

# Auf Zielen, nicht auf Sand gebaut: Wie gute Ziele zum richtigen System führen

*Das Requirements Engineering (RE) fokussiert sich darauf, ein gewünschtes System mit Hilfe von Anforderungen zu beschreiben. Aber wie kann man wissen, ob man das richtige System beschrieben hat? Und woran lässt sich am Ende der Erfolg messen? Man muss die Ziele kennen, die man mit Hilfe des gewünschten Systems zu erreichen hofft. Aber Ziele sind zu mehr zu gebrauchen als nur zur Kontrolle des Erfolgs: Sie sind das Fundament, auf dem die Anforderungen stehen, und unerlässlich, um zwischen richtigen und falschen Anforderungen unterscheiden zu können. Wie das geht, soll im Folgenden gezeigt werden.*

## Richtige Anforderungen und ihre Basis

Anforderungen beschreiben Eigenschaften oder Verhaltensweisen eines gewünschten Systems (vgl. [Wei14]). Das System, das mit den Anforderungen beschrieben wird, kann dabei irgendein System sein: Ein soziotechnisches System wie ein ganzer Flugplatz oder eine Versicherung, eine komplexe Maschine, bestehend aus Hardware und Software wie ein Airbus A380, oder auch nur eine bestimmte Software. An jedes dieser Systeme lassen sich Anforderungen richten. Dabei ist es für gute Anforderungen unerlässlich, dass allen Beteiligten klar ist, von welchem System die Rede ist.

Die Aufgabe des klassischen *Requirements Engineering (RE)* ist es, die *richtigen Anforderungen* an das gewünschte System zu finden und zu dokumentieren. Um das zu tun, werden die unterschiedlichsten Quellen wie Stakeholder, Dokumente, Prozesse, bestehende Systeme und Fachleute hinzugezogen. Was dabei leicht vergessen gehen kann, ist die *Basis*, auf der alle Anforderungen stehen sollten: die *Ziele*, die der Auftraggeber mit dem System erreicht will.

### Beispiel WebShop

Eine Versicherung möchte ein neues System *WebShop* entwickeln oder beschaffen, mit dem Endkunden einfache Policen direkt über das Web abschließen können. Dafür sollen nun Anforderungen erhoben werden. Schnell kommt die Frage nach den Zielen auf, die mit dem WebShop erreicht werden sollen.

### System und Kontext

Das gewünschte System genügt sich nicht einfach selbst, vielmehr muss es einen Grund geben, warum das System genau so beschaffen sein soll, wie es in den Anforderungen beschrieben wird. Dieser Grund liegt in

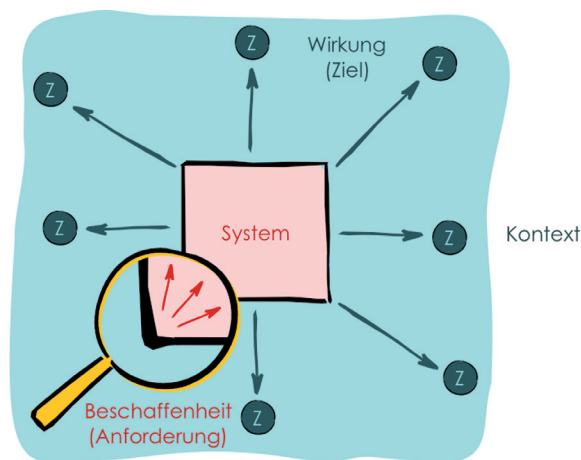


Abb. 1: Gewünschtes System und sein Kontext.

der *Wirkung* oder den *Effekten*, die das gewünschte System in seinem Kontext erzielen soll, den zu erreichenden *Zielen* eben. Ein Ziel ist „ein in der Zukunft liegender, gegenüber dem Gegenwärtigen im Allgemeinen veränderter, erstrebenswerter und angestrebter Zustand“ (vgl. [Wik]). Werden die Ziele nicht erreicht, so hat das System seinen Nutzen nicht erbracht und somit keine Daseinsberechtigung. Die Ziele, die mit Hilfe des Systems erreicht werden sollen, sind dabei etwas grundlegend anderes als die in den Anforderungen beschriebenen Eigenschaften und Verhaltensweisen des Systems.

Das *Business Motivation Model (BMM)* der OMG (vgl. [OMG15]) bietet einen Ansatz, der es erlaubt, nicht nur Zielhierarchien, sondern auch Strategien und Taktiken als Mittel zur Umsetzung dieser Ziele zu modellieren. Mit dem BMM lassen sich zudem sehr gut Lösungsvarianten beschreiben und mit einer *SWOT-Analyse (Strength/Weaknes/Opportunity/Threat)* ihre Stärken, Schwächen, Chancen und Gefahren bewerten

**Kasten 1: Seitenblick auf das BMM der OMG.**

Die Ziele beschreiben den angestrebten Zustand im Kontext des gewünschten Systems und nicht den Zustand des Systems selbst (siehe **Abbildung 1**). Bei der Formulierung von Zielen sollte man sich dessen immer im Klaren sein. Eine gute Möglichkeit, Ziele zu erarbeiten, bietet das *Business Motivation Model (BMM)* der Object Management Group (OMG) (siehe **Kasten 1**).

