

# Teil I

## Projektarchitektur und Kommunikationsschnittstellen

---

### ▷ **Software- und Projektstruktur** 3

Unsere Projektstrukturen ähneln in ihrer Komplexität den Softwarestrukturen. Um unsere Softwarearchitektur kümmern sich Architekten, Designer und Entwickler. Aber wer kümmert sich um die Schnittstellen zwischen den Teams und zu den anderen Projektbeteiligten? Der Schlüssel für erfolgreiche Projekte liegt in der Architektur unserer Projektstruktur!

### ▷ **Projektpolitik? Projektumfeldanalyse!** 13

Softwareprojekte sind so gut wie immer eng mit Projektpolitik verknüpft. Wir sollten wissen, wer und was alles mit unserem Projekt zu tun hat und welche Interessen dabei mitspielen, um nicht zur passiven Spielfigur zu werden. Die Projektbeteiligten verfolgen leider nicht immer dasselbe Ziel. Die Projektumfeldanalyse gibt uns ein Mittel in die Hand, die daraus resultierenden Probleme in den Griff zu bekommen.

### ▷ **Projektmarketing** 25

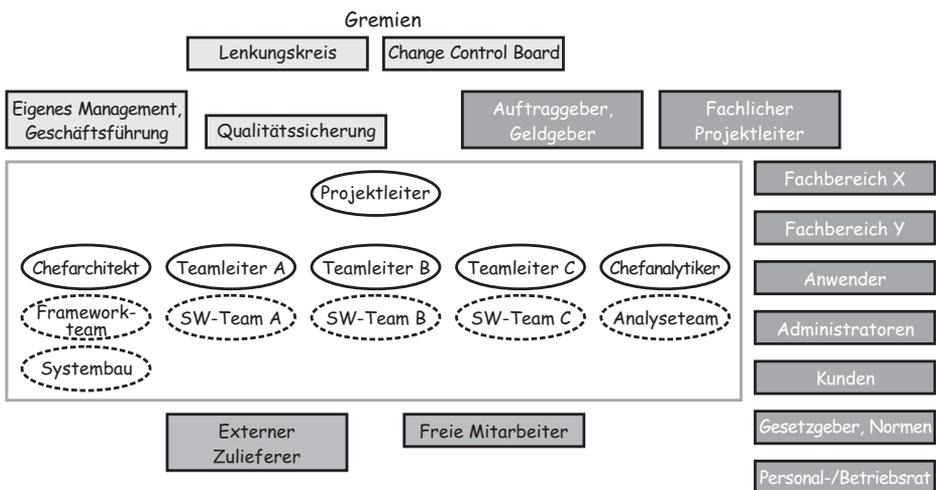
Tue Gutes und rede darüber! Projekterfolg ist eine subjektive Größe, die sich im Auge des Betrachters ergibt. Ein paar kleine Projektmarketingaktivitäten können viel bewegen! Eine mögliche Schlüsselposition kann dabei der Begeisterung zukommen, die wir mit meist kleinen Anpassungen für die Anwender schaffen können.



# 1 Software- und Projektstruktur

## 1.1 Komplexität von Projektstrukturen

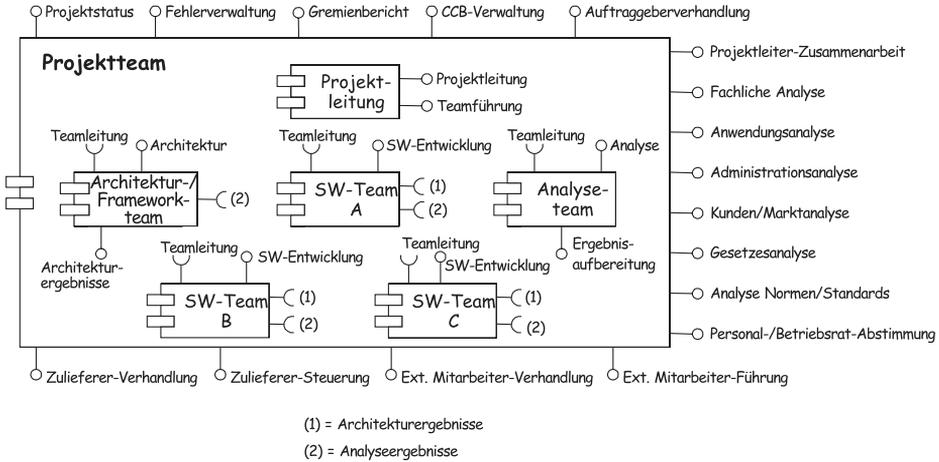
Wir stellen im Rahmen von Projekten hochkomplexe Software her, die maßgeschneidert die Anforderungen erfüllt. Jedenfalls ist das unser Ziel. Die Zeiten, in denen tolle Projekte von einer einzelnen Person gestemmt wurden, sind lange vorbei. Die Komplexität der Anforderungen und der Integration in bestehende Systemlandschaften erfordert angemessene Projektstrukturen. Diese können schnell ähnlich komplex werden wie die zu realisierende Softwarestruktur (Abb. 1.1).



**Abbildung 1.1:** Heutige Projekte erreichen schnell eine komplexe Projektstruktur.

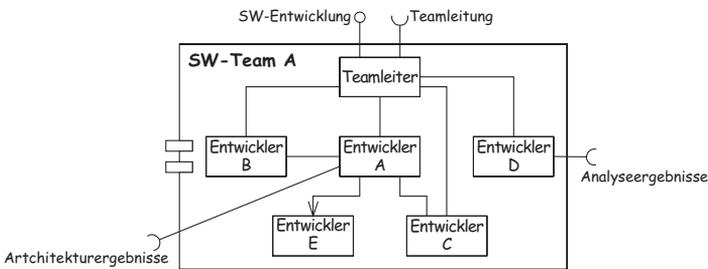
Wenn wir die Managementsicht aus Abbildung 1.1 technisch weiter auflösen und die Schnittstellen explizit modellieren, erhalten wir eine hochgradig vernetzte Struktur, die wir vermutlich in unserer Software nicht dulden würden (Abb. 1.2). Zu aufwendig wären Wartung und Erweiterungen. Die

abgebildeten Interfaces sind nur Beispiele. Im Einzelfall kann Ihre Realität etwas einfacher oder noch komplexer aussehen.



**Abbildung 1.2:** Die Verfeinerung der Projektstruktur aus Abbildung 1.1 führt zu einem komplexen Kommunikationsnetzwerk. (Beispielhafte Darstellung in UML: Alle Interfaces sind bidirektional.)

Innerhalb unserer Teams finden wir zudem eine Mikrostruktur vor, die eigene Kommunikationsinterfaces ausgebildet hat (Abb. 1.3). Die Gesamtkomplexität wird so noch um eine Stufe erhöht.



**Abbildung 1.3:** Zusätzlich zur Struktur aus Abbildung 1.2 finden wir innerhalb unserer Teilteams eine Mikrokommunikationsstruktur vor. (Beispielhafte Darstellung in UML: Alle Interfaces sind bidirektional.)



