele Unternehmen müssten sich erst einmal reorganisieren, um MbO überhaupt sinnvoll einsetzen zu können. Und die Unternehmen müssen sich darauf einstellen, dass sie sich ständig neu organisieren müssen, weil die Organisationsstruktur sich an den Unternehmenszielen ausrichten muss.

- **Varianzen**

Business-relevante Ziele sind anfällig für Varianzen, die die Zielerreichung mehr oder weniger zufällig machen.

- **Lokale Optimierungen**

Die Summe von Einzelzielen führt zu lokaler Optimierung und damit zu globaler Suboptimierung.

- **Hands-off-Management**

Führung findet über Ziele statt und die Interaktion zwischen Führungskräften und Mitarbeitern leidet.

- **Außerordentliche Anstrengungen**

Herausfordernde Ziele werden häufig durch außerordentliche Anstrengungen erreicht, die nicht zu nachhaltigem Erfolg führen.

- **Zielkonflikte**

Individuelle Ziele geraten in Konflikt mit Teamzielen.

- **Innovationshemmnis**

Sehr klare Ziele sind Innovationen abträglich. Innovation braucht Doppeldeutigkeit.

Unpassende Organisationsstruktur

Wie oben beschrieben forderte Drucker, dass Unternehmen sich nach Märkten, Produkten oder Phasen organisieren sollen. Nur dann wären dezentrale Entscheidungen überhaupt möglich. MbO setzt also eine derartige Unternehmensstruktur voraus.

Viele Unternehmen sind allerdings nach funktionalen Silos geschnitten. Wendet man in so einem Unternehmen MbO an, so sind enge Abhängigkeiten zwischen den Zielen die Folge: Im oben angeführten schlechten MbO-Beispiel hängen die Ziele des Vertriebs sehr stark mit den Zielen des Marketings zusammen. Dadurch unterliegen die Ziele einer sehr großen Dynamik und müssten im Grunde kontinuierlich mit allen Beteiligten angepasst werden. Der damit einhergehende Aufwand ist aber so groß, dass er selten investiert wird. In der Folge laufen die einzelnen Organisationseinheiten unabgestimmt längst obsolet gewordenen Zielen hinterher.

Varianzen

Definiert man z. B. für den Vertrieb ein bestimmtes Umsatzziel, so hat er dieses nur zum Teil unter seiner Kontrolle. Wenn sich die gesamte Marktlage verschlechtert oder einfach das Marketing ineffektiv arbeitet, erreicht der Vertrieb sein Ziel vielleicht nicht. Der Vertrieb bekommt die »Schuld« zugewiesen, obwohl er vielleicht sehr gute Arbeit geleistet hat. Andersherum könnte ein plötzlicher Marktaufschwung oder ein medienwirksamer Skandal um den Hauptkonkurrenten die Umsätze plötzlich ankurbeln. Dann steht der Vertrieb als »Held« dar, ohne dass er dafür etwas getan hätte.

Die von Drucker geforderte Unternehmensstrukturierung nach Märkten oder Produkten statt nach funktionalen Silos mildert das Varianzproblem zumindest auf den oberen Hierarchieebenen. Wie gut das Wachstum bei den Mittelständlern in unserem besseren MbO-Beispiel funktioniert, hat die Abteilung auch nicht alleine unter Kontrolle. Auch hier spielt die gesamtwirtschaftliche Lage eine Rolle genauso wie das Verhalten der Wettbewerber.

Wenn Aufstiegschancen, Boni oder Ansehen mit der Zielerreichung einhergehen, belohnt und bestraft das Unternehmen seine Mitarbeiter damit zufällig.

Nun könnte man auf die Idee kommen, die Ziele so zu definieren, dass sie varianzfrei sind. Dummerweise mindert sich damit ihre Geschäftsrelevanz. Allzu häufig sind sogenannte *Vanity Metrics* das Ergebnis (siehe [Ries 2011]): Sie schmeicheln uns, ohne relevant für das Unternehmen zu sein. So lässt sich z. B. die Anzahl von Besuchern auf der Homepage durch geschickte Werbung

erkaufen. Wenn die gekauften Besucher der Homepage aber nur feststellen, dass sie sich für das dargebotene Angebot nicht interessieren, hat das Unternehmen nichts davon. Die in der Metrik ersichtliche steigende Anzahl von Homepage-Besuchern sah gut aus, bedeutete aber nichts: *Vanity Metric*.

Diesen Zusammenhang von Geschäftsrelevanz und Varianzen visualisiert Abbildung 4-14: Größere Geschäftsrelevanz geht mit mehr Varianzen einher. Man kann sich diesen Zusammenhang auch so herleiten: Unternehmen haben mit dem Markt zu tun, der u. a. aus Kunden und Wettbewerbern besteht. Die meisten geschäftsrelevanten Ziele haben daher auch mit dem Markt zu tun (Kosten sind eine Ausnahme). Am Markt nehmen neben dem eigenen Unternehmen offensichtlich noch weitere Akteure teil – vor allem Kunden und Wettbewerber. Diese lassen sich durch uns nicht kontrollieren und stellen externe Varianzquellen dar. Ergo gilt: Geschäftsrelevante Ziele unterliegen meist vielfältigen Varianzen.



Abb. 4-14 Geschäftsrelevante Ziele unterliegen meist Varianzen.

Lokale Optimierungen

Natürlich wird jede Organisationseinheit versuchen, ihr Ziel zu erreichen. Je reizvoller die Zielerreichung (z. B. in Form von Boni oder in Aussicht gestellter Beförderungen), desto fokussierter wird die Organisationseinheit am eigenen Ziel arbeiten.

Das hört sich zunächst wünschenswert an, kann für das Unternehmen aber außerordentlich schädlich sein. Nach Ackoff (siehe [Ackoff 2008]) führt eine lokale Optimierung der Einzelteile fast immer zu einer Suboptimierung des Ganzen. Viele Unternehmen haben beispielsweise den Vertrieb auf möglichst viele

Abschlüsse hin optimiert (z. B. durch Abschlussprovisionen). Wenn das Unternehmen die ganzen Verkäufe aber gar nicht umsetzen kann, kommt es zu Problemen (Überlastungen, Verzögerungen, Qualitätsdefizite). Diese Probleme führen wiederum zu unzufriedenen Kunden.

Die Gefahr der lokalen Optimierung existiert auch in unserem besseren MbO-Beispiel. Die Abteilungen für Konzerne und Mittelständler werden auf ihre Ziele hin optimieren. Hat die Frankreich-Abteilung aber Schwierigkeiten, Entwickler für die neuen Teams einzustellen, könnte es aus Unternehmenssicht sinnvoll sein, lokale Ziele zu opfern. Vielleicht wäre es die beste Lösung, wenn die Abteilungen *Konzerne* und *Mittelstand* einen Teil ihrer Entwickler in die Abteilung *Frankreich* transferieren und dafür in Kauf nehmen, ihre Ziele nicht zu erreichen.

Eine Organisationseinheit muss sich also stets bewusst sein, dass das eigene lokale Ziel nur Mittel zum Zweck ist: Es geht darum, mit den anderen Organisationseinheiten gemeinsam das globale Ganze zu optimieren. Daher muss jede Organisationseinheit immer auch das Gesamtziel im Auge behalten und mit benachbarten Organisationseinheiten kooperieren.

Hands-off-Management

Für Manager kann es sehr verlockend sein, sogenanntes *Hands-off-Management* zu praktizieren: Der Manager definiert mit seinen Untergebenen Ziele und muss sich dann nicht weiter mit ihnen beschäftigen. Das ist bequem, aber nicht zielführend. Den Mitarbeitern fehlt häufig die Unterstützung, die sie brauchen – z. B. in Form von Mentoring. Um diese Unterstützung geben zu können, ist ein kontinuierlicher Dialog notwendig, der durch messbare Ziele behindert werden kann.

Außerordentliche Anstrengungen

Herausfordernde Ziele führen zu außerordentlichen Anstrengungen. Das Ergebnis ist natürlich beeindruckend und jeder sieht die Hingabe und das Engagement, mit dem das Ziel erreicht wurde: Geschichten, in denen Helden geboren werden.

Dieser kurzzeitige Erfolg bringt das Unternehmen aber nicht nachhaltig weiter. Wenn das Ziel erreicht wurde, ist das Unternehmen genauso (wenig) leistungsfähig wie vorher.

Um bei immer stärkerem Wettbewerb mithalten zu können, reicht es nicht aus, einzelne herausfordernde Ziele zu erreichen. Das Unternehmen muss sich nachhaltig weiterentwickeln, sodass es immer leistungsfähiger wird – nicht durch Heldentaten, sondern durch seine inhärente Leistungsfähigkeit.

Zielkonflikte

Werden Ziele bis hinunter auf Mitarbeiterebene formuliert, geraten diese leicht in Konflikt mit Teamzielen. Insbesondere agile Teams sollen sich sehr schnell an neuen Prioritäten ausrichten. Individuelle Quartalsziele stehen dieser Neuausrichtung dann häufig im Weg. Und damit hat der Mitarbeiter auf einmal konkurrierende Ziele: sein individuelles und das Teamziel. Entweder priorisiert der Mitarbeiter eines der Ziele höher oder er zerreit sich zwischen beiden Zielen (z. B. durch Überstunden). In so einem Kontext wirken Zielsysteme wie MbO oder OKR dann tatsächlich als Störfaktoren des agilen Kernzyklus.

Innovationshemmnis

Nonaka und Takeuchi weisen darauf hin, dass Innovation eine gewisse innere Spannung benötigt, die durch Uneindeutigkeiten geschaffen werden kann (siehe [Nonaka & Takeuchi 1995]). Ein sehr klares Ziel wie »Baut ein familientaugliches Auto mit einem Marktpreis von 25.000 Euro« erzeugt wenig Innovation. Es legt den Rahmen so klar fest, dass ein typisches Familienauto dabei herauskommen wird. Ein auf den ersten Blick absurd erscheinendes Ziel wie »Familienauto auf zwei Rädern« regt hingegen die Fantasie an. Es ist eben nicht eindeutig, wie so ein Auto sinnvoll aussehen kann. Ein anderes Beispiel für ein innovationserzeugendes Ziel könnte »Suchfunktion ohne Blättern« sein, das sich für Marktplätze wie eBay, mobile.de oder ImmobilienScout24 eignen könnte. Damit würde sich dem Team die Frage stellen: Wie können wir dafür sorgen, dass die Einträge, die für den Anwender wirklich relevant sind, auf der ersten Seite erscheinen und er nicht blättern muss.

Mit solchen uneindeutigen Zielen geht einher, dass man vorher nicht sinnvoll darüber sprechen kann, ob und bis wann das Ziel erreicht werden kann. Man darf in solchen Fällen also keine Bewertung an das Ziel knüpfen.

4.3.6 Ziele ohne die MbO-Gefahren

Die oben genannten Probleme sind keine K.-o.-Kriterien für MbO. Man kann mit den Herausforderungen umgehen, wenn man sich ihrer bewusst ist:

Zuerst muss eine **passende Organisationsstruktur** nach Märkten, Produkten oder zumindest Phasen (engl. stages) geschaffen werden. Ansonsten braucht man kein Zielsystem, das dezentrale Entscheidungen unterstützt.

Wenn man seine Ziele regelmäßig überprüft und anpasst, kann man mit den **Varianzen** umgehen. Dazu muss allerdings das Definieren von Zielen zügig von

der Hand gehen. Das wird am Anfang noch nicht der Fall sein, deshalb muss in die Fähigkeit investiert werden, schnell gute Ziele zu definieren.

Wenn alle Beteiligten alle Ziele oberhalb des eigenen Ziels kennen und sie den Auftrag erhalten, gemeinsam mit anderen Teams, Abteilungen etc. auch die übergeordneten Ziele zu erreichen, können sie **lokalen Optimierungen** entgegenwirken.

MbO darf von Managern nicht dafür benutzt werden, den Dialog mit den Mitarbeitern zu vermeiden (**Hands-off-Management**). Die Grundlage aller agilen Werkzeuge, das Agile Manifest, schätzt Individuen und Interaktionen mehr als Prozesse und Werkzeuge. Das beschreibt auch den Umgang zwischen Führungskräften und ihren Mitarbeitern. Damit müssen Manager die Verantwortung für ihre Mitarbeiter akzeptieren. Die Mitarbeiter brauchen Unterstützung/Mentoring, um die Ziele zu erreichen. Manager müssen verstehen, *wie* die Mitarbeiter die Ziele erreichen (wollen), sodass sie Mentoring anbieten können. Manager müssen darüber hinaus intervenieren, wenn mit **heroischen Anstrengungen** statt mit nachhaltigen Verbesserungen gearbeitet wird.

Für Teams muss man sich Gedanken machen, welchen Wert individuelle Ziele für Mitarbeiter überhaupt haben. Ein Team zeichnet sich gerade durch gegenseitige Abhängigkeit aus: Das Ziel kann nur gemeinsam erreicht werden. Individuelle Ziele legen nahe, dass die Teammitglieder ihre Ziele alleine erreichen können und sich dadurch das Teamziel von selbst ergibt. Wäre dies der Fall, läge kein Team vor. Wenn also wirklich Teamarbeit angemessen für die Aufgabe ist, sollten keine individuellen Ziele definiert werden. Damit verschwinden natürlich die **Konflikte zwischen individuellen und Teamzielen**.

Innovation braucht **uneindeutige Ziele**, die logischerweise mit einer größeren Unsicherheit einhergehen. Es darf also niemand auf die Erreichung eines solchen Ziels festgenagelt werden. Auch hier verweisen wir erneut auf das Agile Manifest. Wenn wir tatsächlich das Reagieren auf Veränderung mehr als das Befolgen eines Plans schätzen, dann freuen wir uns mit jedem täglichen Lernschritt über eine Verschiebung und Neudefinition der Ziele. Es wäre dumm, an einem Ziel festzuhalten, das laut heutigem Wissensstand obsolet oder zumindest fragwürdig geworden ist.

Und nicht zuletzt sollte man sich folgende Frage stellen: Wenn das Unternehmensziel klar ist, inwieweit braucht es dann noch heruntergebrochene Subziele? In Konzernen wird vermutlich mehr als nur das Gesamtkonzernziel erforderlich sein, aber vielleicht müssen diese Ziele nicht so detailliert heruntergebrochen werden, wie es heute passiert?

4.4 Feedbackschleifen statt statischer Ziele

Eines der Probleme mit MbO besteht darin, dass die Ziele statisch sind. Auf der Ebene von Unternehmenszielen ist dies meistens auch sinnvoll – das Unternehmen kann sich nicht jeden Tag komplett neu erfinden, sondern braucht eine gewisse Stabilität in der Ausrichtung. Wir haben schon gesehen, dass weiter unten in der Hierarchie statische Ziele zu erheblichen Problemen führen können. Im Grund wäre ein kontinuierlicher Anpassungsprozess notwendig, der allerdings mit hohen Anpassungsaufwänden verbunden wäre.

Die Alternative sind Feedbackzyklen, die eine dynamische Steuerung erlauben. Sie basieren auf den Prinzipien der Transparenz, der direkten Kommunikation, der globalen Optimierung der Wertschöpfung und der Überprüfung und Anpassung.

4.4.1 Feedbackschleife für den Umweltschutz

Jerry Weinberg beschreibt in [Weinberg 1985] eine interessante Geschichte zu Feedbackschleifen (er schreibt selbst, dass nicht gesichert sei, dass die Geschichte wahr ist – nützlich ist sie auf jeden Fall): Zu Beginn der Industrialisierung hatten die USA erhebliche Probleme durch Umweltverschmutzung, insbesondere mit verschmutzten Flüssen. Die neu entstandenen Fabriken leiteten Chemikalien in Flüsse ein, wodurch Fischerei und Grundwasser flussabwärts gefährdet wurden (siehe Abb. 4–15).

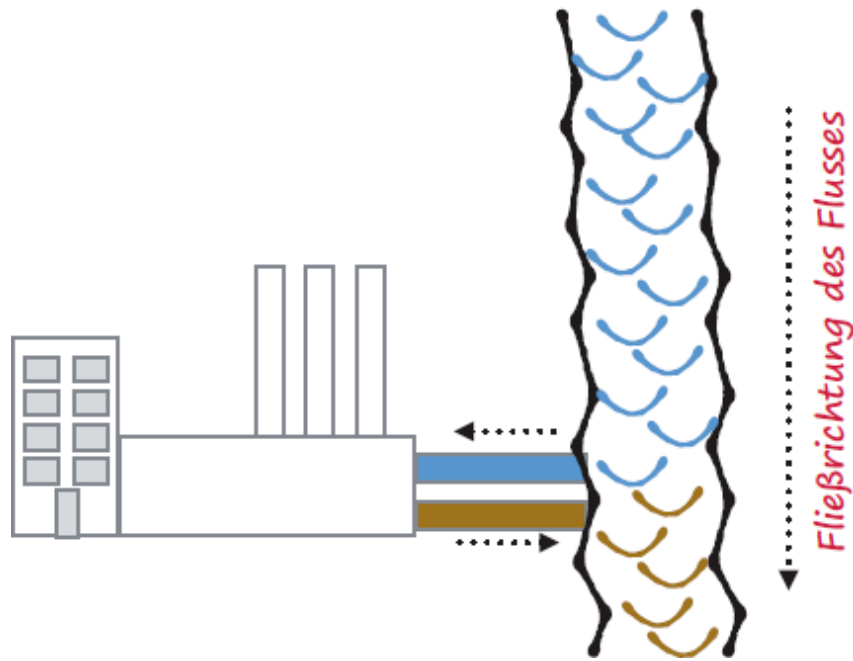


Abb. 4-15 Das Problem: Fabriken verschmutzen Flüsse.

Man setzte eine Kommission ein, die das Problem beseitigen sollte. Diese Kommission führte Interviews mit Industrievertretern, um ihre Sichtweisen und Ideen kennenzulernen. Unter anderem sprach die Kommission mit Henry Ford (dem Gründer der Ford-Autowerke). Dieser sagte, dass zur Beseitigung des Problems keine umfangreichen gesetzlichen Regelungen notwendig seien, sondern eine einzige einfache Regel ausreichen würde:

»Jeder darf so viel Wasser aus dem Fluss entnehmen, wie er braucht, und mit dem Wasser machen, was er will. Er muss lediglich die gleiche Menge Wasser flussaufwärts wieder einleiten« (visualisiert in Abb. 4-16).