### Beispiel WebShop

Die Versicherung hat für das System *WebShop* folgende Ziele formuliert:

- zusätzlicher Umsatz
- Entlastung der Verkäufer von Routineaufträgen

Diese Ziele beschreiben nicht den gewünschten Zustand des Systems *WebShop* selbst, sondern den des Kontexts, in dem der *WebShop* betrieben wird, also des sozioökonomischen Systems „Versicherung“.

### Verifikation und Validierung

Bei einem Systemabnahmetest wird geprüft, ob das System die an es gerichteten Anforderungen erfüllt. Diese *Verifikation* kann unmittelbar nach der Fertigstellung

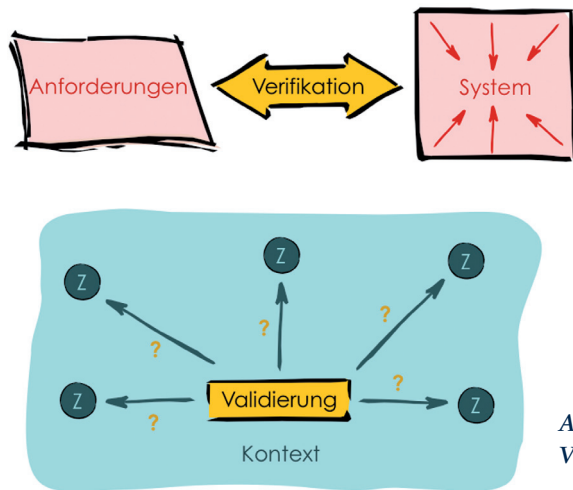


Abb. 2: Verifikation und Validierung.

und noch vor Inbetriebnahme des Systems erfolgen. Sie zeigt, ob das System *richtig gebaut* wurde. Der Blick ist von außen, d.h. vom Umfeld des Systems, gegen innen auf das System gerichtet. Aber selbst ein System, das alle Anforderungen erfüllt, ist keine Garantie dafür, dass damit auch die gesteckten Ziele erreicht werden.

Die zweite Prüfung, ob die Ziele erreicht wurden und das System die gewünschte Wirkung tatsächlich erzielt, kann meist erst zu einem späteren Zeitpunkt durchgeführt werden, da der in den Zielen angestrebte Zustand im Kontext nicht sofort eintritt. Erst diese *Validierung* zeigt, ob das *richtige System* gebaut wurde. Der Blick ist nach außen auf den Kontext gerichtet (siehe **Abbildung 2**).

Leider wird diese zweite Prüfung in der Praxis vielfach nicht durchgeführt. Gründe dafür können sein, dass die Ziele nicht klar genug formuliert wurden oder dass nach einer gewissen Zeit das Interesse an einer Überprüfung nicht mehr vorhanden ist. Nicht selten will man aus politischen Gründen gar nicht so genau wissen, ob die Ziele erreicht wurden.

Ein weiterer Grund ist, dass die Resultate der Validierung interpretiert werden müssen. Man muss genau untersuchen, warum die Ziele erreicht oder eben nicht erreicht wurden. Die Ziele können trotz des richtigen Systems nicht erreicht werden, aber auch trotz des falschen Systems erreicht werden. Bei einem System, das die Verifikation besteht (also gemäß den Anforderungen gebaut wurde), das aber seine Ziele nicht erreicht, ist die Wahrscheinlichkeit dennoch groß, dass man nicht die richtigen Anforderungen hatte.

**Beispiel WebShop**

Ein Jahr nach der Einführung des Systems WebShop stellt die Versicherung fest, dass ihre Verkäufer tatsächlich weniger Routi-

neaufträge abgewickelt hatten, dass aber der Umsatz insgesamt leicht zurückging. Die gesetzten Ziele wurden also nicht vollständig erreicht. Eine detaillierte Analyse zeigt aber, dass die Umsätze in der Versicherungsbranche insgesamt rückläufig waren. Daraus schließt man, dass die Ursache für das Nicht-Erreichen des Zieles nicht in Mängeln des Systems WebShop liegt.

**Woher kommen die Ziele?**

In einer idealen Welt hat eine Organisation eine dokumentierte *Zielhierarchie*, bestehend aus Zielen, die jeweils ihrerseits in Teil- oder Unterziele aufgeteilt sind, vielleicht sogar als Teil eines umfassenden Motivationsmodells (vgl. [OMG15]). Das gewünschte System soll die Erreichung einiger dieser Ziele unterstützen.