Glücklicherweise brauchen wir uns der Projektstruktur nicht machtlos zu ergeben. Wir sollten dies auch nicht tun, denn gerade in den Kommunikationsschnittstellen liegt der wesentliche Schlüssel zum Projekterfolg! Genau-

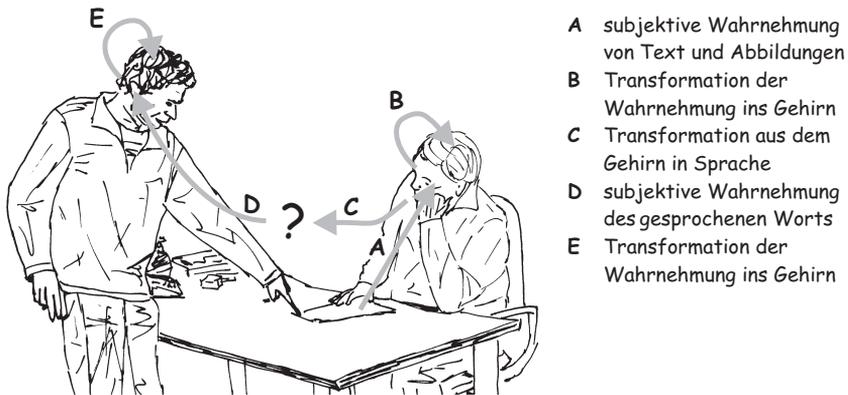
so, wie wir technische Mittel besitzen, um die Abhängigkeiten innerhalb unserer Software in den Griff zu bekommen, gibt es Techniken für die Projektstruktur und Kommunikationsschnittstellen.

Die Themen Selbstorganisation und komplexe Systeme behandeln wir in diesem Buch nicht weiter. Damit haben wir uns eingehend in unserem Buch *Soft Skills für IT-Führungskräfte und Projektleiter* befasst, das wir Ihnen als Weiterführung empfehlen [84].

### 1.1.1 Was sind Kommunikationsschnittstellen?

Wo liegt nun eigentlich das Problem? Die Schnittstellen sind eindeutig definiert und die erwarteten Ergebnistypen detailliert vorgegeben. Es sollte doch alles klar sein!? Doch egal, wie wir kommunizieren: Der Vorgang lässt sich gut mit einem Filterprozess vergleichen. Bei jedem Transfer bleibt etwas auf der Strecke! Die Anteile können je nach Kommunikationskanal und beteiligten Personen variieren, aber es geht immer Kommunikationsinhalt verloren: Wir haben ein Sender-Empfänger-Problem!

Ein einfaches Kommunikationsmodell beschreibt Kommunikation als Folge von Transformationen [69]. Dabei kann es bei jeder Transformation zu Verlusten im Informationsgehalt kommen. Dies erfolgt sowohl zwischen Personen wie auch innerhalb eines Menschen (Abb. 1.4).



**Abbildung 1.4:** Kommunikation als Transformations- und Filterprozess: das Kommunikationsmodell nach Shannon und Weaver [69]

Jede Wahrnehmung ist subjektiv. Dazu kommt das Ausfiltern von Informationen aufgrund der physikalischen Einschränkungen unserer Wahrnehmung. Das Wahrgenommene wird transformiert und als Erinnerung im Gehirn abgelegt. Bei dieser Transformation helfen uns unsere bisherigen Erfahrungen. Sie erleichtern uns das schnelle Erfassen, filtern aber erneut

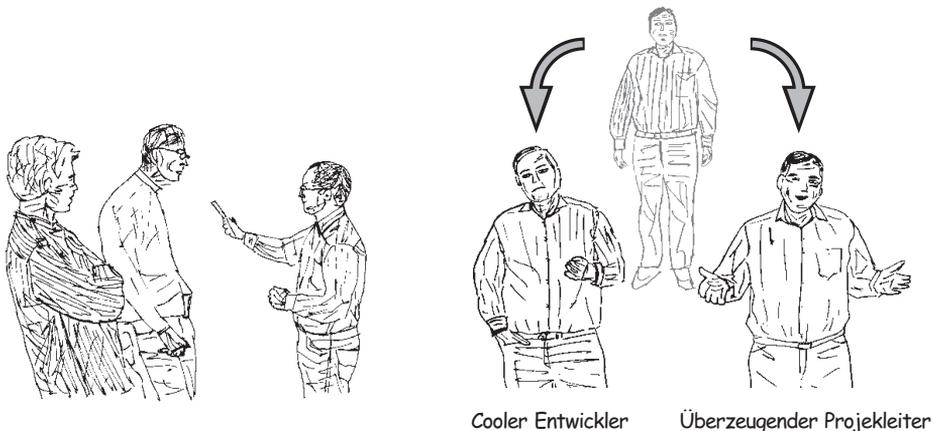
Informationsgehalt aus. Außerdem ist unsere Wahrnehmung durch unsere individuelle Auffassungsgabe begrenzt.

Bei der Rücktransformation aus unserem Gehirn in extern Kommunizierbares, also z. B. Sprache, bleibt erneut einiges auf der Strecke. Besonders bewusst wird uns dies, wenn wir nicht in unserer Muttersprache, sondern in einer Fremdsprache kommunizieren müssen. Wir spüren förmlich, wie Informationsgehalt versickert. Bei unserem Gesprächspartner spielen sich noch einmal dieselben Prozesse ab.

Kommunikationsschnittstellen sind also an sich bereits kritisch. In Teil II ab Seite 41 werden wir Techniken kennenlernen, mit denen wir den Informationsverlust drastisch minimieren können. Das löst zwar noch nicht alle unsere Probleme, doch wir kommen damit deutlich besser voran. Die verbleibenden Probleme beruhen darauf, dass wir es nicht mit technischen Interfaces zu tun haben, sondern mit Menschen.

### 1.1.2 Andere sind nicht komisch, sondern nur anders!

Wie die Objekte einer Klasse sind auch wir Menschen Individuen. Wir sind nach dem gleichen Bauplan erstellt, aber unsere Attribute lassen einen großen Spielraum an Individualität zu. Kommen wir mit Menschen in Kontakt, so verstehen wir diejenigen schneller, die ähnlich wie wir gestrickt sind. Andere hingegen scheinen uns völlig fremd zu bleiben (Abb. 1.5).



**Abbildung 1.5:** Menschen sind absolut individuell (links). Derselbe Mensch kann in bestimmten Situationen unterschiedliche Rollen einnehmen (rechts).

In homogenen Gruppen kann dies leicht dazu führen, dass Menschen, die »anders« sind, abgewertet werden. Gruppen von Softwareentwicklern ver-

halten sich da nicht anders. Die Anwender haben sowieso keine Ahnung, die Administratoren machen alles kaputt und der Fachbereich weiß nicht, was er will, obwohl die eigene Lösung doch eigentlich perfekt ist. Und Manager lassen sich von bunten Bildern eher beeindrucken als von detaillierten Informationen. Wie können wir in so einem Umfeld überhaupt arbeiten?