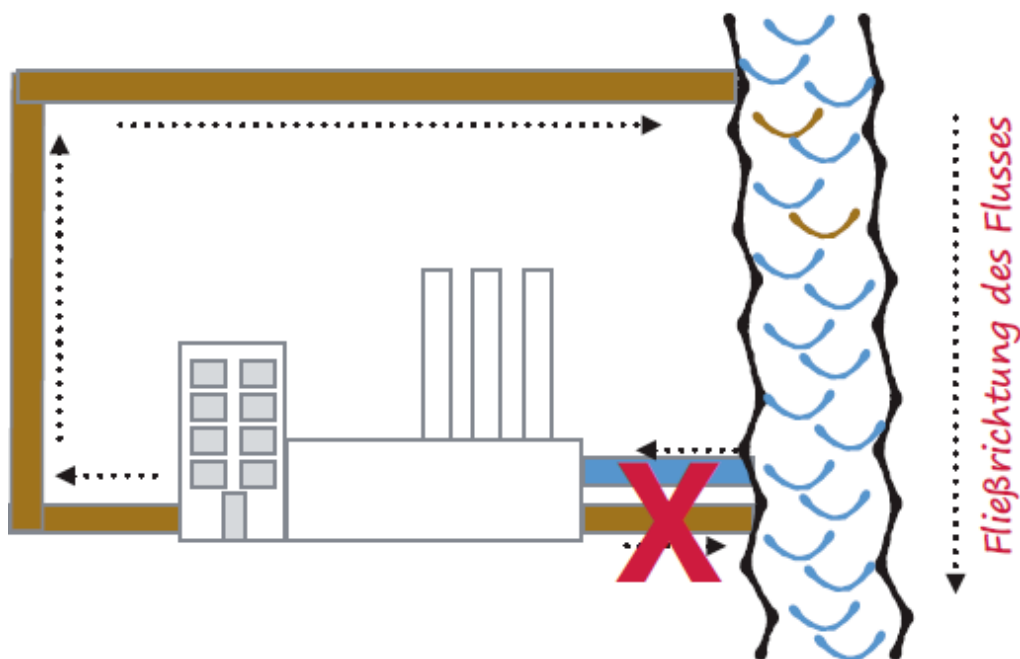


Abb. 4-16 Lösung durch Feedbackschleife

Mit der Regel würde eine einfache, aber mächtige Feedbackschleife geschaffen. Würde eine Fabrik das Wasser nicht ausreichend reinigen, würde sie selbst immer stärker verdrecktes Wasser aus dem Fluss entnehmen, das immer schlechter für die Produktion geeignet wäre. Defizite im Umweltschutz würden dem Verursacher also selbst schaden.

Fallbeispiel: Nearshore-Bugfixing

Einer unserer Kunden hatte einen externen Nearshore-Anbieter mit Wartung und Bugfixing seiner Software betraut. Das erschien sehr attraktiv, weil dadurch die hoch bezahlten Experten in Deutschland nicht durch langweilige Wartungsaufgaben demotiviert wurden und sich ganz auf die Entwicklung wertvoller neuer Features fokussieren konnten.

Im Endeffekt wurden die Feature-Teams in Deutschland immer schneller und das Nearshore-Team immer größer. Man hatte die Feedbackschleife zerstört, die es den Entwicklern in Deutschland ermöglichte, die geschaffene Qualität zu erfahren. So konnten sie dem Druck des Business nach mehr Features leicht nachgeben, indem sie schlechtere Qualität ablieferten. Diese schlechte Qualität machte sich dann an einer anderen Stelle im Unternehmen bemerkbar. Allerdings wurde das Problem dort nicht als zu schmerzhaft wahrgenommen, weil die Nearshore-Entwickler verhältnismäßig preisgünstig waren. Die Kunden litten aber unter immer mehr Fehlern.

Schließlich wurde dieser Zusammenhang erkannt und Bugs wurden den Teams zugeteilt, die sie verursacht hatten. In der Folge wurden die Feature-Teams wieder langsamer – die vorherige hohe Geschwindigkeit war geschummelt, weil sie durch Qualitätsdefizite erkauft wurde. Faktisch reduzierte sich die Geschwindigkeit der Feature-Teams auf ihre reale Geschwindigkeit, sodass auch wieder besser prognostiziert werden konnte. Außerdem gab es weniger Bugs in der Produktion, was den Kunden zugutekam. Man hatte die Feedbackschleife wieder geschlossen.

Fallbeispiel: Spesen bei it-agile

Ein weiteres Beispiel für eine Feedbackschleife findet sich in der Spesenabrechnung bei it-agile. Dort entscheidet jeder Mitarbeiter anhand einiger einfacher Regeln, wie er reist und übernachtet sowie welche beruflichen Anschaffungen (Notebooks, Handys) er tätigt. Diese Regeln lauten konkret:

- Verhalte dich wirtschaftlich sinnvoll.
- Reiche nur das ein, was rechtlich zulässig ist.
- Sorge dich um dein Wohlergehen.

Es gibt damit z. B. keine formelle Obergrenze, wie viel eine Übernachtung kosten darf. Es ist auch nicht festgelegt, ob man mit der Bahn erster Klasse fahren darf. Stattdessen sind die Spesenabrechnungen innerhalb von it-agile für alle Mitarbeiter einsehbar. Was auch immer man als Spesen einreicht, sollte man bei Nachfragen begründen können. Dadurch wird eine Feedbackschleife geschaffen: Wenn ein Kollege unangemessen hohe Ausgaben für Spesen hat, wird er entsprechendes Feedback von Kollegen bekommen und kann sein Verhalten anpassen. Dadurch bildet sich ein gemeinsames Verständnis darüber heraus, welche Ausgaben bei it-agile akzeptabel sind und welche eher nicht. Dabei gibt es immer noch Flexibilität, sodass man bei einer Messe deutlich mehr für eine Übernachtung ausgeben »darf« als zu normalen Zeiten.

4.4.2 Feedbackschleifen bei Command & Control-Strukturen

Klassische Command & Control-Strukturen schaffen auch Feedbackschleifen. Der Vorgesetzte stellt diese her. Wenn ein Mitarbeiter inakzeptables Verhalten an den Tag legt, erhält er vom Vorgesetzten entsprechendes Feedback und kann sein Verhalten anpassen (siehe Abb. 4–17).

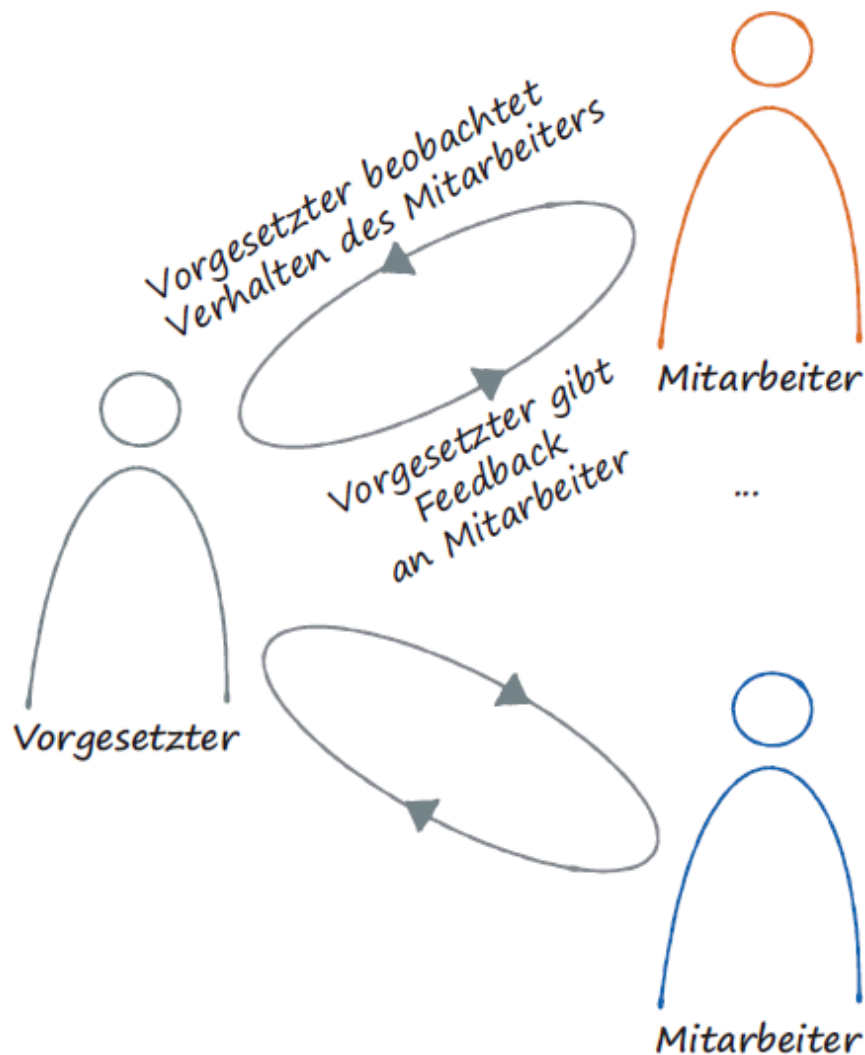


Abb. 4-17 Managerbasierte Feedbackschleifen

Das erklärt auch, warum es in der Regel nicht besonders pfiffig ist, auf dem Weg zu einem agilen Unternehmen einfach die Manager ersatzlos abzuschaffen. Man eliminiert auch die Feedbackschleifen, die es den Mitarbeitern ermöglichen, herauszufinden, was akzeptables und was inakzeptables Verhalten ist.

Allerdings können auch nicht einfach die bestehenden Strukturen beibehalten werden. Managerbasierte Feedbackschleifen bringen eine Reihe von Problemen mit sich, die Agilität behindern:

- Ein Vorgesetzter muss Feedback an mehrere Mitarbeiter geben. Oft genug ist er mit Arbeit überlastet, sodass Feedback durch den Vorgesetzten verzögert erfolgt. Je langsamer das Feedback, desto langsamer das Lernen.
- In einer Command & Control-Kultur führt das Feedback nicht zu gemeinsamem Lernen. Allenfalls lernt der Mitarbeiter durch Ausprobieren, was der Vorgesetzte für akzeptabel hält. Der Vorgesetzte lernt allerdings

nicht, was für die Arbeit des Mitarbeiters notwendig ist. Das Feedback führt also nicht zu besseren Lösungen, sondern lediglich zu Compliance.

- Dieser Fokus auf Compliance erzeugt bei den Mitarbeitern regelmäßig Angst davor, Dinge zu tun, die negatives Feedback nach sich ziehen. Sie neigen also dazu, vorher um Erlaubnis zu fragen. Dadurch versuchen sie, »Fehler« zu vermeiden. Tatsächlich ist dieses Verhalten meist aber ökonomisch schädlich, weil dadurch Wartezeiten entstehen, in denen die Arbeit nicht vorangeht.
- Durch dieses Um-Erlaubnis-Fragen wird die Autonomie der Teams beschnitten, was sich auf Motivation und Ergebnis negativ auswirkt.
- Und nicht zuletzt ist die Wirkung der Feedbackschleife hochgradig von der Person des Vorgesetzten abhängig. Der eine ist vielleicht eher streng, der andere eher nachsichtig. So entsteht für die Mitarbeiter keine verlässliche Arbeitsumgebung und bei jedem Wechsel des Vorgesetzten muss sich das Verhältnis zwischen Vorgesetztem und Mitarbeiter neu einschwingen.

4.4.3 Feedbackschleifen in einem agilen Unternehmen

In einem agilen Unternehmen versucht man die Feedbackschleifen dezentraler zu gestalten, um die oben geschilderten Nachteile zu vermeiden. Die Feedbackschleife zur Spesenabrechnung bei it-agile ist ein Beispiel dafür.

Das Nachdenken über Feedbackschleifen kann mit Causal-Loop-Diagrammen⁴ (kausale Rückkopplungsdiagramme) unterstützt werden. Sie zeigen die Wirkungen zwischen prinzipiell quantifizierbaren Eigenschaften.

So können wir im oben genannten Beispiel mit dem Bugfixing-Team folgende Überlegung anstellen: Druck auf die Feature-Teams führt zu mehr ausgelieferten Features, allerdings auch zu schlechterer Qualität, also mehr Bugs. Mehr Bugs führen dazu, dass mehr Mitglieder im Bugfixing-Team notwendig werden und dass die Kundenzufriedenheit sinkt. Unzufriedene Kunden führen zu mehr Druck auf die Entwicklungsteams, die Kunden durch neue Features glücklich zu machen. Abbildung 4-18 zeigt das zugehörige Causal-Loop-Diagramm. Ein O zeigt eine umgekehrte Wirkung an (O für Opposite). Wenn es mehr Bugs gibt, sinkt die Kundenzufriedenheit. Zwei Striche auf der Linie zeigt eine verzögerte Wirkung an. Die Kunden sind nicht direkt nach Auslieferung der neuen Version zufriedener oder unzufriedener, sondern mit einer Verzögerung. Verzögerte Wirkungen sind tückisch, weil sie zu Ergebnissen führen, deren Ursachen nicht mehr sofort ersichtlich sind. Dadurch fällt es dem Unternehmen schwer, aus diesen Wirkungen etwas zu lernen.

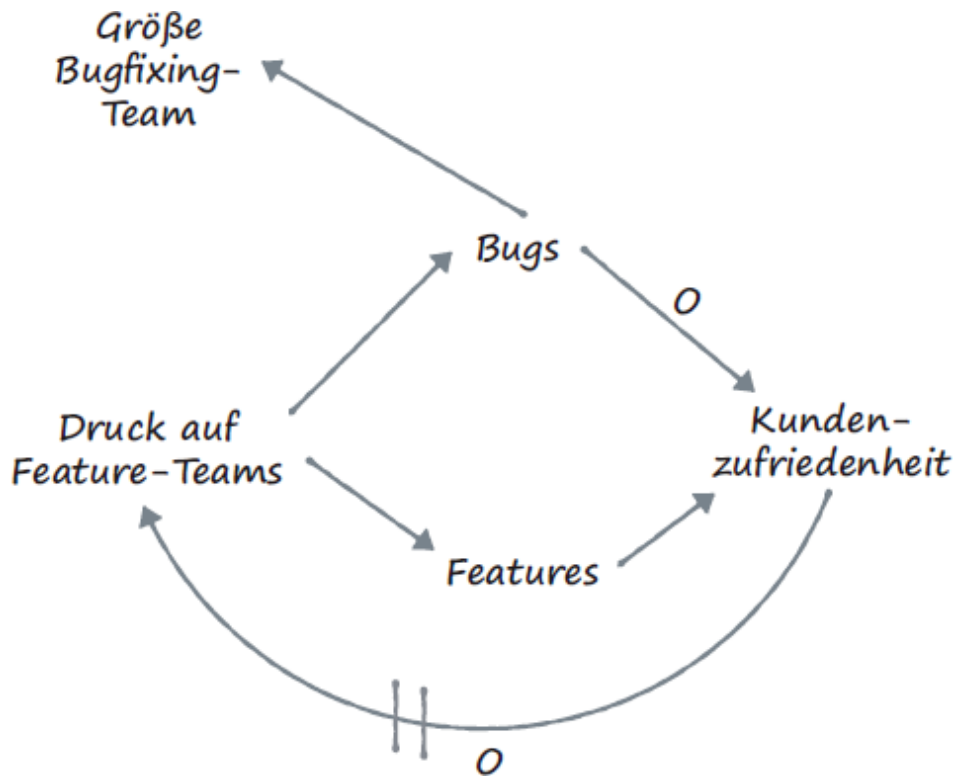


Abb. 4-18 Causal-Loop-Diagramm

An dem Diagramm kann man sehr schön erkennen, dass es einen sich selbst verstärkenden Teufelskreislauf gibt (Druck auf Feature-Teams führt zu mehr Bugs, führt zu unzufriedeneren Kunden, führt zu mehr Druck auf die Features-Teams). Es zeigt aber auch, dass prinzipiell ein Gleichgewicht möglich ist, wenn nämlich der Mehrwert der neuen Features die Kunden glücklicher macht, als der Ärger über die Bugs sie unzufrieden macht.

Wenn man jetzt wie oben geschildert das Bugfixing-Team aus der Gleichung herausnimmt, ändert sich das Causal-Loop-Diagramm wie in Abbildung 4-19 gezeigt.