Achtung! Was wir hier vorschnell machen, ist eine abwertende Klassifizierung. Wir packen die Menschen in »Schubladen«, die so negativ vorbelegt sind, dass wir sie dann gar nicht mehr ernst nehmen können. Hier lauert eine enorme Gefahr, die uns isolieren und projektgefährdende Konflikte erzeugen kann!

Schauen wir uns dazu noch einmal Abbildung 1.1 an. Ob wir wollen oder nicht, wir haben es mit einer Reihe von Menschen zu tun, die keinen technischen Background mitbringen. In deren Bereichen sind ganz andere Qualifikationen notwendig. Wenn sie diese nicht hätten, sondern eher technische, wären sie vermutlich unsere Kollegen. Wenn wir ihre Qualifikationen hätten, säßen wir an deren Stelle!

Diese Heterogenität ist besonders wichtig. Um so etwas Komplexes wie ein Softwareprojekt erfolgreich gestalten zu können, sind verschiedenste Qualifikationen erforderlich. Unsere technischen Fähigkeiten sind dafür absolut notwendig, aber eben nicht ausreichend. In Teil III ab Seite 79 wollen wir Ihnen zeigen, wie wir einen angemessenen Zugang zu Menschen aufbauen können, die ganz anders als wir gestrickt sind. Zusätzlich erfahren Sie dabei, wie Konflikte vermieden werden können. Wir werden uns auch »Schubladen« bauen, die im Gegensatz zu unseren bisherigen (Vor-)Urteilen positiv formuliert und belegt sind.



## 1.2 Bedeutung für IT-Projekte

»Ja natürlich mag es Softwareprojekte geben, in denen die Kommunikation eine besondere Bedeutung hat. Aber doch nicht in meinem Projekt oder meinem Team. Da ist alles klar und geregelt.« Auf solche oder ähnliche Gedanken können Sie nach dem Lesen der ersten Seiten kommen. Und vielleicht ist Ihr Umfeld genau die Ausnahme, die die Regel bestätigt. Aber wie wahrscheinlich ist das?

Vielen Lesern mag es einleuchten, dass die Kommunikation in agilen Projekten einen besonderen Wert hat. Wir gehen jedoch davon aus, dass so gut wie alle Softwareprojekte primär Kommunikationsprojekte sind und sich nur sekundär mit technischen Themen befassen. Verstehen wir wirklich, was die Stakeholder brauchen? Verstehen wir wirklich, welche Probleme der Architekt oder das parallel arbeitende Team sieht? Betrachten wir zur Illustration kurz den Wert von Kommunikation und Zusammenarbeit in zwei sehr unterschiedlichen Kontexten.

### 1.2.1 Agilität: Kleine Projekte, kleine Probleme?

Der Begriff *agil* bedeutet so viel wie *beweglich* oder *leicht zu führen* [20]. Im Agilen Manifest [1] sind die folgenden Prinzipien einer *leichtgewichtigen* Softwareentwicklung festgelegt:

**Menschen und Zusammenarbeit** vor Prozessen und Werkzeugen  
**Funktionierende Produkte** vor umfassender Dokumentation  
**Zusammenarbeit mit dem Kunden** vor vertraglichen Verhandlungen  
**Reaktion auf Veränderung** vor der Einhaltung eines Plans

Die Punkte auf der rechten Seite der Aussagen sind wertvoll, aber kein Selbstzweck. Sie sollen die wichtigeren Punkte der linken, hervorgehobenen Seite sinnvoll und angemessen unterstützen. Dabei sind die Art und der Grad des Einsatzes der unterstützenden Maßnahmen genau zu bestimmen. Was steckt hinter den vier Punkten? Es gibt für die Toolfrage keine *Silver Bullet*. Wichtig ist, wie wir im Prozess oder mit Werkzeugen arbeiten. Unser methodisches Wissen ist gefragt. Es kann weder durch ein Vorgehensmodell noch durch ein Tool ersetzt werden: *A fool with a tool is still a fool!*

Was nützen vollständige Analyse- und Designdokumente, wenn unsere Software nicht einsatzfähig ist, weil sie entweder nicht das macht, was eigentlich gebraucht wird, oder aber zu fehlerhaft ist? Ein umfangreiches Vertragswerk scheint eher zum Verstecken der wenigen relevanten Abschnitte zu dienen als zur Klarheit. Selbst mit noch so viel Aufwand und den besten Rechtsanwälten wird uns kein wasserdichter Vertrag gelingen. Und wenn doch, was machen wir, wenn unser Softwarepartner in Konkurs geht?

Was nützt es dem Auftraggeber, wenn der Plan eingehalten wurde, aber sich in der Zwischenzeit die Anforderungen geändert haben? Dies kann bei einer falschen Priorisierung von Bewertungsfaktoren zu seltsamen Auswüchsen führen.<sup>1</sup> Bei der Bewertung von Aktiengesellschaften kann es dazu kommen, dass, nur um die geplanten Gewinne eines Quartalsberichts einzuhalten, auf Gewinne verzichtet wird, weil von einigen Börsianern auch positive Abweichungen negativ beurteilt werden [12]. So etwas sollte uns Entwicklern nicht passieren. Dafür gibt es vier zentrale Prinzipien agiler Softwareentwicklung, die helfen können, diese Probleme zu vermeiden [3]:

#### ■ Mut

- Vertraue darauf, Probleme, die morgen auftreten, auch morgen lösen zu können.
- Spreche aktuelle Probleme noch heute offen und konstruktiv an.

---

<sup>1</sup>In einer Studie konnten wir 2009 zeigen, dass gerade erfolglose Projekte das Ziel der Umsetzung der *ursprünglichen* Anforderungen besonders oft erreichen, wogegen erfolgreiche Projekte die *aktuellen* Anforderungen umgesetzt haben [85].

### ■ Kommunikation

- Sorge dafür, dass sich die Menschen persönlich kennenlernen.
- Interveniere bei sozialen Problemen zwischen Beteiligten sofort, denn Kommunikationsprobleme sind dringlich.

### ■ Feedback

- Vergewissere dich regelmäßig, auf dem richtigen Weg zu sein.
- Entwickle im Team.
- Führe Reviews mit der jeweiligen Zielgruppe wie z. B. anderen Teams, Fachexperten, Designern, Anwendern durch.
- Führe Akzeptanztests durch.

### ■ Einfachheit

- Suche die einfachste Lösung, um die Anforderungen zu erfüllen. Sie wird immer noch kompliziert genug sein.
- Entwickle für drei ähnliche Probleme lieber drei verschiedene Lösungen als einmal eine komplexe, generische Universallösung, die doch nicht alle Anforderungen erfüllt und um ein Vielfaches teurer werden wird.
- Erst beim vierten Mal bist du dir ausreichend sicher, dass du eine abstrakte, generische Lösung erstellen kannst.<sup>2</sup>
- Vorher ist meist der Aufwand für die Erstellung der generischen Lösung höher als für die drei Einzellösungen.

Die Kommunikation und damit die Kommunikationsfähigkeit wird also als zentraler Erfolgsfaktor gesehen. Dies deckt sich mit unserer Erfahrung aus diversen Negativbeispielen. Obwohl agile Projekte in der Regel von der beteiligten Personenzahl her kleine Projekte sind, spielt die direkte Kommunikation und Kommunikationsfähigkeit bei allen Beteiligten eine überproportional wichtige Rolle.

Schnell kommt es zu Missverständnissen zwischen Entwicklern und Nicht-Entwicklern aus den Fachbereichen oder dem Management. Dadurch entwickeln wir die falsche Software, und die Manager verstehen nicht, was eigentlich vor sich geht. Die Situation wird noch verschärft, indem sowohl die Fachbereiche als auch das Management erheblich mit ihren Erfolgen von uns Entwicklern abhängig sind. Sie sind also abhängig von Menschen, die sie nicht verstehen! Eine solche Situation wird beinahe zwangsläufig zu Ängsten führen – und Angst ist ein schlechter Berater.

---

<sup>2</sup>Über die Anzahl *vier* lässt sich streiten. Hier geht es weniger um Redundanzfreiheit im Code als um den Sinn und Widersinn generischer Lösungen.

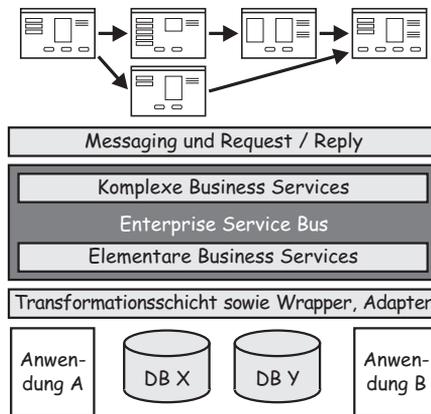
Es kann auch sein, dass unser Gesprächspartner blockiert und wir nicht an die Informationen herankommen, die wir benötigen. Und wir merken das noch nicht einmal. Unser Auftreten kann so weit führen, dass wir uns politische Gegner schaffen, mit denen wir dann Machtspiele austragen, anstatt uns auf unser Projekt zu konzentrieren. Auch innerhalb unseres Entwicklungsteams entgehen uns schnell durch mangelhafte Kommunikation mögliche Synergien. So werden wir kaum eine effiziente Lösung finden können.

Dieses Buch befasst sich im Wesentlichen mit den direkten Kommunikationstechniken. In jedem Teil werden Sie daher Techniken kennenlernen, die Ihnen besonders in agilen Projekten helfen können.

### 1.2.2 SOA: Großprojekte und direkter Kontakt

Hinter SOA (Service-oriented Architecture) steckt eine Architektursicht zur flexiblen Umsetzung von Geschäftsprozessen in Software. Häufig etwas vorschnell wird SOA auf ein technisches Problem zur Bereitstellung von Webservices reduziert. Schauen wir uns erfolgreiche SOA-Projekte an, stellen wir fest, dass die Technologie gegenüber der Methodik häufig eher sekundär ist [73].

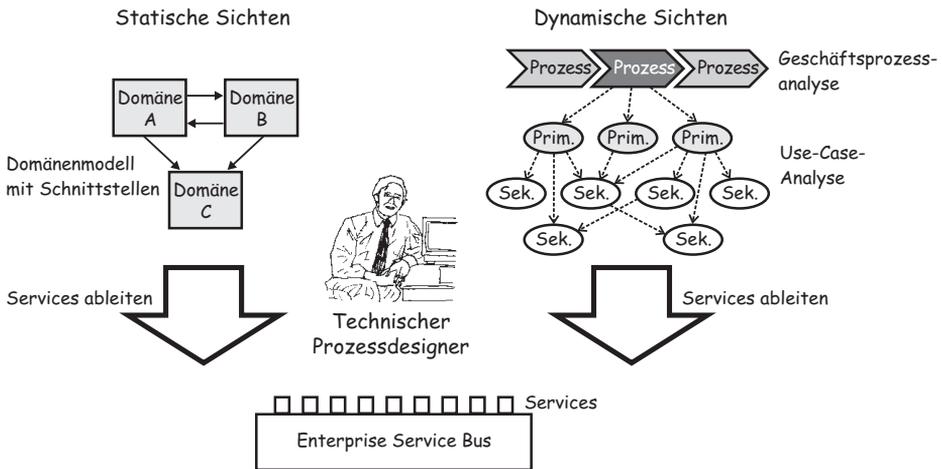
Der Rahmen für SOA-Projekte liegt entgegengesetzt zu den vorher behandelten agilen Projekten. Das Ziel ist dabei die Flexibilisierung der Softwareunterstützung von Geschäftsprozessen in Großfirmen und Konzernen. Waren früher Prozesse über lange Zeiträume konstant, fordert der Markt heute deutlich schnellere Reaktionen, sodass Prozesse durchaus alle paar Monate angepasst werden können (Abb. 1.6).



**Abbildung 1.6:** SOA ist mehr als die Integration von Anwendungen. Durch die Zerlegung in Services kann die Flexibilität der IT-Lösungen erhöht werden, um der Dynamik der Anforderungen besser gerecht zu werden.

Aus technischer Sicht wird mit SOA der bestehende heterogene Zoo von Anwendungen und Informationssystemen durch einen zentralen Enterprise Service Bus gekapselt, der als Schnittstelle für die Anwendungen *Services* zur Verfügung stellt.

Die zentrale Fragestellung lautet: Wie kommen wir zu den einzelnen *Services*? Dies kann nur in enger Zusammenarbeit mit den Fachbereichen erfolgen, um hinterher auch das angestrebte Ziel der Flexibilität zu erreichen. Ausgangspunkt sind die Geschäftsprozesse, die weiter zerlegt werden und zu statischen wie dynamischen Modellen führen, aus denen dann die konkreten *Services* abgeleitet werden. Um die gewünschte Dynamik an Änderungen und Erweiterungen realisieren zu können, erfolgt diese methodische Ableitung der *Services* im direkten Kontakt aller Beteiligten in Form von Workshops. Dabei werden die Geschäftsprozesse analysiert und, falls notwendig, IT-kompatibel modelliert. Des Weiteren werden in einem Domänenmodell Verantwortlichkeiten sowie Schnittstellen festgelegt (Abb. 1.7). Mit den so gefundenen *Services* können dann (hoffentlich) schnell Prozessanpassungen in unseren Anwendungen erfolgen, indem die *Services* als Bausteine aufgefasst und neu rekombiniert werden.