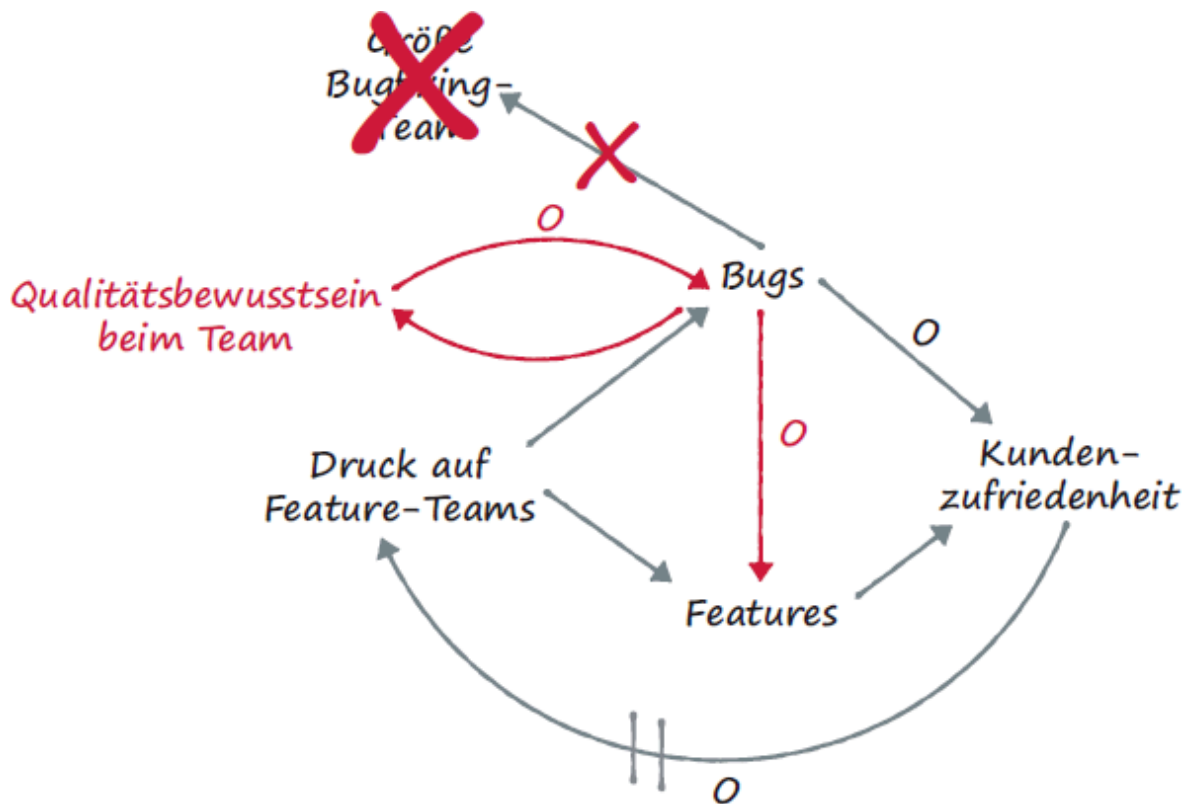


Abb. 4-19 Causal-Loop-Diagramm nach Abschaffung des Bugfixing-Teams

Ohne Bugfixing-Team gehen die Bugs direkt an die verursachenden Teams zur Beseitigung. Dadurch haben die Feature-Teams weniger Kapazität, um neue Features zu entwickeln. Allerdings erhöht sich auch das Qualitätsbewusstsein der Feature-Teams und sie können selbst den angemessenen Kompromiss aus Feature-Output und Qualität herstellen. In diesem Diagramm haben wir »Qualitätsbewusstsein beim Team« als neue Eigenschaft eingeführt, die einen Zyklus mit der Menge der Bugs enthält. Im Gegensatz zum Zyklus über die Kundenzufriedenheit ist dieser allerdings nicht verzögert und erlaubt dem Team damit direkteres und schnelleres Lernen.

4.4.4 Das Unternehmen als Organismus

Die Unternehmenssteuerung über statische Ziele folgt – meist unbewusst – dem »Unternehmen als Maschine«-Modell. Man muss »nur« die Einzelteile richtig steuern, dann wird die Gesamtmaschine das tun, was gewünscht ist.

Die Arbeit mit Feedbackschleifen entspricht eher dem »Unternehmen als Organismus«-Modell. Die einzelnen Organe »spüren« kontinuierlich, was die anderen Organe tun und brauchen, und richten sich dazu passend aus. So findet das Unternehmen durch unzählige kleine Überprüfungen und Anpassungen seinen Weg.

4.5 Übergreifende Entscheidungsfindung bei dezentralen Strukturen

Wenn man sich eine Organisationsstruktur gibt, die das Prinzip der dezentralen Entscheidungen ins Extrem treibt, kommt man z. B. zu dem oben beschriebenen Zellmodell. Damit stellt sich sofort die Frage nach übergreifenden Entscheidungen. Schließlich soll das Zentrum der Peripherie keine Vorgaben machen. Woher kommen dann aber z. B. Unternehmensziele, die Strategie oder größere Investitionsentscheidungen? Dazu gibt es zwei bewährte Ansätze: *Konsent* und *Advice-Prozess*. *Konsent* ist eine Technik, um mit Vertretern von Organisationseinheiten (z. B. Peripheriezellen) eine gemeinsame Entscheidung zu fällen. Der *Advice-Prozess* setzt einen temporären Entscheider ein mit der Verpflichtung, sich Rat an verschiedenen Stellen im Unternehmen einzuholen, bevor er entscheidet.

4.5.1 Konsent

Im *Konsent* [Westphal 2009] (absichtlich nicht *Konsens*) wird in einer Gruppe ein Vorschlag für eine Entscheidung unterbreitet, z. B. »Wir stellen ein Team zusammen, das eine Dating-Plattform für Katzen entwickelt und dafür maximal 1 Mio. Euro ausgeben darf«. Die Teilnehmer der Gruppe geben nacheinander ihre Einschätzung zu dem Vorschlag mit einer der folgenden Bewertungen ab:

- **Daumen hoch**

Ich finde den Vorschlag gut und bin dafür.

- **Daumen zur Seite**

Ich finde den Vorschlag nicht wirklich gut. Ich trage ihn aber mit, wenn der Rest der Gruppe es möchte.

- **Daumen runter**

Ich habe einen schwerwiegenden Einwand.

Solange es schwerwiegende Einwände gibt, ist der Vorschlag nicht angenommen. Jetzt versucht die Gruppe schrittweise die vorgebrachten Einwände in den Vorschlag zu integrieren, bis es keine schwerwiegenden Einwände mehr gibt.

Das Ziel ist nicht, dass alle den Daumen nach oben zeigen. Es reicht, dass es keine Daumen nach unten gibt. Die Überlegung dahinter ist, dass sich die Qualität der Entscheidung letztlich erst in der Umsetzung zeigt. Daher reicht es, wenn der

Vorschlag »Safe-to-Fail« ist, also keinen schweren Schaden anrichten kann. Dann kann man nach der Umsetzung über die Entscheidung reflektieren und ggf. über eine weitere Entscheidung nachjustieren.

Konsent führt zu einem hohen Commitment auf die Entscheidung, weil alle Beteiligten Gehör finden können und ihre Einwände integriert werden. Außerdem deckt Konsent blinde Flecken auf, die jede Einzelperson hat, und führt so zu besseren (sichereren) Entscheidungen.

Allerdings bedarf das Verfahren – so einfach es auch erscheinen mag – Moderation und Übung. Zu leicht wird der Daumen nach unten als Veto interpretiert und keine schwerwiegenden Einwände mehr diskutiert. Dann heißt es nur noch »so ein Schwachsinn, nicht mit mir – ich lege ein Veto ein« und es entsteht eine Blockadehaltung, in der im schlimmsten Fall gar nichts mehr entschieden werden kann. Der Moderator achtet darauf, dass das Gesamtbild in der Gruppe deutlich wird (z. B. indem anfänglich der Reihe nach jeder seine Einschätzung gibt), dass tatsächlich nur schwerwiegende Einwände formuliert werden und dass gemeinsam daran gearbeitet wird, die Einwände in den Vorschlag zu integrieren (und nicht einfach den Vorschlag abzulehnen).

In der Praxis ist es üblich und auch praktikabel, dass der Daumen nach unten auch dann verwendet wird, wenn man den Vorschlag noch nicht ausreichend gut verstanden hat.

4.5.2 Advice-Prozess

Beim Advice-Prozess kann prinzipiell jeder Mitarbeiter jede Entscheidung fällen (siehe [Laloux 2014]). Er muss sich lediglich ausreichend Rat bei Kollegen einholen. Die Grundidee ist hier, dass durch diesen Advice-Prozess jeder potenzielle Entscheider dieselben Informationen zur Verfügung hat und dieselben Perspektiven kennen kann. Dann spielt es am Ende keine so große Rolle mehr, wer konkret die Entscheidung fällt.

Niels Pfläging beschreibt mit dem konsultativen Einzelentscheid eine konkrete Ausprägung des Advice-Prozesses (siehe [Pfläging 2014]). Der konsultative Einzelentscheid folgt dem folgenden Schema:

1. Zuerst wird die zu fällende *Entscheidung* identifiziert und ein passender *Entscheider* wird ausgewählt. Der Entscheider kann derjenige sein, der die größte Expertise zum Thema hat, am meisten für das Thema brennt oder dem ein Interessensausgleich am ehesten zugetraut wird.

2. Der Entscheider *konsultiert* einige Personen, und zwar angemessen in Bezug auf die Auswirkung der Entscheidung. Sind nur wenige von der Entscheidung betroffen und ist der Entscheid reversibel, spricht er vielleicht nur mit zwei oder drei Mitarbeitern. Betrifft die Entscheidung große Teile der Kollegen und ist sie nur schwer wieder rückgängig zu machen, konsultiert er eine größere Anzahl von Mitarbeitern und mitunter auch Personen von außerhalb (z. B. Kunden, Partner, Experten).
3. Der Entscheider *entscheidet* auf Basis der Informationen, die er bei der Konsultation erlangt hat. Dabei ist wichtig, dass der Entscheider selbst entscheidet und diese Entscheidung auch verantwortet. Es geht nicht darum, nach einem objektiven Schema aus den Einzelmeinungen eine Gesamtmeinung zu ermitteln.
4. Der Entscheider veröffentlicht seine Entscheidung in der Firma. Er sollte deutlich machen, wen er konsultiert hat, welche Perspektiven er dadurch erlangt hat und natürlich wie die Entscheidung aussieht.
5. Die Kollegen akzeptieren die Entscheidung (bis sie ggf. durch eine neue explizite Entscheidung geändert oder aufgehoben wird) und hegen keinen Groll gegen den Entscheider (üben sich in Vergebung). Jeder weiß, dass es perfekte Entscheidungen nicht gibt und der Entscheider sein Bestes getan hat.
6. Es wird gemeinsam über die Entscheidung reflektiert und es wird gemeinsam gelernt, wie es nächstes Mal noch besser funktionieren kann.

Abbildung 4–20 visualisiert den Ablauf grafisch.

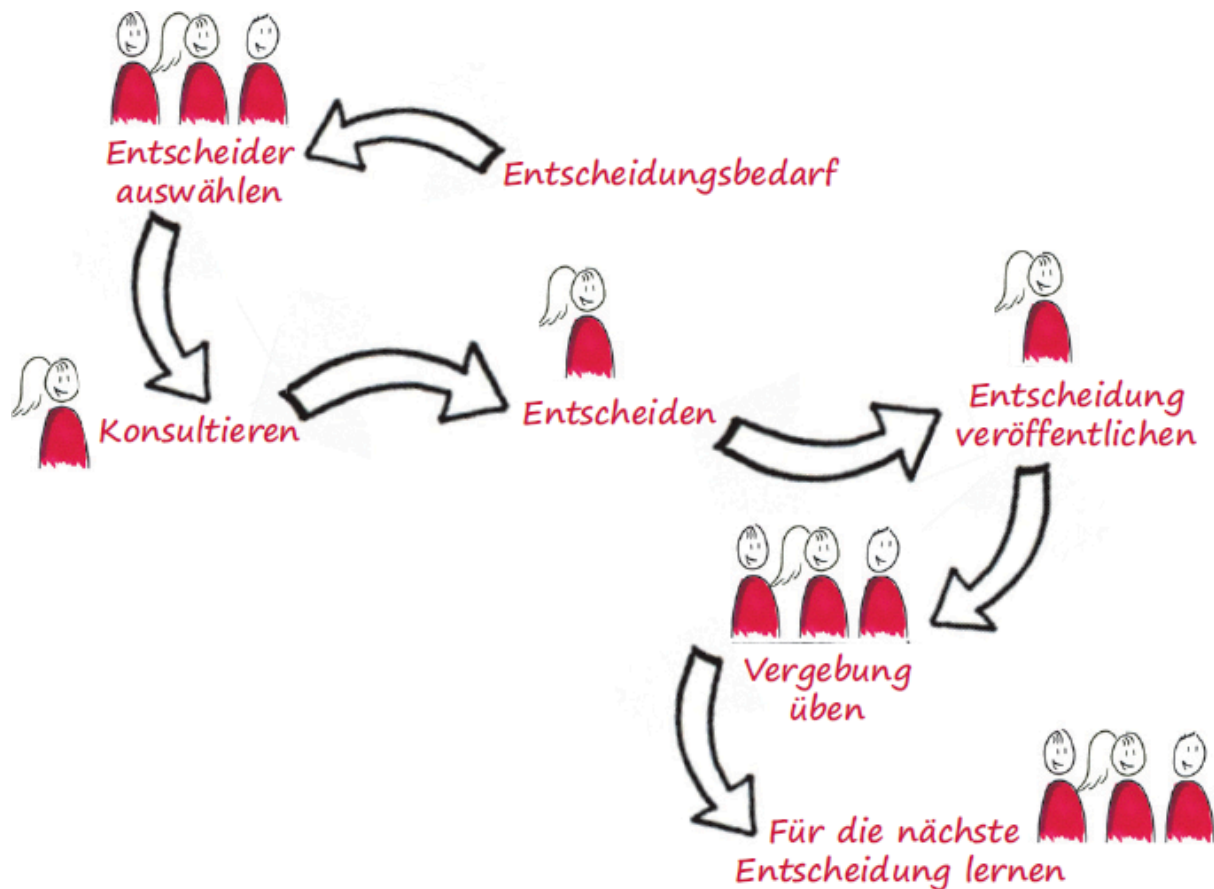


Abb. 4-20 Ablauf eines konsultativen Einzelentscheids

Fallbeispiel: Einführung des konsultativen Einzelentscheids bei it-agile

Bei it-agile haben wir immer schon viel Wert auf dezentrale Strukturen und eine große Autonomie von Mitarbeitern gelegt. Das hatte irgendwann zur Folge, dass wir bei sehr strittigen Themen nicht mehr entscheidungsfähig waren. Wir entschieden daher im Jahr 2013, den konsultativen Einzelentscheid auszuprobieren (siehe auch [Brandes et al. 2014]). Wir wollten lernen, ob und wie er unser Problem adressieren könnte.

Für unser Experiment zum konsultativen Einzelentscheid hatte die Geschäftsführung drei Entscheidungen vorbereitet. Als Erfolgskriterium für das Experiment definierten wir, dass wir *bessere Entscheidungen mit größerem Commitment* der Mitarbeiter bekommen. Die drei Entscheidungen stellten wir in einer Plenumsveranstaltung vor und aus der Diskussion unter den Mitarbeitern entstand spontan noch eine vierte Entscheidung, die wir zu unserer Liste hinzufügten:

1. Wollen wir Office-Fridays einführen und wie stellen wir eine ausreichend große und kontinuierliche Beteiligung sicher?

2. Wie wollen wir in Zukunft den fachlichen Austausch der Mitarbeiter untereinander gestalten?
3. Wollen wir für 2014 eine generelle Gehaltsanhebung für alle Mitarbeiter und wie hoch sollte diese ausfallen?
4. Welches inhaltliche Thema wollen wir als nächstes Schwerpunktthema in der Firma angehen?

Anschließend sammelten wir Vorschläge für die Entscheider. Das funktionierte schnell und reibungslos. Wir setzten für die Entscheidungen eine Timebox von zwei Monaten. Wäre es nicht kurz vor Weihnachten gewesen, hätten wir eine Timebox von einem Monat gewählt. Nach Ablauf der zwei Monate waren drei der vier Entscheidungen gefällt. Die Entscheider stellten die Entscheidungen und den Weg dahin im Plenum vor. Die beiden Entscheider zu den Themen 1 und 2 hatten sich eng untereinander abgestimmt, eine Reihe von Kollegen persönlich konsultiert, bei anderen Unternehmen nachgefragt, wie sie es machen, und schließlich ein gemeinsames Modell beschlossen. Der Entscheider zum Thema 3 hatte eine Umfrage unter allen Mitarbeitern durchgeführt.

Der Entscheider zum Thema 4 konnte keine Entscheidung fällen. Aus der Darstellung der Ursachen konnten wir aber viel über konsultative Einzelentscheidungen lernen und unser Verfahren anpassen. So haben wir im konkreten Fall gelernt, dass die Entscheidung zu groß und zu früh gewesen wäre. Wir legten daraufhin fest, dass es auch in Ordnung ist, wenn der Entscheider das Thema der Entscheidung ändert. So wurde aus der Entscheidung zur Festlegung des Schwerpunktthemas die Entscheidung, im April über das aktuelle Schwerpunktthema zu reflektieren und dann zu entscheiden.

Wir fragten anschließend, ob die Mitarbeiter das Experiment zum konsultativen Einzelentscheid erfolgreich fanden oder nicht. Ca. 80 % der Mitarbeiter meinten, dass die Entscheidungen besser geworden seien und das Commitment höher sei. Die restlichen 20 % waren der Meinung, dass es zumindest nicht schlechter geworden sei. Also beschlossen wir, den konsultativen Einzelentscheid in den »Regelbetrieb« zu überführen.

Das ganze Verfahren hat bei it-agile formale Lücken, die groß wie Scheunentore sind. So gibt es kein formelles Kriterium, wann konsultativer Einzelentscheid angewendet wird und wann Konsent in der Gesamtgruppe versucht wird. Außerdem haben wir keine Festlegung, wer oder wie viele

notwendig sind, um einen Entscheider zu bestimmen. Das stellte bisher aber kein echtes Problem dar. Und im Zweifel wird ein Entscheider eingesetzt, um eine Klärung herbeizuführen.

Die positiven Erfahrungen mit dem konsultativen Einzelentscheid setzen sich bei it-agile bis heute fort. Bei it-agile wurde bzw. wird das Verfahren z. B. für diese Fragestellungen verwendet:

- Sollen wir den Mietvertrag für das Büro verlängern oder ein neues Büro anmieten?
- Wie wollen wir mit einem Team mit Problemen umgehen? Kann es seine Probleme selbst lösen oder braucht es eine Intervention von außen? Welche? (Das kann so weit gehen, dass die Teamzusammensetzung geändert oder das Team ganz aufgelöst wird.)
- Wollen wir einen neu eingestellten Mitarbeiter nach der Probezeit übernehmen?

4.5.3 Das Unternehmen verstehen

Damit jeder Mitarbeiter weitreichende Entscheidungen treffen kann, muss es ein breites Verständnis darüber geben, wie das Unternehmen insgesamt Wert schöpft. Dazu müssen die Mitarbeiter mindestens verstehen,

- wer die Kunden des Unternehmens sind,
- welche Probleme der Kunden das Unternehmen löst,
- mit welchen Produkten und Dienstleistungen das Unternehmen diese Kundenprobleme löst,
- wie das Unternehmen die Produkte entwickelt und die Dienstleistungen erbringt und
- wie das Unternehmen dabei profitabel ist.

Diese Kenntnisse stellen sich nicht über Nacht ein. Die Situation ist aber auch nicht so kompliziert, dass nur eine kleine Elite sie verstehen kann. Die Mitarbeiter müssen in diesen Fragestellungen ausgebildet werden.

Ricardo Semler schreibt, dass bei SemCo auch »einfache« Produktionsmitarbeiter darin ausgebildet wurden, die gesamtwirtschaftlichen Zusammenhänge des Unternehmens zu verstehen (siehe [Semler 2001]).

In seiner Autobiographie [Werner 2015] schildert Götz Werner im 12. Kapitel die Wertbildungsrechnung von dm-drogerie markt als Element, um Transparenz für

Mitarbeiter herzustellen. »Jeder Mitarbeiter soll verstehen, wann seine Initiative für andere von Nutzen ist und wann nicht.« Die Wertbildungsrechnung ist ein innerbetriebliches Werkzeug, um ein gegenseitiges »Verständnis für die Leistungen und Prozesse des anderen [zu] entwickeln ...« [Werner 2015, S. 223]. Laut Götz Werner ist sie für eine Solidargemeinschaft ein besseres Werkzeug als eine Gewinn- und Verlustrechnung, weil diese Denkkategorien für die Unternehmensentwicklung bremsend wirken können.

Solche Ausbildungsmaßnahmen werden sich nicht nur dadurch auszahlen, dass besser dezentral entschieden wird. Die Mitarbeiter werden auch mehr Möglichkeiten sehen, sich im Firmensinne zu engagieren und zu Verbesserungen im Sinne der Kunden und des Unternehmens beizutragen.

4.5.4 Bewertung und Vergleich von Konsent und Advice-Prozess

Konsent und Advice-Prozess ist gemein, dass sie Formen der partizipativen Entscheidungsfindung sind. Sie basieren auf der Annahme, dass prinzipiell überall im Unternehmen für die Entscheidung relevante Informationen und Perspektiven vorhanden sein können.

Weiterhin gehen beide Ansätze davon aus, dass nicht die perfekte Lösung gefunden werden muss oder kann und dass »Safe-to-Fail« ausreicht. Wenn sich in der Praxis herausstellt, dass die Entscheidung ungünstig war, kann über eine neue Entscheidung nachjustiert werden.

Unterschiedlich ist, dass im Konsent gefundene Lösungen immer von allen mitgetragen werden. Das ist beim Advice-Prozess nicht notwendig. Daher führt der Advice-Prozess bei sehr strittigen Themen meist schneller zu einem Ergebnis. Außerdem ist er auch für »harte« Entscheidungen geeignet, bei denen der Konsent dazu führen *kann*, dass die Entscheidung verwässert wird.

Bei it-agile wird der Advice-Prozess z. B. bei der Gehaltsfindung der Mitarbeiter und bei Entlassungen eingesetzt. In beiden Fällen erscheint es zu langwierig, eine Entscheidung per Konsent zu finden.

4.6 Neue Rolle für Führungskräfte

Klassische Führungsstrukturen sind vor dem Hintergrund klassischer Arbeitsorganisation entstanden. Diese hierarchischen Führungsstrukturen passen nicht zu dezentralen Organisationen und teambasierter Arbeitsorganisation, z. B. mit Scrum.

Das Weglassen von Führung funktioniert allerdings auch nicht. Es ist also ein neues Paradigma für Mitarbeiterführung notwendig.

Dieser Abschnitt beschreibt, warum die klassische Führungsstruktur in der agilen Welt problematisch ist, und skizziert Ansätze, die besser funktionieren. Es gibt (zumindest bisher) nicht die eine richtige Lösung für Mitarbeiterführung in der agilen Welt. Es existieren aber verschiedene Ansätze, die sich bewährt haben. Diese Ansätze werden beschrieben und gegenübergestellt.

4.6.1 Klassische Mitarbeiterführung

In der klassischen Welt werden alle Führungsaufgaben über eine hierarchische Struktur abgewickelt (siehe Abb. 4-21). Der Geschäftsführer führt die Abteilungsleiter und die Abteilungsleiter führen ihre Mitarbeiter.

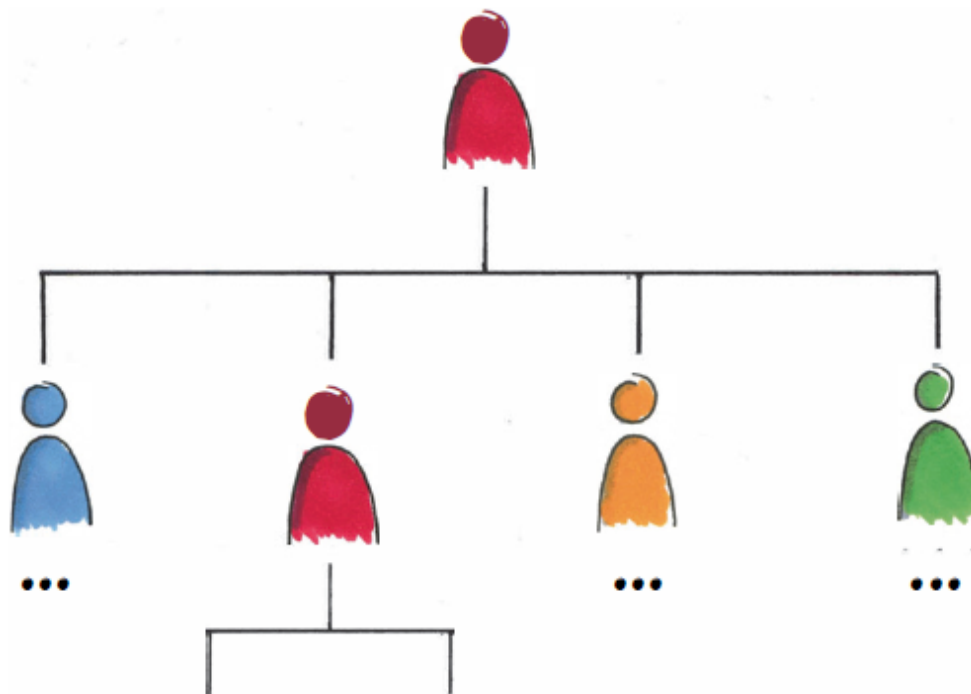


Abb. 4-21 Klassische Hierarchie

Führung erfolgt also top-down und folgt dem Paradigma von »Teile und herrsche«. Der Vorgesetzte zerlegt das Gesamtthema in einzelne Teile und gibt diese Teile an seine Untergebenen weiter. So wird das Gesamtthema immer weiter zerlegt, bis kleine Arbeitspakete bei den Mitarbeitern auf der unteren Ebene ankommen.

Wenn ein Mitarbeiter seine Arbeit erledigt hat, meldet er »Vollzug«: Er erstattet seinem Vorgesetzten Bericht. (Daher spricht man auch häufig von Reporting-Lines). Dieses Modell basiert deutlich auf Command & Control.

Die Abteilungsstruktur folgt in den meisten Unternehmen dem Paradigma funktionaler Zergliederung. Typische Abteilungen sind: Marketing, Vertrieb, Entwicklung, Qualitätssicherung, Service.

Für diese Struktur wurden die einzelnen Bestandteile der Wertschöpfungskette identifiziert und einzeln für sich optimiert. Dem liegt der Gedanke zugrunde, dass eine Optimierung der Einzelteile zu einer Optimierung des Ganzen führt. Das klingt plausibel und einfach, ist aber leider falsch (siehe dazu [Ackoff 2008]): Eine Optimierung der Einzelteile führt fast immer zu einer Suboptimierung des Ganzen.

Fachliche und disziplinarische Führung

Etwas neuer ist die Unterscheidung zwischen fachlicher und disziplinarischer Mitarbeiterführung.

Wikipedia

(<https://de.wikipedia.org/wiki/Disziplinarvorgesetzter>) definiert:

Bei den Vorgesetzten unterscheidet man zwischen **Disziplinarvorgesetzten** und **Fachvorgesetzten**. Ihnen gemeinsam ist, dass sie als Führungskräfte die Befugnis besitzen, Personalführung über ihnen unterstellte Mitarbeiter wahrzunehmen. Sie unterscheiden sich jedoch nach dem Inhalt der Führungsaufgabe. Während Fachvorgesetzte im Rahmen eines bestimmten Fachgebiets oder Arbeitsgebiets über alle zur Aufgabenerfüllung notwendigen Handlungen ihrer Mitarbeiter entscheiden und entsprechende Weisungen erteilen dürfen, sind Disziplinarvorgesetzte mit Disziplinarrechten ausgestattet. Es kann daher vorkommen, dass ein Mitarbeiter sowohl einen Disziplinar- als auch einen Fachvorgesetzten hat. Der Disziplinarvorgesetzte kann zugleich auch Fachvorgesetzter, der ausschließliche Fachvorgesetzte aber niemals Disziplinarvorgesetzter sein.

Häufig wird diese Differenzierung so umgesetzt, dass der im Organigramm visualisierte Vorgesetzte der disziplinarische Vorgesetzte ist: Abteilungsleiter, Gruppenleiter etc. Der fachliche Vorgesetzte ist z. B. ein Projektleiter, der eine Gruppe für die Zeit des Projektes fachlich führt.

4.6.2 Probleme klassischer Führung in einer dynamischen Welt

Das beschriebene Modell klassischer Führung ist nicht per se schlecht oder fehlerhaft. Das Modell ist in sich schlüssig und vielfach erfolgreich angewendet worden.

In einer zunehmend dynamischen Welt stößt das Modell allerdings an seine Grenzen. Auch ohne agile Entwicklung haben Unternehmen längst festgestellt, dass immer häufiger interdisziplinäre Projekte notwendig werden. Die starren Kommunikations- und Kooperationsstrukturen der funktionalen Abteilungsstrukturen müssen in dynamischen Umfeldern situativen Kommunikations- und Kooperationsstrukturen weichen. Das können interdisziplinäre Projekte viel besser. Die Unterscheidung zwischen disziplinarischer und fachlicher Führung versucht, genau diesem Umstand Rechnung zu tragen. Der fachliche Vorgesetzte (z. B. Projektleiter) führt ein interdisziplinäres Team.

Matrixorganisation

Werden disziplinarische und fachliche Führung voneinander getrennt, wird die Matrixorganisation faktisch unvermeidlich (siehe Abb. 4-22).

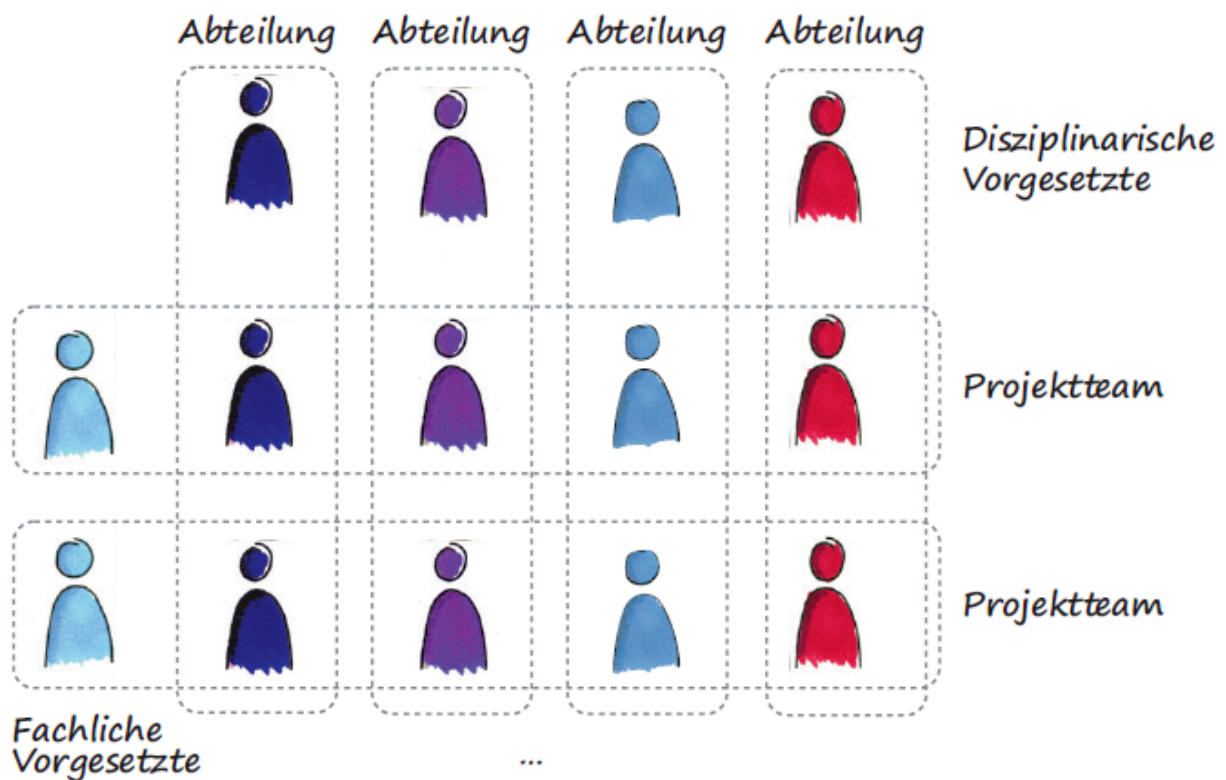


Abb. 4-22 Matrixorganisation aus Abteilungen und Projekten

Diese Struktur ist unabhängig von agiler Arbeitsweise als problematisch erkannt worden. Wenn der disziplinarische Vorgesetzte Anweisungen an seine Mitarbeiter gibt, stehen diese mitunter in Konflikt zu den Ansagen des fachlichen Vorgesetzten. Außerdem kann der disziplinarische Vorgesetzte kaum noch einschätzen, was seine Mitarbeiter im Projektteam leisten, und bekommt Schwierigkeiten mit der Leistungsbeurteilung.

Alle diese Probleme sind aushaltbar, wenn die Arbeit in Projektteams die Ausnahme ist. In der agilen Welt werden sie jedoch zum Regelfall.

4.6.3 Supporting Lines statt Reporting Lines

Im Unternehmen muss Wissen transferiert werden (vor allem von Managern an Mitarbeiter bzw. Teams) und die Mitarbeiter benötigen ggf. Unterstützung bei der Erledigung ihrer Aufgaben. In diesem Sinn sollte man in der agilen Welt von *Supporting Lines* statt *Reporting Lines* sprechen. Die klassische Hierarchie wird damit quasi auf den Kopf gestellt (siehe Abb. 4-23). Bei it-agile gab es mal den scherzhaft gemeinten Ausspruch der Geschäftsführung: »Die da unten. Mit mir können sie es ja machen.«

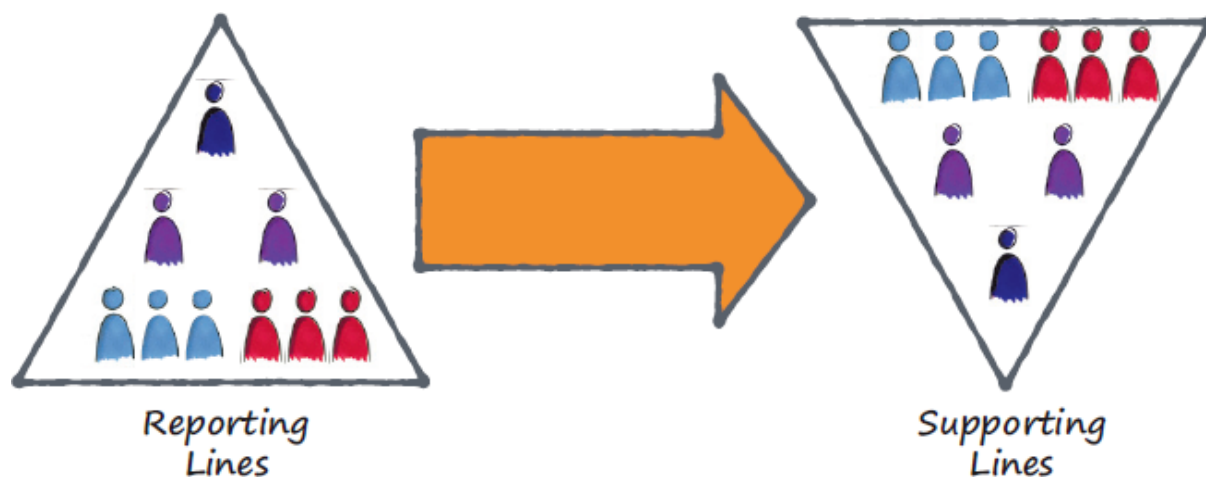


Abb. 4-23 *Supporting Lines statt Reporting Lines*

Dieses Kippen der Hierarchie wird auch *Reverse Accountability* genannt. Es sind nicht mehr die Mitarbeiter gegenüber den Managern dafür verantwortlich, dass sie z. B. ein bestimmtes Arbeitspensum schaffen. Stattdessen sind umgekehrt die Manager gegenüber ihren Mitarbeitern verantwortlich, dass diese störungsfrei arbeiten können.

Wie das konkret funktionieren kann, zeigt die indische Firma HCL mit mehreren Tausend Mitarbeitern. Dort bewertet jeder Mitarbeiter seinen Manager und den Manager des Managers und die Bewertungen sind öffentlich im Unternehmen einzusehen.

Wenn ein Mitarbeiter in seiner Arbeit behindert wird (z. B. weil Arbeitsmittel fehlen, eine andere Abteilung ungünstige Vorgaben macht etc.), erstellt er für diese Störung ein Ticket in einem internen Ticketsystem. Dieses Ticket geht an seinen Manager, der es zu bearbeiten hat. Schließen kann das Ticket nur der Mitarbeiter, der es erstellt hat. Der Manager muss also in der Regel das

persönliche Gespräch mit dem Mitarbeiter suchen, um das Problem zu verstehen und ggf. die Lösung zu erläutern. Genauso kann der Manager dem Mitarbeiter auch zusätzliche Hintergrundinformationen geben, sodass dieser z. B. Entscheidungen besser versteht. Wird das Ticket nicht binnen 24 Stunden vom Mitarbeiter geschlossen, wird es automatisch an den Manager des Managers eskaliert.