**Abbildung 1.7:** Die Entwicklung einer SOA ist primär ein organisatorischer und geschäftsprozessorientierter Ablauf.

Dabei wird auf IT-Seite eine neue Rolle definiert, eine Art *technischer Prozessdesigner*. Aufgrund der hohen Dynamik der Anforderungsänderungen ist das primäre Arbeitsmittel des technischen Prozessdesigners die Durchführung von Workshops. Er transformiert die Ergebnisse der Workshops in technische Modelle und stimmt diese mit den fachlichen Ansprech-

partnern ab. Meist wird diese Aufgabe von einem kleinen Kernteam wahrgenommen. Für diese Rolle sind offensichtlich neben technischen und methodischen Fähigkeiten starke kommunikative Fertigkeiten notwendig.

In der direkten Kommunikation der Beteiligten werden die Abläufe optimiert und angepasst. Der Prozessdesigner sorgt für die Transformation der fachlichen Sicht in eine technische und für eine angemessene Modellierung. Damit ist er verantwortlich für die Dokumentation der statischen und dynamischen Sichten auf die Abläufe in den betrachteten Bereichen.

## 4 Grundlegende Fragetechniken

»Wer fragt, führt!« lautet eine der Grundregeln in der Gesprächstechnik. Da steckt viel Wahres drin. Durch Fragen können wir Gesprächen z. B. neue Impulse geben, sie in eine bestimmte Richtung lenken oder die inhaltliche Thematik erweitern bzw. begrenzen.

Mit Fragen signalisieren wir Interesse und können unsere Gesprächspartner aktivieren und sogar motivieren. Mit Fragen können wir Klärungen herbeiführen, Dinge verdeutlichen oder Verwirrung stiften.

Das alles geht nicht mit nur einer Art von Fragen, sondern dazu ist ein breites Spektrum an Fragearten notwendig. Wir ITler beschränken uns aber meist auf Fragen, die Probleme analysieren. Wie wir in diesem Kapitel sehen werden, gibt es alternative Möglichkeiten, Lösungen zu *erfragen*.

Situativ passend eingesetzte Fragen können Gespräche lenken, festgefahrene Situationen wieder in Gang bringen oder helfen, inhaltlich auf den Punkt zu kommen. Die Bandbreite unserer Handlungsmöglichkeiten lässt sich dadurch recht einfach erweitern bzw. verbessern.

### 4.1 Informationsfrage

»Wieso, weshalb, warum, wer nicht fragt, bleibt dumm!« Diesen Ausspruch kennen die meisten sicherlich entweder aus der eigenen Kindheit oder von ihren Kindern. Diese *Informationsfragen* sind die sogenannten *W-Fragen*: wie, was, wer, womit, warum usw.

Mit Informationsfragen kommen wir an Wissen heran. Ziel ist, dass sie von unserem Gesprächspartner möglichst ausführlich beantwortet werden. Dieser Typ von Fragen wird als *offene* Fragen bezeichnet im Gegensatz zu den *geschlossenen* Fragen (siehe den Kasten auf Seite 42). So könnten wir unseren Anforderungsverantwortlichen konkret fragen: »Welche Anforderungen haben Sie an die Kontoverwaltung?« Wir erfahren auf diese Weise, wie Konten angelegt, bearbeitet und gelöscht werden sollen.

Zu jedem genannten Punkt können wir mit weiteren Informationsfragen nachhaken. »Welche Daten brauchen Sie mindestens, um ein Konto anlegen zu können?« usw. Wir werden diese Technik der *Rückfrage* im folgenden Kapitel 5 ab Seite 51 im Zusammenhang mit der Sechs-Stufen-



Fragetechnik kennenlernen. Mit Rückfragen zeigen Sie Interesse an einer bestimmten Thematik. Deshalb wirken diese oft positiv auf die direkte Gesprächsbeziehung, insbesondere wenn sich die Gesprächspartner noch nicht so genau kennen.

### Entdecke die Möglichkeiten: offene und geschlossene Fragen

Als *geschlossene Fragen* bezeichnen wir Fragen, die mit *Ja* oder *Nein* beantwortet werden können. Typische Beispiele sind:

- Sind Sie fertig?
- Haben Sie den Hauptschalter eingeschaltet?

Die *offenen Fragen* lassen mehr Spielraum, sie verlangen eine strukturierte Antwort und lassen sich **nicht** mit *Ja* oder *Nein* beantworten. Wir erkennen diese Fragen an den typischen W-Fragewörtern:

- Was kann ich für Sie tun, damit wir rechtzeitig fertig sind?
- Was für Optionen bleiben uns jetzt noch?
- Was genau meinen Sie damit?

Beide Fragetypen haben ihre Vorteile. Wollen wir etwas genau festhalten, eine Übereinstimmung oder eine Ablehnung usw., dann sind geschlossene Fragen bestens geeignet. Wir erhalten eine klare Aussage: ja oder nein. Außerdem können wir so Personen, die dazu tendieren, zu viel zu reden oder sich nicht festzulegen, im Zaum halten.

Wollen wir eine Diskussion in Gang bringen oder offen halten, dann eignen sich offene Fragen dazu. Wir können sie auch einsetzen, um Personen, die eher einsilbig sind, aus der Reserve zu locken.

## 4.1.1 Festgefahrene Situationen lösen

Mit Informationsfragen können wir mehr als nur Fakten sammeln. Auch festgefahrene Situationen können mit ihnen wieder in Gang gebracht werden. Wir arbeiten z. B. weitere Optionen heraus oder identifizieren Blockadeursachen. Zwei Beispiele illustrieren, wie das aussehen kann.

- Ein Softwarefehler in der aktuell ausgelieferten Version legt einen Teil des Betriebs lahm. Für solche Ausnahmesituationen haben wir meist keine Standardlösung parat. Eine gute Idee ist es daher, alle Informations- und Entscheidungsträger zu fragen, welche Möglichkeiten für eine Lösung sie sehen. Manchmal hat einer die rettende Idee, manchmal ist es eine Kombination verschiedener Vorschläge. Die offene Informationsfrage kann uns zur Lösung führen.

- Sie stellen einem potenziellen Auftraggeber Ihr Lösungskonzept für seine Anforderungen vor. Entgegen Ihren Erwartungen springt er nicht sofort darauf an. Irgendetwas scheint ihn zu stören. Die Frage »Welcher Aspekt an meiner Idee stört Sie?« schafft die Möglichkeit, sich an die relevanten Punkte heranzutasten. Vielleicht ist alles nur ein Missverständnis, das Sie schnell auflösen können. Das geht jedoch nur, wenn Sie das wissen!

### 4.1.2 5 × Warum

An einem Beispiel wollen wir nun die 5×-Warum-Fragetechnik erläutern. Die Intention, die hinter Softwareanforderungen steht, kann für uns von großem Interesse sein. Unsere fachlichen Ansprechpartner kennen häufig nur die technische Sicht der Altsysteme als Anwendungsmöglichkeit. Das kann den Horizont für die Anforderungsgestaltung unnötig einengen.