Diese Sichtweise passt sehr gut in das Bild, das wir am Kapitelanfang gezeichnet haben. Das Team und damit die Mitarbeiter stehen im Vordergrund. Sie arbeiten wertschöpfend. Die Aufgabe von Führungskräften ist, diese wertschöpfende Arbeit störungsfrei zu ermöglichen.

4.6.4 Verteilte Führung

Die Diskussion aus dem letzten Abschnitt um die gekippte Hierarchie stellt die Frage nach dem Selbstverständnis von Managern. In diesem Abschnitt fragen wir nach dem Verständnis von Führung und Management im Unternehmen. In vielen modernen Unternehmen lässt sich nämlich beobachten, dass Führung und Management auf vielfältige Art und Weise *verteilt* werden.

Alleine Scrum verteilt viele der klassischen Führungsaufgaben auf die drei Scrum-Rollen:

- Der Product Owner sorgt dafür, dass das Richtige getan wird (Produktvision, Priorisierung der Produktvision).
- Das Entwicklungsteam entscheidet darüber, wie die Arbeit konkret organisiert wird.
- Der Scrum Master sorgt dafür, dass das Team effizient arbeiten kann (z. B. durch Hindernisbeseitigung).
- Das Entwicklungsteam sorgt zusammen mit dem Scrum Master dafür, dass bei der Arbeit weder Über- noch Unterlast entstehen.
- Der Scrum Master arbeitet darauf hin, dass die Aufgabenerledigung optimiert wird (natürlich zusammen mit dem Team).

Es gibt also einen deutlichen Trend weg von der Bündelung der Führungsaufgaben hin zu einer Verteilung auf unterschiedliche Rollen und Personen. Diese Denkweise kann man auch benutzen, um über Führungsaufgaben nachzudenken, die in Scrum nicht thematisiert sind, wie z. B.:

- Genehmigung von Urlaubsanträgen
- Festlegung von Arbeitszeiten

- Langfristige Weiterentwicklung von Mitarbeitern
- Einstellungen
- Entlassungen
- Gehaltsfestlegungen

Im Grunde kann man für jede dieser Aufgaben einzeln prüfen, wie sie erfüllt werden kann, sodass das Team bei seiner wertschöpfenden Aufgabe möglichst wenig gestört wird. Wir haben in der Praxis vielfältige Ansätze dafür gesehen:

- Über den Urlaub entscheidet das Team selbst (und achtet dabei darauf, bestimmte Rahmenbedingungen einzuhalten, z. B. Verfügbarkeit von Support für die produktive Software).
- Über Einstellungen entscheidet das Team.
- Bei Einstellungen redet das Team auf Augenhöhe mit.
- Teams entscheiden darüber, ob Teammitglieder das Team verlassen müssen. Diese setzen sich dann auf die Ersatzbank und sind für andere Teams verfügbar.
- Die langfristige Weiterentwicklung der Teammitglieder liegt beim Scrum Master.
- Die langfristige Weiterentwicklung der Teammitglieder liegt im Team.
- Die langfristige Weiterentwicklung der Teammitglieder liegt in selbstselektierten Peergroups.
- Der Scrum Master verantwortet die Mitarbeiterzufriedenheit.
- Ein Feelgood-Manager verantwortet die Mitarbeiterzufriedenheit.
- Das Team verantwortet die Mitarbeiterzufriedenheit.

Warum sollte so eine lange Liste von Funktionen auch unbedingt in nur einer Person gebündelt werden? Mitunter wird argumentiert, dass es »so schön einfach« wäre. Das ist definitiv der Fall, aber es ist das falsche Kriterium. Es geht nicht darum, ob etwas einfach im Organigramm aussieht, sondern darum, ob es wirksam ist. In einer agilen Welt ist die Bündelung der oben genannten Funktionen in *einer* Vorgesetzten-Position meist nicht besonders wirksam.

4.6.5 Situative Führung

Wenn Führungsaufgaben immer breiter verteilt werden, ist eine weitere Flexibilisierung des Führungskonzeptes angesagt: weg von festen Positionen hin zu flexiblen Rollen. Wenn Manager immer nur Manager sind, Product Owner immer nur Product Owner und Scrum Master immer nur Scrum Master, haben wir eine Struktur, die zwar besser zu agilem Arbeiten passt als die vorherige. Die

Organisationsstruktur ist aber ähnlich unflexibel wie bisher. Wenn wir feststellen, dass wir gerade jetzt weniger Product Owner brauchen, sind Product Owner »übrig«. Wenn wir feststellen, dass wir dauerhaft weniger Product Owner brauchen, sind diese sogar dauerhaft überflüssig. Jetzt haben wir einen schwierigen und langwierigen Prozess vor uns, mit dieser Situation umzugehen. Dieser Prozess stellt eine ernsthafte Störung im Unternehmen dar, die in der Regel auch auf die wertschöpfenden Teams ausstrahlt. Spätestens dann, wenn Product Owner übrig sind, ein neu gegründetes Team aber einen Scrum Master benötigt, zeigt sich die Struktur von ihrer schlechten Seite.

Viel eleganter ist es, nur mit temporären Rollen zu arbeiten. Für einen bestimmten Kontext (z. B. ein Projekt, ein Team, eine Entscheidung) werden eine oder mehrere Rollen definiert und besetzt. Nach Ende des Kontextes entfällt die Rolle und auch zwischendurch kann der Rolleninhaber wechseln, wenn das sinnvoll erscheint. Der oben beschriebene Einzelentscheider übernimmt für die konkret anstehende Entscheidung eine Führungsrolle im Unternehmen. Für die nächste Entscheidung kann dies eine andere Person sein.

Auf diese Art und Weise funktionieren übrigens viele der Unternehmen, denen man nachsagt, sie hätten keine Hierarchie oder kein Management. Beides stimmt streng genommen nicht. Wird für ein Projekt ein Product Owner benannt, entsteht dadurch bzgl. der fachlichen Ausrichtung eine temporäre Hierarchie und der Product Owner übernimmt einen Teil der Managementaufgaben. Es gibt allerdings keine feste Hierarchie mit festen Managerpositionen.

Die Idee, mit temporären Rollen statt fester Positionen zu arbeiten, ist bereits einige Jahrzehnte alt und hat sich im Rahmen des Soziokratie-Modells auch für große Unternehmen bewährt [Bockelbrink & Priest 2017].

Mit diesem Ansatz, Führungsfunktionen auf temporäre Rollen zu verteilen, wird übrigens jedes klassische hierarchische Organigramm vollkommen nutzlos.

Fallbeispiel: Gehalts-Checker bei it-agile (von Stefan Roock)

Der it-agile-Arbeitsvertrag kennt nur eine Position: Mitarbeiter. Es gibt keine vertraglich festgelegten Stellen mit zugehörigen Stellenbeschreibungen. Natürlich sprechen wir bei der Einstellung über das intendierte Tätigkeitsfeld. Es ist aber auch klar, dass sich dieses temporär oder dauerhaft ändern kann.

Für Gehaltserhöhungen sind bei it-agile sogenannte Gehalts-Checker verantwortlich, die jährlich neu gewählt werden (siehe Abschnitt 4.7.3). Dadurch fühlen sich die Mitarbeiter bzgl. ihrer Gehälter keinen Willkür-Entscheidungen auf Dauer ausgeliefert. Sollte ein Gehalts-Checker seine

Aufgabe aus Mitarbeitersicht nicht gut erfüllen, wird er einfach in der nächsten Runde abgewählt.

Außerdem können wir auf diese Weise die Gehalts-Checker jeweils so wählen, wie es für die anstehenden Herausforderungen angemessen ist. Wenn der Prozess zur Gehaltsfindung als ausreichend gut empfunden wird und »nur« durchgeführt werden muss, sind andere Charaktere bei den Gehalts-Checkern nützlich, als wenn große Veränderungen im Prozess anstehen.

4.6.6 Ausbildung

Wenn man Ideen entwickelt hat, wie Führung in Zukunft anders organisiert werden kann, sollte man nicht erwarten, dass diese Ideen per Proklamation Wirklichkeit werden.

Das Mutual-Learning-Modell von Schwarz [Goodfellow et al. 2017] zeigt eine Fußangel auf, in der sich Manager und Mitarbeiter bei agilen Transitionen viel zu oft verfangen. Die Manager bekommen vom Scrum Master, von Coaches und ihren eigenen Mitarbeitern zu hören, dass sie sich heraushalten sollen. Also verfallen sie in Hands-off-Management und ziehen sich gänzlich aus dem Geschehen zurück. Allerdings wissen Manager Dinge, die die Teams nicht wissen und daher auch nicht bei der Entscheidungsfindung berücksichtigen können (siehe Abb. 4-24). Die Folge sind schlechte Teamentscheidungen. Wenn Manager dies feststellen, kehren sie häufig wieder zum alten Modell zurück, in dem sie für das Team entscheiden.

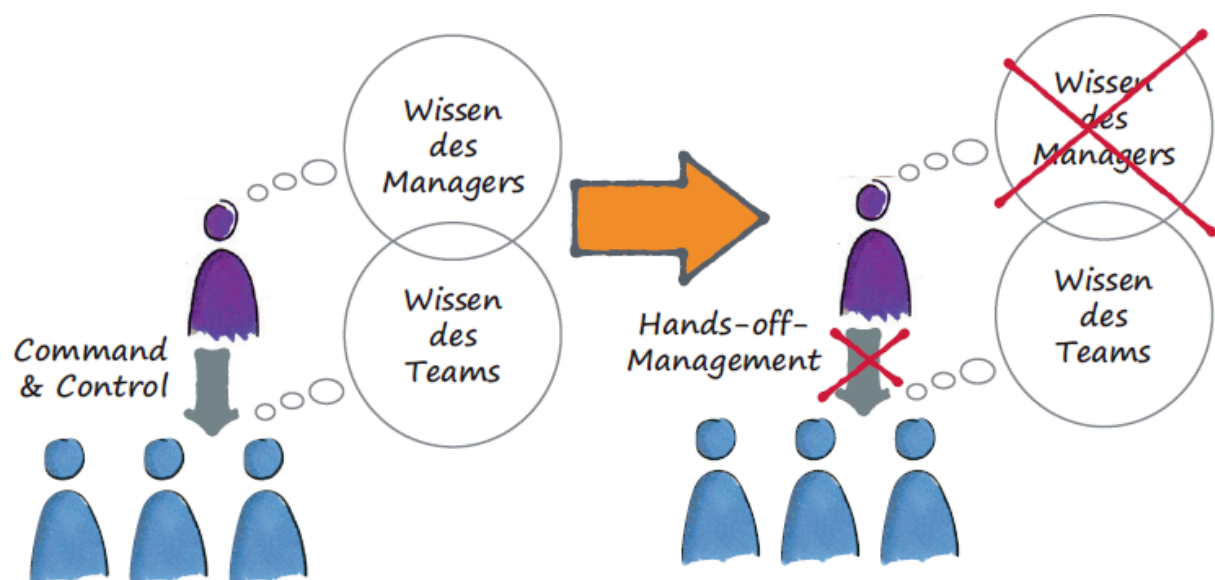


Abb. 4-24 Falsch verstandene Agilität: Hands-off-Management

Man kann nicht erwarten, dass z. B. Teammitglieder anspruchsvolle Führungsaufgaben einfach so aus dem Stand heraus meistern. Sie haben es in der Regel nicht gelernt. Die Mitarbeiter müssen geschult und in der Praxis betreut werden. Die Betreuung in der Praxis können die jetzigen Führungskräfte übernehmen – sie haben die Führungsfunktionen schließlich bisher wahrgenommen und wissen, worauf zu achten ist. Sie werden zu Mentoren ihrer Mitarbeiter. Durch diese Kooperation lernt nicht nur das Team vom Manager, sondern auch der Manager vom Team. Die Wissensbereiche von Manager und Team wachsen und beide können bessere Entscheidungen fällen (siehe Abb. 4–25).

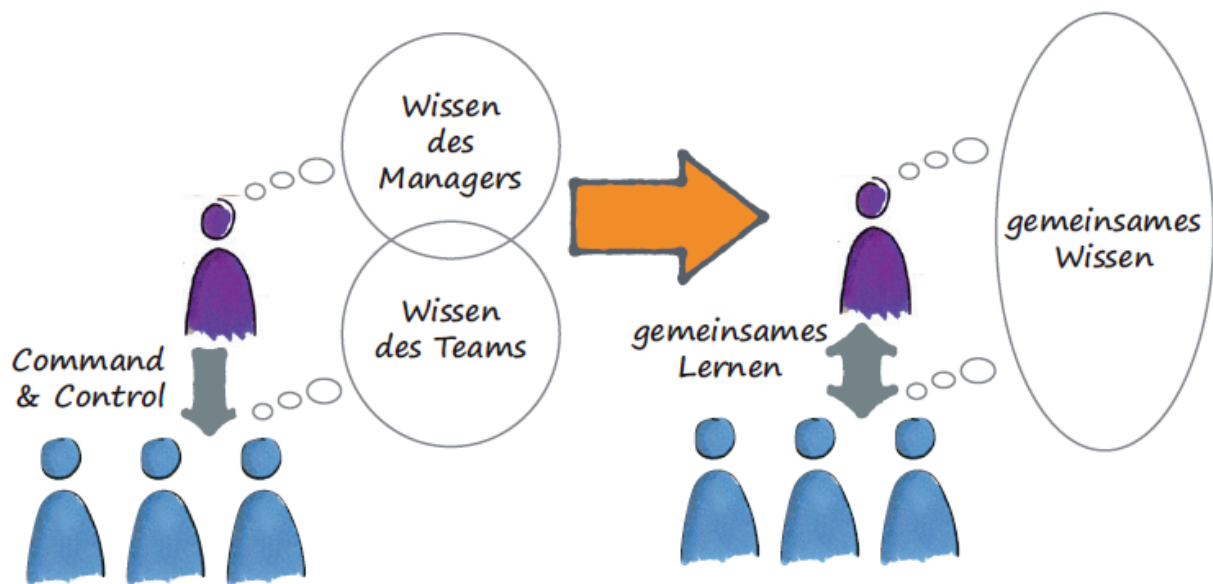


Abb. 4–25 Gegenseitiges Lernen

Diesen Weg des gegenseitigen Lernens geht Toyota. Die Manager der Fabrikarbeiter sehen sich primär als Mentoren. Wenn ein Problem in der Produktion entsteht, wird der Manager nicht einfach auf das Problem hinweisen. Er wird mit dem oder den Mitarbeitern arbeiten, sodass diese das Problem und seine Ursache selbst erkennen und dann auch selbst beseitigen können. Sie lernen über das Mentoring aber nicht nur, wie sie das akute Problem beseitigen, sondern auch, wie sie die Grundursache (Root Cause) identifizieren und beseitigen können. Und damit wird es sehr unwahrscheinlich, dass die gleiche Art von Problem erneut auftaucht. Auch wenn andere Probleme auftreten, steigt die Wahrscheinlichkeit, dass die Mitarbeiter die Probleme und Problemursachen selbst beseitigen können (siehe [Rother 2013]).

4.7 Fallbeispiele zu moderner Mitarbeiterführung

In diesem Abschnitt zeigen wir verschiedene Beispiele aus der Praxis, wie Mitarbeiterführung in Unternehmen organisiert wird, die dauerhaft auf agile Entwicklung setzen. Die meisten Beispiele stammen aus den Entwicklungsbereichen der jeweiligen Unternehmen. In anderen Unternehmensbereichen existieren mitunter abweichende Strukturen. Außerdem ist die Realität nicht ganz so stringent, wie die Beispiele vielleicht auf den ersten Blick vermuten lassen. Es gibt immer auch Ausnahmen, wo es nicht genauso ist wie beschrieben. Im Sinne der leichteren Verständlichkeit haben wir uns entschieden, nicht jede Ausnahme aufzuzeigen.

Die präsentierten Informationen basieren auf Interviews mit Mitarbeitern der vorgestellten Unternehmen ImmobilienScout24, siggate und it-agile sowie unseren Beobachtungen vor Ort (siehe auch [Roock 2016b]).

4.7.1 ImmobilienScout24

Das Unternehmen

ImmobilienScout24 ist das führende Immobilienportal im deutschsprachigen Internet und die Nr. 1 rund um Immobilien. Das Portal hat jeden Monat ca. 12 Mio. Besucher. ImmobilienScout24 beschäftigt über 500 Mitarbeiter.

Scrum-Einführung 2008

Im Jahre 2008 führte ImmobilienScout24 Scrum für die Softwareentwicklung ein. Es wurden cross-funktionale Teams mit 5 – 9 Teammitgliedern plus Product Owner und Scrum Master gebildet. Die vorher existierende Struktur der disziplinarischen Führung über Teamleiter wurde beibehalten. Die Teamleiterstruktur orientierte sich an den Spezialisierungen wie Design, Frontend-Entwicklung, Backend-Entwicklung oder Qualitätssicherung. Abbildung 4-26 zeigt die resultierende Struktur.

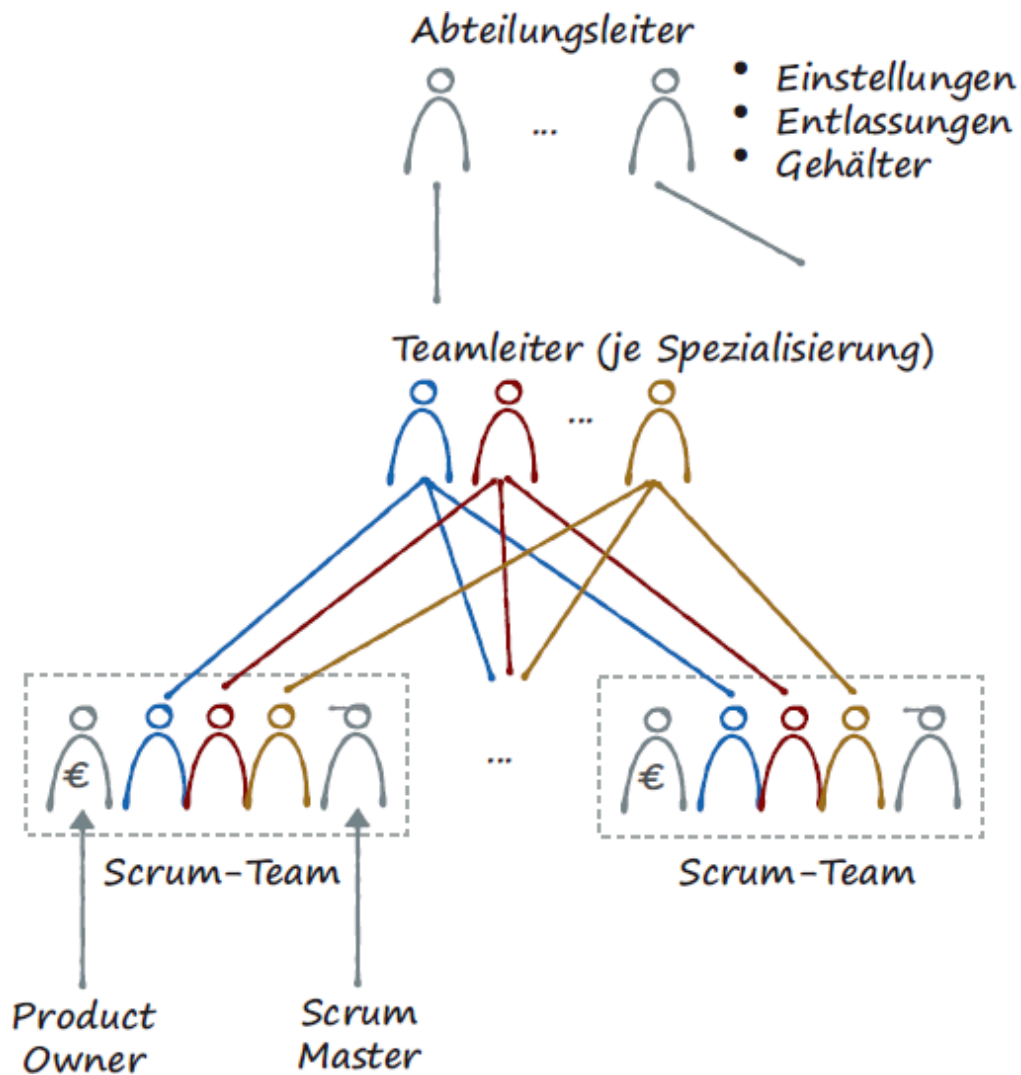


Abb. 4-26 Teamleiter bei ImmobilienScout24

Dabei hatten die Teamleiter sehr unterschiedliche Führungsspannen. Einige betreuten eine Handvoll Mitarbeiter und arbeiteten auch in Entwicklungsteams mit. Andere betreuten 25 Mitarbeiter und hatten nur sehr sporadisch Kontakt mit ihnen.

Abgrenzung Scrum Master und Teamleiter

Insbesondere zwischen Scrum Master und Teamleiter gab es bzgl. der Rollenbeschreibungen große Überschneidungen in den Verantwortlichkeiten. Scrum Master und Teamleiter handelten selbst aus, wie sie die Aufgaben konkret aufteilen. So kümmerten sich in einigen Fällen die Teamleiter um die Mitarbeiterentwicklung und in anderen die Scrum Master.

Abteilungsleiter

Einstellungen, Entlassungen und Gehaltsfestlegungen fanden auf der Abteilungsleiterenebene oberhalb der Teamleiter statt. Teamleiter wurden dabei

mitunter konsultiert.

Probleme

Diese Arbeitsweise war tragfähig und funktionierte jahrelang ausreichend gut. Es gab allerdings auch Probleme damit, wie Frank Schlesinger von ImmobilienScout24 berichtet: »Die Rollenbeschreibungen von Teamleiter und Scrum Master waren nicht trennscharf. Dadurch war viel Energie notwendig, um die jeweiligen Tanzbereiche abzustecken. Manchmal waren Scrum Master im Wesentlichen damit beschäftigt, das Team vor einem Teamleiter zu schützen. Das schien unnötige Verschwendung zu sein und das System verschleierte Probleme, die wir in der Organisation auch mit der Weiterentwicklung von Teamleitern hatten.«

Reorganisation 2015

ImmobilienScout24 reorganisierte sich 2014/2015. Dabei wurde auch die Führungsstruktur überdacht. Man verfolgte jetzt den Ansatz, dass nur diejenigen Führung übernehmen dürfen, die »Agil« verinnerlicht hatten. Für diejenigen, die das nach Schulungen und auch nach Jahren im agilen Unternehmen ImmobilienScout24 sich nicht zu Eigen gemacht hatten, würde im Unternehmen kein Platz mehr in einer Führungsposition sein.

Neue Rolle Delivery Lead ersetzt Scrum Master und Teamleiter

In der neuen Struktur gibt es keine Unterscheidung mehr zwischen Teamleitern und Scrum Mastern. Um den Wechsel im Rollenverständnis explizit zu machen, wurden beide Rollen abgeschafft und durch die Delivery-Lead-Rolle ersetzt (siehe Abb. 4-27). Delivery Leads sind jeweils verantwortlich für ein Team aus 7 – 9 Mitarbeitern und sitzen im Teamraum (so wie vorher die Scrum Master).

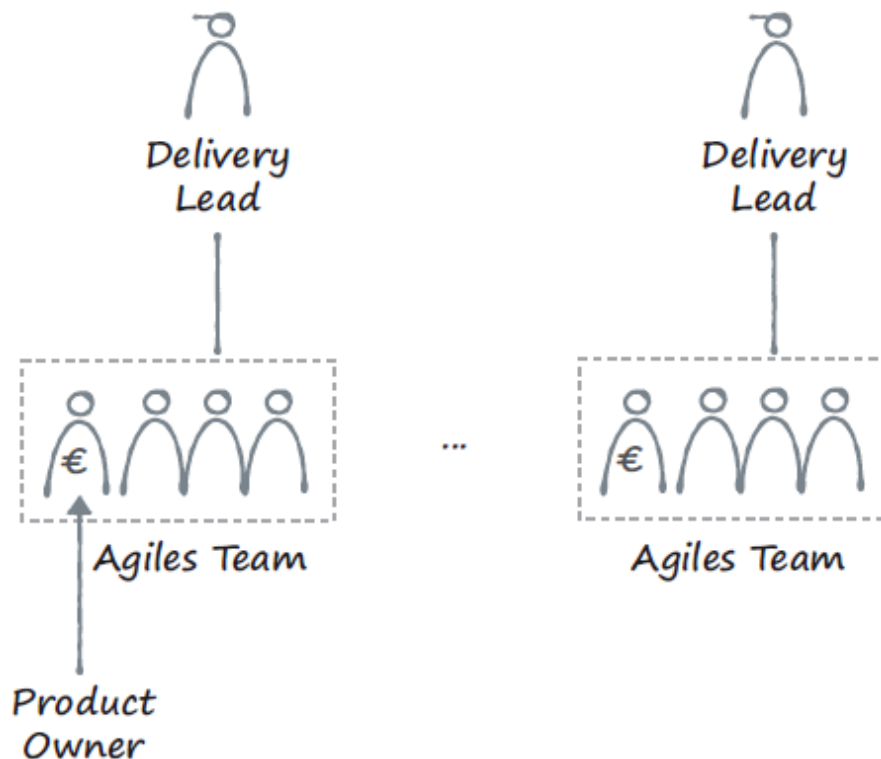


Abb. 4-27 Delivery Leads bei ImmobilienScout24

Die Delivery Leads sind im Kern dafür verantwortlich, dass ihre Teams effektiv Software liefern können. Arbeitszufriedenheit der Mitarbeiter, Alignment und Autonomie sind dafür nach Auffassung von ImmobilienScout24 ebenfalls notwendig und fallen daher auch in den Verantwortungsbereich der Delivery Leads.

Die Delivery Leads rekrutieren sich aus ehemaligen Teamleitern, Scrum Mastern und Senior Softwareentwicklern. Da konsequenter als bisher auf handhabbare Führungsspannen geachtet wurde, gibt es jetzt mehr Delivery Leads, als es zuvor Teamleiter gegeben hat. Ein Teil der Agile Coaches hat das Unternehmen mittlerweile verlassen, weil sie mit der Veränderung nicht einverstanden waren oder nicht als Delivery Lead arbeiten wollten.

Im Zielmodell sind die Delivery Leads jeweils für komplette Teams verantwortlich. Sie wären also bzgl. der Mitarbeiterentwicklung über mehrere Disziplinen (Design, Frontend, Backend, Qualitätssicherung etc.) verantwortlich. Das ist heute noch nicht komplett verwirklicht. So haben Product Owner und Designer heute noch andere Vorgesetzte. Sollte das für die Lieferung der Software problematisch werden, kann ein Delivery Lead aber den Designer in seinem Team durch einen selbst eingestellten Designer ersetzen.

Heads

Über den Delivery-Managern finden sich in der Hierarchie Heads (Abteilungsleiter). Es ist nicht im Detail festgelegt, was die jeweilige Hierarchiestufe zu tun hat, sondern »nur« wofür sie verantwortlich ist. Entscheidend ist der Grundgedanke von Empowerment, den Delivery Leads mehr Verantwortung zu übertragen, beispielsweise für ihre Mitarbeiter und deren Gehälter. Sie können dann entscheiden, ob sie Gehälter klassisch festlegen, mit Gehaltstransparenz arbeiten etc. Entsprechend sind auch Einstellungen und Entlassungen geregelt.

Die Aufgabe der Heads ist es, strategische Entscheidungen zu treffen und für die geeigneten Rahmenbedingungen zu sorgen, damit die Teams ihre Ziele bestmöglich erreichen können.

Es gibt regelmäßige Abstimmungen zwischen den Heads (6 Personen), in denen sie sich darüber austauschen, wie die Dinge in ihren jeweiligen Bereichen geregelt werden. Dadurch erhofft man sich gegenseitiges Lernen und ausreichend einheitliche Vorgehensweisen.

Herausforderungen der Delivery-Lead-Rolle

Eine Herausforderung für die Delivery Leads liegt offensichtlich in der Kombination aus Coaching und Gehaltsfragen/Entlassungen. Für Coaching benötigt der Delivery Lead das Vertrauen seiner Mitarbeiter. Dass er über ihre Gehälter oder im Extremfall auch über ihre Entlassung entscheidet, kann es erschweren, das notwendige Vertrauen aufzubauen. Wenn aus Sicht der Mitarbeiter das notwendige Vertrauensverhältnis nicht existiert, haben sie die Möglichkeit, dies an anderer Stelle (Personalabteilung, Betriebsrat) zu melden. Außerdem gibt es vierteljährlich ein Feedback der Mitarbeiter zu ihren Führungskräften, aus dem Vertrauensprobleme ablesbar sein sollten.

Außerdem ist man sich bei ImmobilienScout24 klar darüber, dass dieser Konflikt innerhalb der Delivery-Lead-Rolle hohe Anforderungen an das Selbstmanagement der Delivery Leads stellt. Eine entsprechende Persönlichkeitsentwicklung wird über persönliche Coaches und Schulungen unterstützt.

Nach Einschätzung von Frank Schlesinger funktioniert dieses Modell jetzt, weil vorher sieben Jahre lang agile Führungsansätze gelebt und laterale Führung eingeübt wurde. »Wenn wir die heutige Struktur von Anfang an eingeführt hätten, hätte das nie und nimmer funktioniert, weil uns die entsprechende Führungskultur gefehlt hätte«, sagte Frank Schlesinger dazu.

4.7.2 sipgate

Das Unternehmen

Die sipgate GmbH wurde 2004 gegründet und ist mit mehreren Hunderttausend Kunden in Deutschland und Großbritannien einer der größten VoIP-Anbieter Europas. Sie bietet Cloud-Telefonielösungen sowie Mobilfunk für Privatkunden und Firmen an. Bei sipgate arbeiten 128 Mitarbeiter.

Teams

Das Unternehmen sipgate arbeitet mit langfristig stabilen cross-funktionalen Scrum-Teams. Die Teammitglieder (ausgenommen Product Owner und Scrum Master) sind alle Vollzeit im Team. Die Teammitglieder entscheiden selbst darüber, das Team zu wechseln.

Manchmal sind spezielle Aufgaben zu bearbeiten, die nicht gut in einem existierenden Team gelöst werden können. Dann bilden die Teams selbst temporäre Teams (Temp-Teams), die als Taskforce konzentriert die anstehende Aufgabe erfüllen. Die Mitarbeiter sind für die Dauer des Temp-Teams Vollzeitmitglieder im Temp-Team.

Product Owner und Scrum Master

Die meisten Teams haben keine exklusiven Product Owner und Scrum Master, sondern teilen sich die Personen mit anderen Teams. Einige reifere Teams haben gar keine fest zugeordneten Scrum Master.

Keine Linienvorgesetzten

Bei sipgate liegt sehr viel Verantwortung in den Händen der cross-funktionalen Produktteams. Die Teams stellen ein, entscheiden über Entlassungen und regeln die persönliche Weiterentwicklung der Teammitglieder. Das Personalteam steht dabei unterstützend zur Seite.

Lediglich die Gehaltsfestlegung findet (noch) direkt bei der Geschäftsführung statt. Es gibt aber Pläne, das Gehalt über eine transparente Gehaltsformel festzulegen.

Einstellung neuer Mitarbeiter

Sieht ein Team Bedarf an einem zusätzlichen Teammitglied, startet es einen Einstellungsprozess. Es gibt hier interessanterweise keine Einschränkungen bzgl. Budget oder Anzahl Teammitglieder – sipgate möchte gerne deutlich mehr Leute einstellen, die Teams sind da eher die Bremser.

In Kooperation mit dem Personalteam schreibt das Team die Stelle aus, sichtet Bewerbungen, führt das Einstellungsgespräch durch und entscheidet über die Einstellung. Bei der Einstellungsentscheidung gibt es keine formale Regelung über die notwendige Mehrheit. Es gibt aber ein geteiltes Verständnis darüber, dass eine einfache Mehrheit nicht ausreicht, sondern eher eine Dreiviertelmehrheit. Angestrebt wird Konsens.

Neueingestellte Mitarbeiter bekommen einen Paten an die Seite gestellt, der die Einarbeitung begleitet und sicherstellt, dass der Mitarbeiter nach einem und nach vier Monaten Feedback bekommt. Wenn sich dabei herausstellt, dass es Probleme gibt, versucht man diese noch in der Probezeit zu beheben (Feedback an den Mitarbeiter; ggf. anderes Team, andere Rolle etc.). Gelingt das nicht, entscheidet das Team über die Entlassung während der Probezeit.

Gehälter

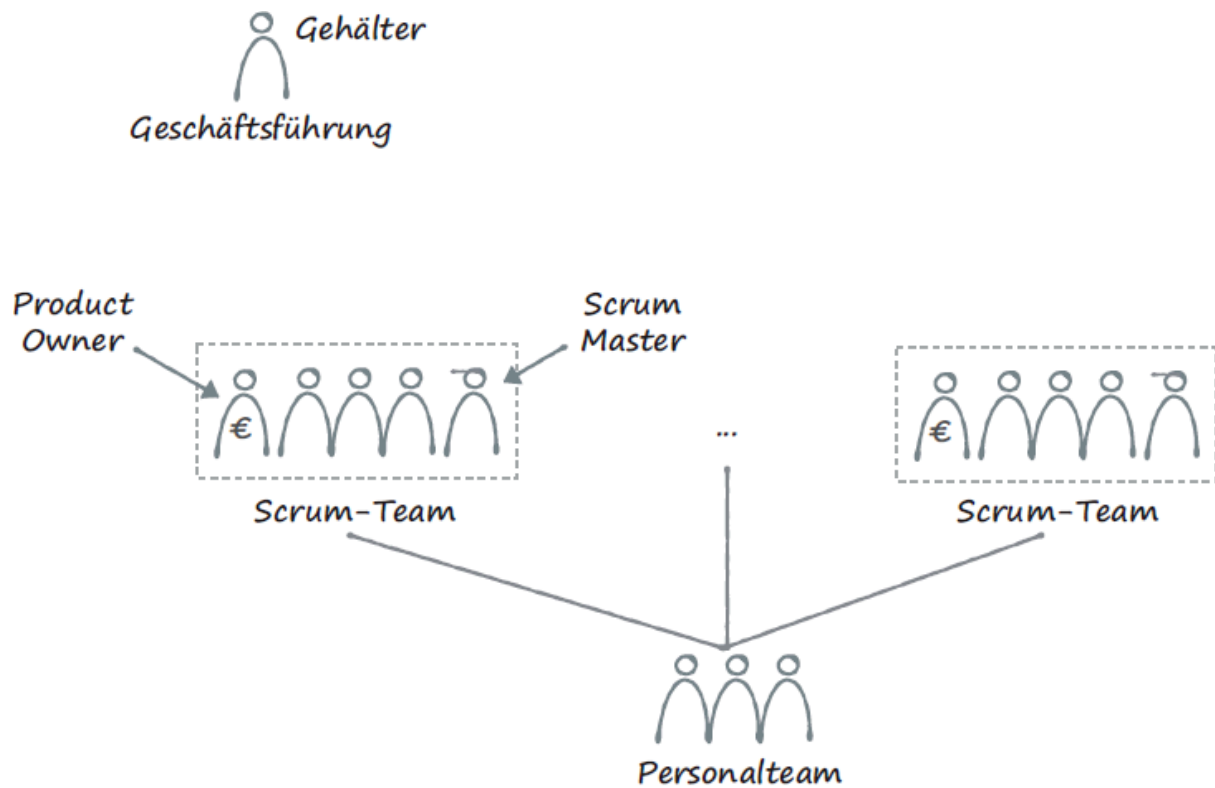
Bei Neueinstellungen fragt sipgate vor dem Einstellungsgespräch nach der Gehaltsvorstellung. Wenn diese nicht ins allgemeine Gehaltsgefüge passt, bekommt der Kandidat entsprechendes Feedback und kann seine Gehaltsvorstellung anpassen. Nach der Einstellungsentscheidung unterhält sich noch mal die Geschäftsführung mit dem Kandidaten über das Gehalt, insbesondere wenn die Gehaltsvorstellung ungewöhnlich ist – sonst folgt das Personalteam meist dem Wunsch des Kandidaten.