Altsysteme bieten eine technische Umsetzung der Anforderungen. Diese Umsetzung kann bei einem neuen System vielleicht anders erfolgen und damit z. B. benutzerfreundlicher ausfallen. Um solche Lösungen entwickeln zu können, müssen wir hinter die Kulissen schauen und erkennen, welche fachliche Motivation hinter einer konkreten Anforderung steckt. Das ist manchmal gar nicht einfach herauszubekommen.

Die *Warum*-Frage ist dabei hilfreich. Leider reicht einmal *Warum* fragen oft nicht aus, um an den Kern zu gelangen. Der kleine Beispieldialog zwischen Entwickler (links) und Anwender (rechts) illustriert dies.

»Warum benötigen Sie die vollständigen Kundendaten auch in dieser Maske?«

»Weil wir sie hier ändern!«

»Warum möchten Sie in diesem Ablauf die Kundendaten ändern können?«

»Weil wir keine andere Änderungsmöglichkeit haben!«

»Eigentlich benötigen Sie also eine einfache Möglichkeit, die Kundendaten zu ändern?«

»Ja, da haben Sie Recht!«

Der Abschluss dieses Dialogs findet über eine *geschlossene Frage* statt (siehe den Kasten auf Seite 42). Damit erreichen wir die Bestätigung zu unserer neuen Erkenntnis. Das Warum-Frageszenario kann bis zu fünf Mal erfolgen, bis wir wirklich an der ursächlichen fachlichen Intention angekommen sind. Mehr als fünf Warum-Fragen in Folge erzeugen erfahrungsgemäß kein

sinnvolles Ergebnis mehr. Wenn sich die Intention nicht in maximal fünf Runden klären lässt, dreht man sich im Kreis.

Die Warum-Fragen führen uns gleich zum nächsten Problem: Manche Personen fühlen sich durch diese Fragen angegriffen und fangen an, sich zu rechtfertigen. Das liegt an dem rückwärts gerichteten Aspekt, den eine Warum-Frage oft beinhaltet. Unser Ziel ist es jedoch, an Informationen zu gelangen. Rechtfertigungen für das bisherige Vorgehen lenken davon nur ab.

### 4.1.3 Risiko Rechtfertigung

Wie eben gezeigt, kann eine der *W-Fragen* problematisch sein: die Frage nach dem *Warum*. Sie kann dazu führen, dass sich unser Gesprächspartner angegriffen fühlt. Er wird dann eine Verteidigungsposition einnehmen und evtl. sich zu rechtfertigen beginnen.

Die einfachste Möglichkeit, den Druck aus einer Warum-Frage zu nehmen, ist sie zu umschreiben. »Wozu benötigen Sie das Feld genau in dieser Maske?« oder »Was genau erreichen Sie über diese Art der Verteilung?« sind Beispiele dafür, das Wort *warum* zu meiden.

Was ist das Problem mit Rechtfertigungshaltungen? Sie bringen uns in der Regel nicht weiter, sie sind meist kontraproduktiv. Sobald wir eine Rechtfertigungshaltung erkennen, sollten wir versuchen, den Druck aus der Kommunikation herauszunehmen. Die Situation lässt sich z. B. grundlegend entspannen, indem Sie sagen: »Ich möchte nur die Hintergründe besser verstehen. Entschuldigen Sie bitte, wenn Sie sich angegriffen fühlen. Das liegt nicht in meinem Interesse.« Dies ist ein einfaches Beispiel, auf die Meta-Ebene zu gehen, um Probleme zu lösen (Seite 72).

## 4.2 Mit Fragen auf den Punkt kommen

Wollen wir eine Thematik einschränken oder weiter konkretisieren, eignen sich dazu geschlossene oder Alternativfragen. Mit *Alternativfragen* reduzieren wir den Antwortbereich auf wenige oder häufig auf nur noch zwei Möglichkeiten: »Welche der drei vorgestellten Designalternativen erscheint Ihrer Meinung nach angemessener?« So treiben wir die Entscheidungsfindung voran. Durch die Fragestellung wird der Kern des zugrunde liegenden Problems auf den Punkt gebracht. Mit der Beantwortung dieser Frage ist auch eine Entscheidung getroffen.

Mit geschlossenen Fragen wollen wir ein *Commitment* erreichen: »Sind Sie mit dieser Lösung zufrieden?« Dadurch wird eine erneute Diskussion am ehesten vermieden. Diese abschließenden Zustimmungen sind wichtig, um jeden Teilnehmer explizit an der Entscheidung zu beteiligen. Leider

wird viel zu häufig das Ausbleiben von Widerspruch als Zustimmung gewertet. Es ist offensichtlich, dass dies in der weiteren Zusammenarbeit zu Konflikten führen kann. Kontrollieren Sie also ruhig öfter die Zustimmung: »Sind Sie damit einverstanden?« Das ist deutlich wertvoller und ergiebiger als das weitverbreitete: »Hat jemand Einwände?«

Bei der Frage nach Einwänden liegt die Gefahr darin, dass wir die Unentschlossenen nicht gut zu fassen bekommen und Hinweise auf mögliche Risiken übersehen. Dies kann später leicht zu Konflikten führen, denn diese Beteiligten hatten eben nicht explizit zugestimmt! Mit der positiven Frage können wir bei jedem Einzelnen das Commitment abholen und erhalten einen Eindruck von der Stimmung aller Gesprächspartner. Das ist umso wichtiger, mit je mehr Menschen wir kommunizieren. Zu leicht können wir Unklarheiten oder Widerstände übersehen.

Durch das Weglassen von Alternativen konzentrieren wir uns auf das Wesentliche. Dies kann z. B. dazu dienen, für einen Entscheider die Komplexität zu reduzieren. Ein Manager ist oftmals darauf angewiesen, dass seine Entscheidungsfindung entsprechend vorbereitet wird. Je mehr Abstand eine Position von der konkreten Arbeit hat, desto abstrakter stellt sie sich dar (Abb. 4.1). Entscheidungen können dann nicht mehr unter Berücksichtigung der gesamten Detailtiefe getroffen werden.

Ein Manager ist also auf die enge, vertrauensvolle und arbeitsteilige Zusammenarbeit mit den Entwicklern angewiesen – ob als Führungskraft in der Softwareentwicklung oder aufseiten des Auftraggebers bzw. im Fachbereich. Die Grundlagen für Managemententscheidungen werden sinnvollerweise über alle Ebenen zusammengetragen. Dazu werden sie entsprechend den Sichtweisen verdichtet und von ihrer Komplexität her reduziert. Dies kann z. B. über Modelle erfolgen.

Eine andere Form der offenen Frage erwartet nur ein kurzes, konkretes Ergebnis: »Wie lautet der Subversion-Parameter dafür?« Antwort: »/d«. Durch solche Fragen fühlen wir uns ausgefragt. Man möchte dem Fragesteller eigentlich nur antworten: »Schau doch selbst nach!« Seien Sie also bitte vorsichtig bei dieser Art der offenen Fragen. So strahlen Sie kaum Kompetenz aus.

Geschlossene Fragen können durch das bereits erwähnte Weglassen von Alternativen auch einen manipulativen Charakter haben, wie Sie es vielleicht schon einmal in einem Gespräch mit einem Versicherungsvertreter erlebt haben:

- |                                                                     |                      |
|---------------------------------------------------------------------|----------------------|
| »Sie lieben Ihre Familie?«                                          | »Ja!«                |
| »Sie möchten Ihre Familie auch in Krisenzeiten abgesichert wissen?« | »Ja!«                |
| »Dann brauchen Sie unbedingt eine Lebensversicherung!?«             | »Äh, ja, mmh . . . « |

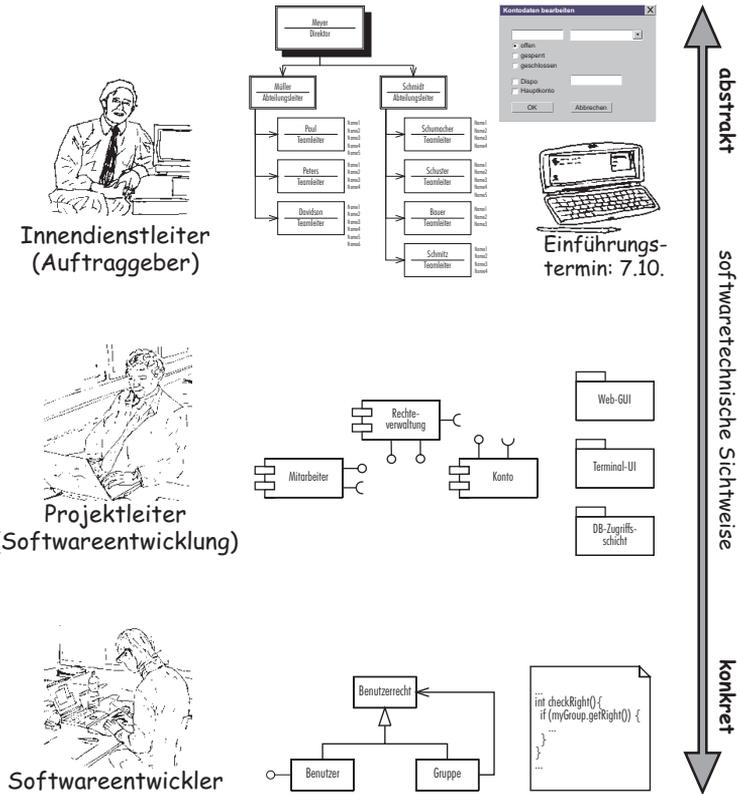


Abbildung 4.1: Je größer der Abstand einer Führungskraft von der konkreten Arbeit ist, desto abstrakter wird die Sicht auf die Softwareentwicklung.

Dieses Risiko ist bei Einschränkungen, Reduktionen oder Konkretisierungen gegeben. Wenn man methodisch-analytisch vorgeht, wird ein solcher Versuch bemerkt werden. Ansehen und Respekt gegenüber dem Fragesteller werden deutlich leiden. Dies ist übrigens auch der Fall, wenn nicht offen gegen einen Manipulationsversuch vorgegangen wurde. Vielleicht hat sich nur keiner getraut oder es erschien nicht wichtig genug. Der Schaden ist auf jeden Fall groß!

### 4.2.1 Gespräche lenken: Wer fragt, führt!

Gespräche lassen sich durch Rückfragen gut lenken, wir wissen nur vor dem Gespräch nicht wohin. Im Gesprächsverlauf können wir durch Rückfragen die für uns interessanten Zusammenhänge und Details erfahren. Das funktioniert gut, hält ein Gespräch für verschiedene Richtungen offen und manipuliert den Gesprächspartner nicht. Dies gilt besonders für

Krisengespräche! Die Kunst ist es dabei, nicht auf der Problemebene zu verharren, sondern auf die Lösungsebene zu gelangen. Durch die richtigen Rückfragen kann dies gelingen (Tab. 4.1). Es gilt, Lösungen zu finden, anstatt Probleme zu diskutieren [46].

konstruktive Rückfrage	destruktiv
Was brauchen Sie, um das Problem erfolgreich zu lösen?	<del>Wie ist das Problem entstanden?</del>
Wenn durch einen Jini alle Probleme gelöst würden, was genau wäre dann anders?	<del>Wer hat das Problem verursacht?</del>
Wie können Sie sich in Zukunft anders verhalten?	<del>Warum haben Sie das so gemacht?</del>
Was genau soll in Zukunft anders sein?	<del>Was ist das Schlimmste dabei?</del>
Woran bemerken andere, dass das Problem gelöst ist?	<del>Warum?</del>

**Tabelle 4.1:** Konstruktive Rückfragen (links) führen zu Lösungen [46].

Die konstruktiven Rückfragen sind in die Zukunft gerichtet, die destruktiven eher in die Vergangenheit. Es gilt also, auf die Gestaltung der Zukunft mit möglichst vielen Optionen zu fokussieren, anstatt die Vergangenheit zu analysieren.

Destruktive Problemanalysen haben ihre Stärken, wenn es darum geht, Prozesse oder Abläufe zu optimieren, sie sind weniger geeignet bei einer akuten Problemlösung. Die Stärken vieler ITler liegen aber gerade im Analysieren komplexer Situationen. Wir verwenden daher gerne die destruktiven Rückfragemuster, die uns leider kaum konstruktiv zu neuen Lösungen bringen. Umso wichtiger ist daher immer wieder das Besinnen auf die konstruktiven, zukunftsorientierten Rückfragen.

Leider kann auch die Rückfragetechnik manipulativ in Form von Suggestivfragen eingesetzt werden. Obwohl Suggestivfragen ähnlich verwendet werden, sind sie keine echten Rückfragen. Mit Suggestivfragen versuchen wir, Gesprächspartner zu beeinflussen: »Sie sind doch auch der Meinung, dass uns ein testgetriebenes Vorgehen nur behindert? Sie glauben doch nicht etwa, dass wir unfähig sind, Anforderungen umzusetzen?« Der manipulative Charakter wird schnell deutlich mit den bereits beschriebenen negativen Konsequenzen für den Fragesteller.

Rückfragen sind also ein hervorragendes Mittel, Gespräche konstruktiv zu lenken. Gerade wenn es um Problemlösungen geht, können konstruktive, zukunftsorientierte Rückfragen den Weg zu kreativen Lösungen ebnen. Analytische Rückfragen sind dabei auf ein absolutes Minimum zu reduzie-



ren. Als Rückfragen kaschierte Suggestivfragen unterbinden dagegen ein konstruktives Gespräch. Suggestivfragen sind keine *echten* Rückfragen.