Gehaltserhöhungen liegen zurzeit ebenfalls noch bei der Geschäftsführung. Sie entscheidet auf Basis der Gehaltsrelation zu anderen Mitarbeitern.

Das Ziel ist aber, perspektivisch zu einer Gehaltsformel zu kommen, damit jeder Mitarbeiter weiß, welche Aspekte wie stark ins Gehalt einfließen.

Entlassungen

Das Team entscheidet über Entlassungen innerhalb und außerhalb der Probezeit. Normalerweise spürt jeder im Team, dass ein Kollege nicht ins Team oder zu sipgate passt. Allerdings wissen die Teammitglieder zunächst nicht, dass ihre Teamkollegen eine ähnliche Einschätzung haben. Oft durchbrechen die Scrum Master diese Hürde und thematisieren die Schwierigkeiten. Dann wird mehrmals versucht, das Problem, das zur Einschätzung der Teammitglieder führt, zu beseitigen. Wenn das nicht funktioniert, fällt das Team die Entlassungsentscheidung und muss diese Entscheidung dann auch persönlich mitteilen. Die Verantwortung liegt auch hier bei den Teammitgliedern als Entscheidern.



Die Produktteams werden vom Personalteam unterstützt bei:

- Einstellungen und Entlassungen
- Fürsorge

Abb. 4-28 Führung bei sipgate ohne Linienvorgesetzte

Persönliche Weiterentwicklung

Die Mitarbeiter sind selbst für ihre Weiterentwicklung verantwortlich. Es existiert die Erwartungshaltung, mindestens zwei Weiterbildungen pro Jahr in Anspruch zu nehmen. Dazu gibt es ein physisches Board, das pro Mitarbeiter zwei Slots vorsieht, in die der Mitarbeiter seine geplanten Weiterbildungen einträgt. Es gibt kein festes Budget für Weiterbildung. Jeder entscheidet selbst, welche Ausgaben er für angemessen hält. Durch Weiterbildung bedingte Abwesenheiten werden im Team abgestimmt.

Fürsorge

Bei sipgate stempeln die Mitarbeiter ihre Arbeitszeiten. Dadurch soll sichergestellt werden, dass die Wochenarbeitszeit von 40 Stunden nicht überschritten wird. Dass es hier nicht um Kontrolle geht, kann man daran erkennen, dass jeder seine Zeiten nachträglich über ein Web-Interface ändern kann. Wenn jemand zu viel arbeitet, sucht das Personalteam das Gespräch mit ihm.

Darüber hinaus achten die Scrum Master und die anderen Teammitglieder aufeinander (z. B. wie es Kollegen persönlich geht). Und nicht zuletzt steht das

Personalteam als Gesprächspartner bei Problemen zur Verfügung.

Der lange Weg zur Freiheit

Die heute bei sipgate existierende Struktur hat sich über Jahre in kleinen Schritten entwickelt. Corinna Baldauf sagt dazu: »Freiheit muss man lernen. Wenn wir das alles von Anfang an gemacht hätten, wäre das nach hinten losgegangen.«

4.7.3 it-agile

Das Unternehmen

it-agile ist ein Beratungsunternehmen, das zu agiler Entwicklung berät und coacht. Zurzeit beschäftigt it-agile gut 30 Mitarbeiter. it-agile wendet die agilen Prinzipien auch für die interne Organisation an. Selbstorganisierte Teams stehen also im Zentrum der Firmenstruktur.

Teams

Die Berater sind in Business-Teams organisiert, die die komplette Wertschöpfung von der Kontaktgenerierung bis hin zur Abrechnung verantworten. Sie nutzen dabei die Unterstützung von zwei Zentralteams: Marketing und Moneypenny (für Abrechnung, Büroorganisation etc.). Marketing ist ein virtuelles Team – die Teammitglieder sind auch Mitglieder in Business-Teams. Im Moneypenny-Team sind die Teammitglieder exklusiv nur im Moneypenny-Team tätig.

Es gibt also keine Mitarbeiter außerhalb von Teams (auch die beiden Geschäftsführer sind Mitglieder in Business-Teams). Innerhalb der Teams gibt es keine weiteren Rollen wie Product Owner oder Scrum Master.

Fürsorge

Die Teams sind auch für die Fürsorge gegenüber ihren Teammitgliedern verantwortlich. Sie haben im Auge, ob Mitarbeiter sich überlasten, häufig krankheitsbedingt ausfallen oder unzufrieden mit ihrer Arbeit sind.

Rollen statt Positionen

Generell versucht it-agile, feste Positionen zu vermeiden und stattdessen mit temporären Rollen zu arbeiten. So sind Geschäftsführung, Datenschutz, Gehaltsfestlegungen etc. über temporäre Rollen organisiert.

Geschäftsführung

it-agile hat eine formelle Definition der Geschäftsführer-Rolle: Die Geschäftsführer sind im Wesentlichen dazu da, um formalrechtliche Unterschriften zu leisten (Arbeitsverträge, Kündigungen, Kontovollmachten), nachdem andere die entsprechenden Entscheidungen getroffen haben. Außerdem haben sie die Aufgabe, nach Bereichen Ausschau zu halten, die durch das existierende System der Selbstorganisation nicht ausreichend abgedeckt sind, und diese Defizite explizit zu machen. Die Reaktion auf diese Defizite obliegt allerdings den Mitarbeitern und nicht der Geschäftsführung.

Dass Geschäftsführung als temporäre Rolle begriffen werden kann, hängt mit den Beteiligungsverhältnissen bei it-agile zusammen. Der Großteil des Unternehmens gehört einer Mitarbeiterbeteiligungsgesellschaft, an der alle Mitarbeiter beteiligt sind. Somit wählen faktisch die Mitarbeiter die bei einer GmbH notwendige Geschäftsführung.

Einstellungen

Der Einstellungsprozess beginnt mit einem informellen Vorgespräch zwischen dem Bewerber und einem Mitarbeiter. Findet sich kein Mitarbeiter für dieses Vorgespräch, wird dem Bewerber abgesagt. Fand das Vorgespräch statt, schildert der Mitarbeiter seine Eindrücke allen Mitarbeitern und sucht nach Interessenten für ein formelles Bewerbungsgespräch. Sind mindestens vier Mitarbeiter dazu bereit, findet das Bewerbungsgespräch mit diesen Mitarbeitern als Einstellungsgruppe statt. Ansonsten wird dem Bewerber abgesagt. Das Bewerbungsgespräch dauert ½ bis 1 Tag. Am Ende des Gespräches entscheidet die Einstellungsgruppe, ob und auf welcher Gehaltsstufe sie dem Bewerber ein Angebot macht. Außerdem wird direkt festgelegt, in welches Team der Bewerber am Anfang kommt und wer sein Einarbeitungsmentor wird. Der Einarbeitungsmentor kümmert sich darum, dass der neue Mitarbeiter gut bei it-agile ankommt. Am Ende der Probezeit entscheidet der Einarbeitungsmentor, ob der neue Kollege übernommen wird – seine Entscheidung basiert auf Konsultationen mit Kollegen.

Gehaltserhöhungen

Für Gehaltserhöhungen sind die Gehalts-Checker verantwortlich. Die Gehalts-Checker sind eine Gruppe aus vier Mitarbeitern, die jährlich von allen Mitarbeitern gewählt werden. Sie legen den Prozess zur Gehaltserhöhung fest und fällen die endgültigen Gehaltsentscheidungen.

Momentan basiert der Prozess zu Gehaltserhöhungen auf dem Konsultationsprinzip. Wer hochgestuft werden möchte, konsultiert Kollegen zu

dem Hochstufungswunsch. Auf Basis der Konsultationen entscheidet der Mitarbeiter, ob er wirklich hochgestuft werden möchte. Wenn das der Fall ist, meldet er diesen Hochstufungswunsch bei den Gehalts-Checkern an und schildert die durchgeführten Konsultationen. Auf dieser Basis entscheiden die Gehalts-Checker, ob der Mitarbeiter hochgestuft wird.

Weiterentwicklung der Mitarbeiter

Jeder Mitarbeiter kümmert sich um einen für ihn passenden Mechanismus zur persönlichen Weiterentwicklung. Einzige Bedingung ist, dass dieser Mechanismus Feedback von Kollegen integrieren muss. Die meisten Kollegen arbeiten mit Peer-groups. Sie suchen sich Peergroups von drei oder vier Kollegen, die ihnen bei der persönlichen Weiterentwicklung helfen. In der letzten Zeit haben sich vermehrt Rudel-Peergroups gebildet: Hier finden sich vier bis sechs Mitarbeiter zu einer Peergroup zusammen. Damit werden die Transaktionskosten zur Terminkoordination gesenkt.

Jeder Mitarbeiter investiert für seine Weiterbildung das, was er für sinnvoll hält, und rechnet die Kosten als Spesen ein. Da die Spesen generell öffentlich für alle Kollegen sind, findet darüber eine soziale Kontrolle statt. Im Arbeitsvertrag sichert it-agile zehn Weiterbildungstage pro Jahr zu. In Absprache mit dem eigenen Team können Mitarbeiter auch mehr Tage in ihre Weiterbildung investieren.

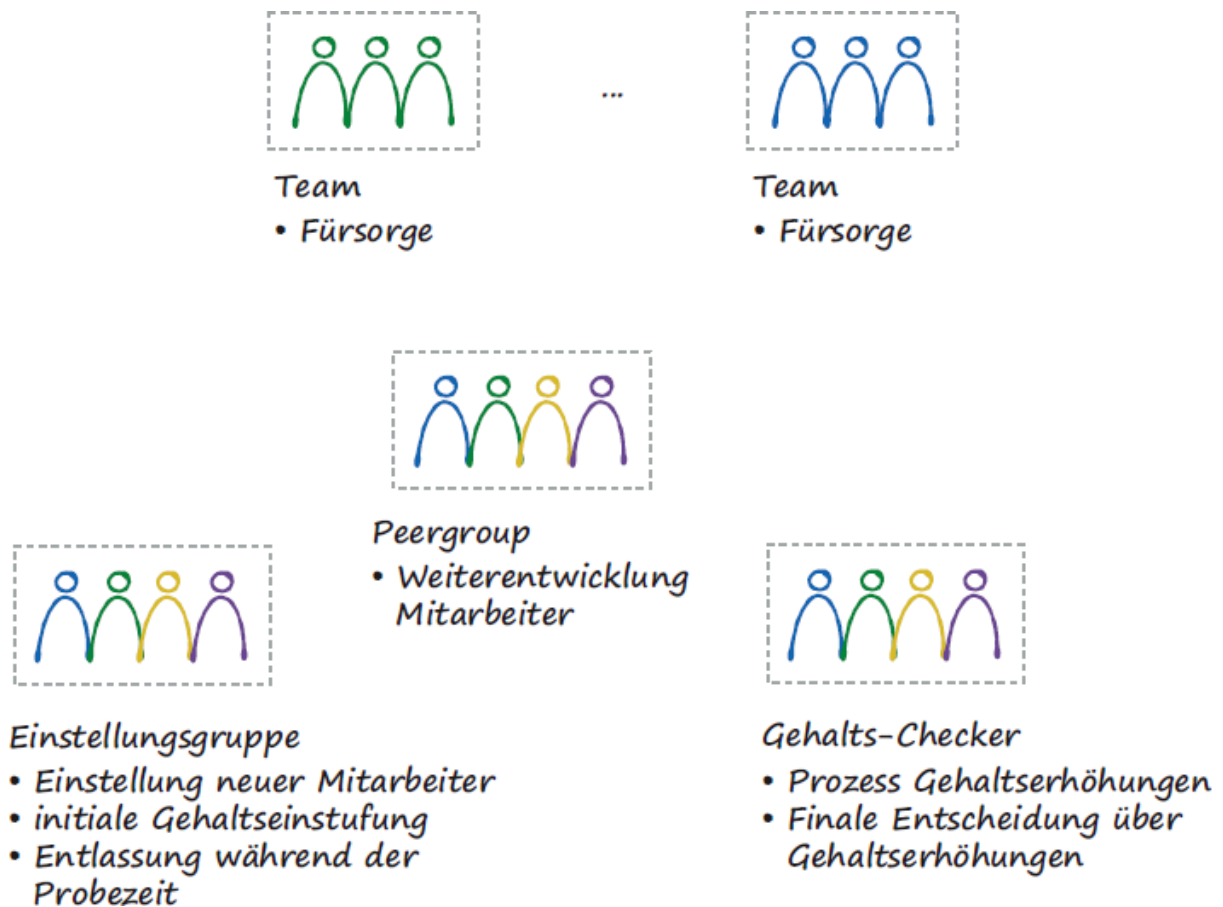


Abb. 4-29 Mitarbeiterführung bei it-agile

Kündigungen

Kündigungen sieht it-agile als Spezialfall von Konflikten an. Es gab ja mal gute Gründe, den Mitarbeiter einzustellen. Zuerst wird daher versucht, wieder ein gutes Miteinander zu ermöglichen. Wenn ein Mitarbeiter der Meinung ist, dass ein Kollege entlassen werden sollte, muss er zuerst das Gespräch mit dem Kollegen suchen. Dort sollte darüber gesprochen werden, worin das Problem liegt und welche Verhaltensänderung das Problem ggf. beseitigen kann. Gelingt den beiden Mitarbeitern keine Klärung, ziehen sie einen internen oder externen Mediator hinzu. Kommt es auch mithilfe des Mediators zu keiner Klärung, benennt der Mediator einen Entscheider, der darüber entscheidet, ob der Mitarbeiter entlassen wird.

4.7.4 Zusammenfassung der Fallbeispiele für Mitarbeiterführung

Der Abschnitt hat sehr unterschiedliche Modelle in sehr unterschiedlichen Unternehmen vorgestellt. Es gibt eine Gemeinsamkeit zwischen den vorgestellten Modellen und viele Unterschiede. Die große Gemeinsamkeit ist, dass inhaltliche

Führung und Weiterentwicklung von Team und Teammitgliedern voneinander getrennt sind – so wie es auch in der Scrum-Rollentrennung zwischen Product Owner und Scrum Master angelegt ist. Die Unterschiede sind:

- Die meisten Unternehmen arbeiten mit stabilen Teams, sodass sich die Personalfrage nicht stellt.
- Die meisten Unternehmen haben neben dem Scrum Master/Team-Coach einen disziplinarischen Vorgesetzten. In den von uns vorgestellten Unternehmen sind in einem Fall diese beiden Rollen in einer Rolle verschmolzen (Delivery Lead bei ImmobilienScout24). In zwei Fällen (sipgate, it-agile) gibt es faktisch keinen disziplinarischen Vorgesetzten.

Es gibt offensichtlich nicht das eine richtige oder perfekte Modell für Mitarbeiterführung. Jedes Modell hat seine Stärken und Schwächen. Jedes Unternehmen muss das Modell finden, das für den eigenen Kontext passt und mit der gewünschten Kultur harmoniert. Und man darf nicht erwarten, dass sich das einmal gefundene Modell nicht mehr ändert. ImmobilienScout24 und sipgate arbeiten heute mit Modellen, die nach ihrer eigenen Einschätzung vor fünf Jahren bei ihnen nicht funktioniert hätten. Das it-agile-Modell hat sich über 12 Jahre hinweg schrittweise entwickelt. Das verwendete Führungsmodell muss sich mit dem Unternehmen weiterentwickeln.

4.8 Unternehmenskultur

Wir haben Fälle erlebt, in denen sich Unternehmen agil(er) strukturiert haben und sich faktisch nichts geändert hat. Und wir haben Unternehmen kennengelernt, die sich deutlich agil verhalten haben, obwohl ihre Strukturen eher klassisch anmuteten. Was wir bisher ausgespart haben, ist die Unternehmenskultur.

Peter Drucker sagt: »Culture eats strategy for breakfast«, und verdeutlicht damit die überragende Bedeutung von Unternehmenskultur.

Unternehmenskultur lässt sich aber leider nicht direkt ändern. Niels Pfläging bezeichnet Unternehmenskultur als Schatten der Strukturen, Prozesse und Verhaltensweisen in Organisationen. Man kann sie wahrnehmen, aber nur indirekt ändern – indem man die Strukturen, Prozesse und Verhaltensweisen beeinflusst. Das PARC-Modell von John Roberts bringt es auf folgende Formel (siehe [Roberts 2007]):

$$\begin{aligned} & \text{People} \\ & + \text{Architecture (formell und informell)} \\ & + \text{Rituals (formell und informell)} \\ \hline & = \text{Culture} \end{aligned}$$

Abb. 4-30 Das PARC-Modell

Offensichtlich spielen die *Menschen* (People) eine gewichtige Rolle für die Unternehmenskultur. Sie bringen ihre Werte, Hoffnungen und Ängste mit ins Unternehmen und prägen damit ihr Verhalten und die Interaktionen mit ihren Kollegen.

Die *formelle Architektur* bezeichnet in dem PARC-Modell sowohl die Gebäudearchitektur (Räume, Stockwerke) wie auch die Aufbauorganisation des Unternehmens. Die *informelle Architektur* bildet sich z. B. über Freundeskreise im Unternehmen.

Formelle Rituale sind z. B. offizielle Besprechungen, Planungssitzungen oder Daily Scrums. *Informelle Rituale* sind z. B. die Gespräche in der Teeküche oder auch der berüchtigte Flurfunk.

Möchte man also die Unternehmenskultur ändern (z. B. stärker in Richtung Vertrauen und Zusammenarbeit ausprägen), muss man Menschen, Architektur und Rituale betrachten. Dabei reicht es häufig nicht aus, an nur einer Stellschraube zu drehen, um eine nachhaltige Veränderung zu bewirken. Das Gesamtsystem Unternehmen befindet sich in einem selbststabilisierenden Zustand. Wird ein Element geändert, kompensieren das die anderen Elemente. Entfernt man beispielsweise ein unnützes Meeting, in dem lediglich Gerüchte verbreitet wurden, wird die Gerüchteküche vermutlich schlicht in die Teeküche verlegt.

4.8.1 Unternehmenskultur und agiles Arbeiten

Warum ist die Frage nach der Unternehmenskultur überhaupt von Relevanz? Wenn wir eine beliebige Firma betrachten und alle Mitarbeiter in Methoden agiler Zusammenarbeit ausbilden ist das eine notwendige, aber nicht hinreichende Bedingung. Die Erfolgchancen sind so zu niedrig.

Die Art, wie z. B. das Scrum Framework heute verstanden wird, basiert auf agilen Werten und Prinzipien – sowie den Scrum-Werten.

Die agilen Werte und Prinzipien sind im Agilen Manifest [Beck et al. 2001] formuliert. Die älteste Referenz auf die Scrum-Werte findet sich in dem ersten Buch zu Scrum [Schwaber & Beedle 2002] im 9. Kapitel. Die Autoren betonen dort, dass die Scrum-Werte keine Voraussetzung für die Anwendung von Scrum sind. Sie haben aber beobachtet, wie die Scrum-Werte ganz automatisch wachsen, wenn Scrum ernsthaft als Werkzeug benutzt wird.

Die agilen Werte und Prinzipien stehen also im Zentrum. Das Scrum Framework ist eine mögliche Umsetzung, die situationsabhängig um weitere agile Praktiken ergänzt wird (siehe Abb. 4-31).

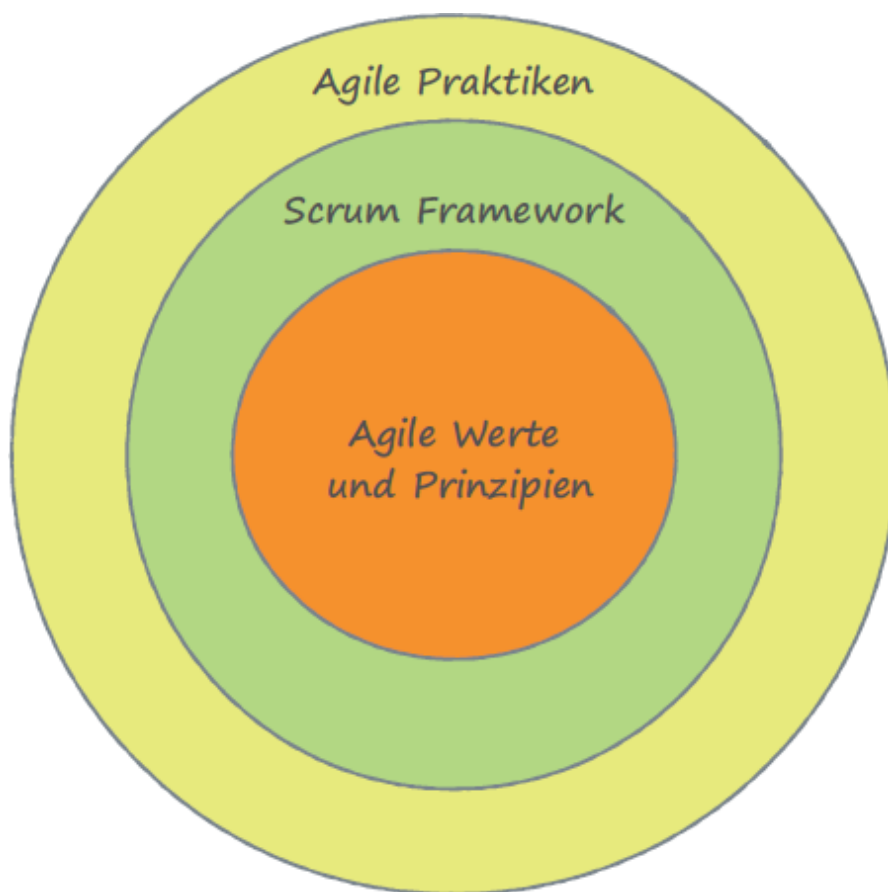


Abb. 4-31 Im Kern stehen die agilen Werte und Prinzipien.

Wenn die in der Firma tatsächlich gelebten Werte und Prinzipien nicht zu agilen Werten und Prinzipien passen, dann wird die Organisation allergisch auf die Anwendung von z. B. Scrum reagieren. Das ist nachvollziehbar, weil die neuen Werte inkompatibel mit den traditionellen Firmenwerten sind.

Fallbeispiel: Zu große Transparenz (von Jürgen Hoffmann)

Vor einigen Jahren durfte ich bei einem großen Konzern eines der ersten Scrum-Teams begleiten. In den ersten Sprints wurde für alle in der

Organisation offensichtlich, dass die Softwareentwickler in diesem Team nicht in der Lage waren, zu einem Ergebnis zu kommen. Die Reviews waren für alle frustrierend und in den Retrospektiven offenbarten sich die nicht vorhandenen Spielräume des Teams für Veränderungen. Diese Transparenz wiederum war nicht tolerierbar, denn in der Firmenlogik bedeutete das Versagen der Mitarbeiter ein Versagen der zuständigen Führungskräfte. Nach einiger Zeit wurde sowohl der Product Owner als auch der Scrum Master aus seiner Verantwortung entlassen und das Projekt unter einem anderen Namen in traditioneller Weise neu aufgesetzt. Mit dem etablierten Projektmanagement war es kein Problem, die nicht vorhandene Lieferfähigkeit über lange Zeiten zu ignorieren, weil die Kollegen ja ständig beschäftigt waren und die nächsten Meilensteine in weiter Ferne lagen.

4.9 Das Kapitel in Stichworten

- Agil bedeutet im Kern, dass autonome Teams in kurzen Zyklen Probleme von Endkunden lösen.
- Viele Unternehmen haben einen ganzen Zoo von Störungen dieses einfachen Zyklus wertschöpfender Arbeit installiert.
- Unternehmen müssen sich so entwickeln, dass sie diese Störungen beseitigen und das ganze Unternehmen der Wertschöpfung unterordnen.
- Das Zellmodell von Niels Pfläging treibt diese Denkweise auf die Spitze: Niemand macht den Teams, die Kundenprobleme lösen, Vorschriften.
- Solch extrem dezentrale Strukturen laufen Gefahr, ins Chaos abzurutschen. Das notwendige Alignment wird hergestellt, wenn allen Beteiligten das Was und Warum klar ist.
- Peter Drucker hat dazu ein Managementsystem namens Management by Objectives (MbO) oder auch »Führen durch Ziele« definiert. Dieses erfährt in der Variante Objectives and Key Results (OKR) zurzeit eine Renaissance.
- MbO/OKR können Alignment herstellen, sind aber auch nicht ohne Probleme. Diese hängen unter anderem damit zusammen, dass das installierte Zielsystem statisch ist und damit die notwendige Dynamik behindert wird. MbO folgt tendenziell dem »Unternehmen als Maschine«-Modell.
- Feedbackschleifen sind eine alternative Möglichkeit zur Unternehmenssteuerung, die dem »Unternehmen als Organismus«-Modell folgt. Sie basieren auf kontinuierlicher Überprüfung des Status quo und Anpassung der nächsten Schritte.

- Bei aller Dezentralisierung müssen Unternehmen auch übergreifende Entscheidungen treffen. Konsent und der Advice-Prozess sind zwei Möglichkeiten, wie das geschehen kann, ohne dass Unternehmen zurück in Top-down-Führung per Command & Control verfallen.
- Neben der Unternehmensstruktur muss sich auch Führungsstruktur und Führungsverhalten ändern. In agilen Unternehmen werden Führungsaufgaben auf mehr Köpfe verteilt und tendenziell von temporären »Führungskräften« wahrgenommen.
- Dadurch wird die gesamte Unternehmensstruktur anpassungsfähiger und kann besser auf den Markt reagieren.
- Zusätzlich muss sich das Selbstverständnis der permanenten oder temporären Führungskräfte ändern. Sie müssen zu Dienstleistern der wertschöpfenden Teams werden. Aus Reporting Lines werden Supporting Lines: Die Hierarchie wird auf den Kopf gestellt.
- Neben den Unternehmensstrukturen ist die Unternehmenskultur relevant. Das PARC-Modell erklärt, welche Einflussmöglichkeiten existieren.

Index

A

Adaptive Planung 7

Advice-Prozess 93, 95, 98

Agile Fluency 5

Agile Fluency Model™ 5, 9

 vier Zonen 6

Agile Kernidee 2

 Zyklus 2, 64

Agile Skalierung 72

Agiles Manifest 86–87, 119

Agilität, echte 2, 7

Alignment 75, 148

Ansoff-Matrix 40

Anstrengung

 außerordentliche 85

 heroische 86

Ausbildung 105

Ausprobieren, Erfolgskontrolle und Anpassung 1

Autonomie 75

A3-Technik 154

B

Bedürfnis 16

Beobachten 34

Build-Measure-Learn-Zyklus 175

Business-fokussiertes Team 5

Business-Modell 36

Business-Plan 47

C

Causal-Loop-Diagramm 91–92

Change-Prozess 123

CIF *siehe Continuous Improvement Framework*

Command Control 75, 100

Command & Control-Struktur 90

Compliance 90

Continuous Improvement Framework (CIF) 1

Culture eats strategy for breakfast 118

D

Design Sprints 30, 161, 166

- Ablauf 168

- Raum 168

- Team 167

Design Thinking 161, 165

- Raum 164

 - Checkliste 164

- Team 163

 - Checkliste 164

Detraktoren 23

Dezentrale Unternehmensstruktur 65, 75

Dimensional Planning 133

Diversifizierung 41

Diversität im Team 49

dm-drogerie markt, Kundengrundsatz 14

3-Horizonte-Modell 10, 13, 17, 71, 161

- Entwicklung 20, 44

- Freiraum 36, 38–39

- Innovation 17

- Organisation 36, 63

- Personalstrategie 42

- Produktentwicklung 33

- Produkt-Roadmap 45
- Serviceentwicklung 33
- Umsetzung 19
- Wachstum 17
- Wert 21, 27, 32
- zyklisches Vorgehen 33

Dysfunktion, organisatorische 65

E

Echte Agilität 2, 7

Effektivität 59

Effizienz 59

Eigenständiges Team 50

Einzelentscheid, konsultativer 61, 95–96

Empathie 162

Endkunde 7

Entscheidung, dezentrale 75–76

Entscheidungsfindung

- partizipative 98

- übergreifende 93

Entwicklungsteam 103

Experimente 139, 142, 175, 183

- reversible 143

- Safe-to-Fail 143

Experimenteboard 144

Experiment-Owner 145

F

Feedbackschleife 87, 90

- im agilen Unternehmen 91

- managerbasiert 90

Feedbackzyklus 87

Führung

- hierarchische 99

- klassische 99, 101

situative 104

über Ziele 77

verteilte 103

Führungsaufgaben 103

Führungskraft 63

Funktionsübergreifendes Team 55

G

Gegenseitiges Lernen 106

Gesamtsysteme 7

Geschäftsmetriken 8

Geschlossene Fragen 34

Go & See 140

Gold Cards 37

H

Hackathons 37

Hands-off-Management 85–86, 105

Hindernisse *siehe Störungen*

I

Impediments *siehe Störungen*

Innovation 17, 36–37, 87

 Hemmnis 85

Innovationszelle 71

Interdisziplinäres Team 55

Interview 34

Investition 40

Investitionsentscheidungen 39

K

Kaizen 148

Kanban 148

Konsent 61, 93, 98

Konsultativer Einzelentscheid 61, 95

 Ablauf 96

Kontinuierliche Verbesserung 123, 146, 148
Kotter Change Model 127, 129
Kreativität 35, 42
Kundenbedürfnis 13, 16, 36, 162, 175
 verstehen 34
Kundenbegeisterung 8, 13
Kundenkontakt 35, 39
Kundennutzen 13
Kundenwertoptimierendes Team 5, 7, 10, 13, 49, 123, 183
Kundenzufriedenheit 24, 91

L

Large-Scale Scrum 183, 185
Lean Startup 174
 Lebenszyklus 180
 Prozesse 161
Lernen, gegenseitiges 106
Lernerfahrung 162, 166
LeSS-Framework *siehe Large-Scale Scrum*
Lieferndes Team 4, 6
Lösungsraum 162, 166
Lösungsskizze 175

M

Management by Objectives (MbO) 76–77, 82
 Gefahren 82
 Hauptnutzen 82
Manager-led Team 51
Mangel 16
Marktdynamik 20
Marktentwicklung 40
Marktpenetration 40
Matrixorganisation 101
MbO *siehe Management by Objectives*
Mehrheitsentscheid 61

Minimum Viable Product (MVP) 176–177

Mission Command 75

Mitarbeiterführung

disziplinarische 100

fachliche 100

klassische 99

moderne 107

Mitarbeiterrotation 71

Monofunktionales Team 55

More with Less *siehe Large-Scale Scrum*

Mutual Learning 105

MVP *siehe Minimum Viable Product*

N

Net Promoter Score 22

Net Promoter System (NPS) 22

Frage 23

offene Frage 24

Wert, Berechnung 23

Neutrale Kunden 23

Nordstern-Konzept 11, 130, 148, 150, 155

NPS *siehe Net Promoter System*

O

Objectives and Key Results (OKR) 78

Offene Fragen 34

OKR *siehe Objectives and Key Results*

Optimierung, lokale 84, 86

Optimize Value 7

Unternehmen, Eigenschaften 8

Organisation, unterstützende 63

Organisationsentwicklung 123, 127, 139, 142, 144

agile 123, 127

iterative 130

Organisationsmodell 67

Organisationsoptimierendes Team 53
Organisationsprinzip 76
Organisationsstruktur 86, 93
Organisationsveränderungen, Experimente 142
Organisatorische Dysfunktion 65
»out of the box«-Denken 35

P

Pairing 57
PARC-Modell 118
Partizipative Entscheidungsfindung 98
PDCA-Zyklus 139, 153
Peripheriezelle 67
 Netzwerk 68
Personalstrategie 42
Persönlichkeiten 58
Pivot 179
Problem/Solution-Fit 175
Product/Market-Fit 176
Product Owner 58, 103, 131–132, 135
 Rolle 58
Product Ownership 59
Product Vision Board 27–28
Produkt, innovatives 42
Produktentwicklung 13, 33, 40
Produktidee 33
 entwickeln 35
 validieren 35
Produktmanagement 7
Produktreview 25, 30
 Feedback 26
Produkt-Roadmap 45, 47
Produktstrategie 29
Produktvision 13, 27, 133

Promotoren 23

Prototyp 162

Q

Qualitätsbewusstsein 93

R

Reporting Line 100, 102

Return on Investment (ROI) 40

Reverse Accountability 102

Risikomanagement, aktives 139

ROI *siehe Return on Investment*

Rollen, temporäre 105

S

Safe-to-Fail 94, 98

- Experimente 139, 143

Satir Change Model 124

Schatten-Transitionsteam 137

Scrum 2, 58, 103, 119, 131–132, 135, 148, 183

Scrum Master 59, 103, 135

Scrum-Rollen 103

Scrum-Werte 119

Selbstorganisiertes Team 4, 6

Self-designing Team 52

Self-governing Team 53

Self-managing Team 51

Serviceentwicklung 33

Service-Idee 33

- entwickeln 35

- validieren 35

Skalierung 10, 49, 180

- agile 72, 183

- Framework 72

Skalierungsstruktur 72

Slack-Time 37

SMART 153

Soziokratie 105

Sprint-Review 25, 30

Stetigkeit 42

Störungen 64

 bei Lösungen 64

 bei Problemen 64

 im Team 65

Struktur, dezentrale 65

Supporting Line 102

T

Team

 Autonomie 50, 55

 Autonomiegrad 51

 Business-fokussiertes 5

 eigenständiges 50

 Entscheidungen 61

 funktionsübergreifendes 55

 interdisziplinäres 55

 kundenwertoptimierendes 5, 7, 10, 13, 49, 123, 183

 lieferndes 4, 6

 manager-led 51

 monofunktionales 55

 organisationsoptimierendes 53

 selbstorganisiertes 4, 6

 self-designing 52

 self-governing 53

 self-managing 51

 Skalierung 49

 Zusammensetzung 56

Teamaufgabe 49

Teamautonomie 50, 55

 Aufgabenbereiche 50

- Ausprägungen 50
- Teambegleitung 59
- Teamentscheidungen 61
- Teamziel 85
- Teamzusammensetzung 56
- Team-Coach 59
- Teile und herrsche 100
- Time-to-Market 60
- Transition Backlog 131–132
- Transitionsteam 131–132, 135
 - agiles 131
 - Einbindung ins Unternehmen 137
 - Kapazität 137
 - Produktvision 133
 - Rollen 135
 - Sprint-Konzept 136
- Transparenz 8, 24, 98, 147, 184
- True North *siehe Nordstern-Konzept*
- Try, inspect and adapt 1

U

- »Unternehmen als Organismus«-Modell 93
- Unternehmenskultur 118
- Unternehmensstruktur 3, 6
 - dezentrale 75, 93
 - teambasierte 67
- Unternehmensziel 87
- User Research 161

V

- Vanity Metric 83
- Veränderungsprozess 124
- Vertrauen 8
- Vision 29, 130–131, 149
- Visionsstatement 27

W

Wachstum 17

Wert

 aus Kundensicht 13

 für Kunden 13

Wertbildungsrechnung 98

Wertschöpfung 22, 49, 63, 98

 Bewertung 25

Wertschöpfungsstruktur 74

Wertstromanalyse 14

Wissensilo 55

Z

Zeithorizont 21

Zellmodell 67, 69, 76, 93

Zentrumszelle 68

Ziel

 individuelles 85

 innovationserzeugendes 85

 statisches 87

 uneindeutiges 87

Zielkonflikt 85