## 4.2.2 Motivation und Provokation

Fragen eignen sich auch dazu, zu motivieren bzw. zu provozieren. Natürlich lassen sich auch unabhängig von Fragen Lob und Anerkennung ausdrücken. Um z. B. Anerkennung wohl dosiert und einfach zu geeigneten Zeitpunkten einfließen zu lassen, können Sie die motivierende Aussage mit einer Frage verbinden und erhalten so eine *motivierende Frage*. Dies erhöht die Gesprächsbereitschaft, wobei die Beziehungsebene im Mittelpunkt steht wie z. B.:

- »Wie beurteilen Sie als langjähriger und erfahrener Mitarbeiter das Projekt?«
- »Sie haben einen tiefen Einblick in die technischen Zusammenhänge. Wie beurteilen Sie die Lösungsideen?«

Eine weitere Möglichkeit, auf der Beziehungsebene mit den Gesprächspartnern zu arbeiten, ist die Provokation. Damit können wir in eine träge Gesprächssituation wieder Bewegung bringen. Ein »Ist das schon alles, was Ihnen dazu einfällt?« ruft sicher eine Reaktion hervor.

Leider wird die Reaktion nicht immer positiv nutzbar sein, denn Provokationsfragen sind riskant. Wenn wir die Gesprächspartner nicht gut kennen oder die Beziehung bereits gestört ist, wird sich eine Provokation kaum positiv auswirken, sondern eher die Situation verschlechtern. Hier ist Fingerspitzengefühl gefragt. Wenn Sie sich also nicht ganz sicher sind, sollten Sie die Provokation besser nicht einsetzen.

## 4.3 Anregende Fragen

Trotz unserer breit aufgestellten Fragetechnik stecken wir manchmal in Gesprächen fest, weil unserem Gesprächspartner keine Antwort einfällt. Dann können kreative Fragen weiterhelfen, indem wir Perspektiven wechseln oder methodische Lösungsideen einfordern.

### 4.3.1 Die Perspektive verändern!

Wenn z. B. in einer fachlichen Analyse unserem Gesprächspartner keine Antworten mehr einfallen und wir sichergehen möchten, auch wirklich an alles gedacht zu haben, können wir durch einen Perspektivwechsel eine kreative Kontrollfrage stellen. »Was würde denn Ihr Chef auf diese Frage antworten?« regt oft den Antwortfluss wieder an.

Ein solcher Perspektivwechsel kann in viele verschiedene Richtungen gehen. Typische Wege sind entlang der Hierarchie auf- bzw. abwärts (Chef – Kollege – Mitarbeiter) oder entlang der typischen Prozesskette (Außendienstmitarbeiter – Verwaltung – Revisor – Kunde). Aber auch das private Umfeld kann hilfreich für einen Perspektivwechsel sein, wenn wir z. B. Projekterfolgskriterien abklären: »Woran merkt Ihr Partner zuhause, dass Ihr Projekt erfolgreich läuft?« Diese Art des Perspektivwechsels wird auch als zirkuläre Frage bezeichnet.

Wir können auch die zeitliche Perspektive ändern. »Wie sind Sie früher mit solchen Problemen umgegangen?« richtet den Blick auf die eigenen Stärken. »Was wird anders sein, wenn Sie das Problem gelöst haben?« lenkt den Blick vom aktuellen Problem auf die zukünftige Lösung.

Ein solcher Wechsel hin zur Sicht auf die Lösung kann auch über die Wunderfrage erfolgen. »Wenn ein guter Geist über Nacht das Problem gelöst hat, woran würden Sie es morgen früh im Büro merken?« kann Wunder wirken bei der Anregung zu neuen Lösungsansätzen. Diese Fragen fallen in die Kategorie der hypothetischen Fragen.

### 4.3.2 Den Lösungsweg methodisch erarbeiten!

Wenn unser Gesprächspartner sich kaum in der Lage sieht, unsere Fragen inhaltlich zu beantworten, und auch ein Perspektivwechsel nicht ausreichend war, können uns Prozessfragen weiterhelfen. »Was brauchen Sie, um diese Frage beantworten zu können?« ist eine typische Prozessfrage. Wir lassen dabei die Verantwortung für die Lösung beim verantwortlichen Gesprächspartner, ohne ihn in ungeeigneter Weise zu beeinflussen.

Der Vorteil dieser Fragen liegt darin, dass wir jetzt gemeinsam einen Weg finden können, wie wir zu belastbaren und ausreichenden Antworten kommen. Gleichzeitig bleibt unser Gesprächspartner in der Verantwortung, insbesondere, wenn der Weg nicht funktioniert. Der Ball liegt mit diesen Fragen immer wieder im Spielfeld unseres Gesprächspartners.

Wir dürfen es mit keiner Fragetechnik übertreiben und ganz speziell nicht mit den Prozessfragen. Gezielt eingesetzt bewirken sie jedoch Wunder bei der Lösung festgefahrener Probleme.

## 4.4 Fazit

Bevor wir im nächsten Kapitel ein ganzes Konzept zur Fragetechnik betrachten, fassen wir den bisherigen Stand kurz zusammen. Was hat es mit den Fragen auf sich?

Wir haben offene und geschlossene Fragen kennengelernt. Über eine geschlossene Frage können wir einen Themenbereich abschließen (»Gibt es

noch Fragen zu klären?«) oder Zustimmung bzw. Ablehnung zu einer Entscheidung klären (»Sind alle hier damit einverstanden?«). Das ist wichtig, bevor wir uns in einer Diskussion einem neuen Themenkomplex widmen. Eine geschlossene Frage kann mit *Ja* oder *Nein* beantwortet werden.

Über offene Informationsfragen, die W-Fragen, erschließen wir neue Diskussionsinhalte. Wir können dabei in die Detailtiefe gehen (»Wie genau kann ich mir das vorstellen?«) oder den Umfang erschließen (»Was gibt es noch?«). Darüber können wir ein Gespräch steuern, weshalb sie auch als Lenkungsfragen bezeichnet werden. Offene Fragen können nicht sinnvoll mit *Ja* oder *Nein* beantwortet werden.

Die Kreativität lässt sich über Perspektivwechsel anregen. Über zirkuläre Fragen können wir uns in die Sicht anderer Menschen versetzen lassen. Mit hypothetischen Fragen schieben wir ein hemmendes Problem in den Hintergrund und fokussieren uns mehr auf die Lösung. Denkblockaden lassen sich so aufbrechen.

Bleibt eine Denkblockade dennoch bestehen, können wir über Prozessfragen gemeinsam zu ersten notwendigen Schritten kommen. Eine Frage nach dem Muster »Was brauchen Sie, damit ...« führt uns auf die Meta-Ebene. Wir sprechen darüber, was benötigt wird und welche Voraussetzungen wertvoll sind, um weiterzukommen. Gleichzeitig bleibt der Gesprächspartner in der Verantwortung für die Lösung seiner Aufgabe, da er *seine* Voraussetzungen und notwendigen Rahmenbedingungen formuliert.