Fehlt eine solche Top-Down-Zielhierarchie oder ist sie nicht dokumentiert, muss man versuchen, wenigstens die unmittelbaren Ziele, die mit dem gewünschten System erreicht werden sollen, zu ermitteln und zu dokumentieren. Natürlich sollen auch die so gefundenen Ziele letztlich den Zielen

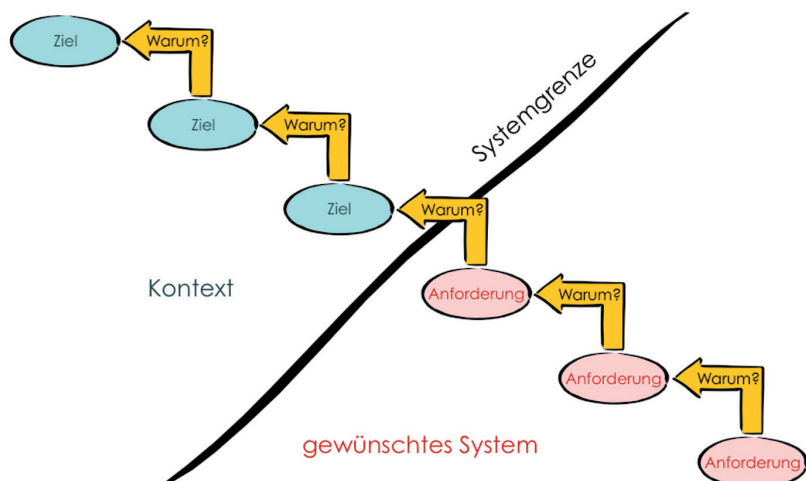


Abb. 3: Die Warum-Kaskade mit Anforderungen und Zielen.

der Organisation dienen, müssen sich also *bottom-up* in eine Zielhierarchie einbinden lassen.

Um die Ziele eines gewünschten Systems zu erfahren, kann man verschiedene Fragen stellen:

- Welchen Effekt, welche Wirkung will ich mit dem gewünschten System erzielen, wozu brauche ich das System?
- Was möchte ich tun können?
- Welche übergeordneten Ziele gibt es im Kontext, kann ich daraus neue Teil- oder Unterziele ableiten (und so top-down an eine Zielhierarchie anknüpfen)?

Auf den ersten Blick ist es nicht immer einfach, zwischen einer Anforderung an das gewünschte System einerseits und einem erwünschten Effekt andererseits zu unterscheiden. Aber solange man klar zwischen dem *gewünschten System* und seinem *Kontext* unterscheidet, kann man eigentlich immer herausfinden, ob von der *Beschaffenheit* oder der *Wirkung* des gewünschten Systems die Rede ist.

**Beispiel WebShop**

Das Ziel „Entlastung der Verkäufer von Routineaufträgen“ wurde aus einem übergeordneten Ziel abgeleitet, nämlich: „Verkäufer verwenden mindestens 20 Prozent ihrer Zeit zur Pflege bestehender Kundenbeziehungen.“

**Die Warum-Frage**

Nennt ein Stakeholder statt eines Ziels eine Beschaffenheit des gewünschten Systems, so kann man durch die Warum-Frage das zu Grunde liegende Ziel erfahren (siehe **Abbildung 3**). Das Hinterfragen einer Anforderung führt nicht immer direkt zu den Zielen,

es kann auch sein, dass man erst nach mehreren Fragen bei einem Ziel anlangt. Dabei ist nicht immer klar, wann man genug gefragt hat. Ein Warum zu wenig und man hat immer noch eine Anforderung und nicht ein Ziel vor sich, ein Warum zu viel und man hat ein Ziel, das nicht direkt durch das gewünschte System beeinflussbar ist.

Will man diesen Schwierigkeiten bei der Frage nach dem Ziel aus dem Weg gehen, kann man auch nach dem Gegenteil eines Ziels fragen, nach unerwünschten Effekten, unbefriedigenden Zuständen und anderen Problemen:

- Welche *unerwünschten* Effekte und Wirkungen will ich mit dem gewünschten System in Zukunft vermeiden, was passt mir an der aktuellen Situation *nicht*?
- Was kann ich mit dem aktuellen System *nicht* tun?
- Welche übergeordneten Ziele im Kontext werden zurzeit nicht erreicht?

Diese aktuellen Mängel, auch als *Painpoints* (wunde Punkte) bezeichnet, sind oft einfacher zu benennen als die Ziele selbst. Außerdem ist die Gefahr viel kleiner, dass anstelle von Zielen Anforderungen gefunden werden. Invertiert man die Painpoints, führt das mehr oder weniger direkt wieder zu Zielen, je nachdem, wie konkret der Painpoint formuliert wurde.

**Beispiel WebShop**

Hinterfragt man die (viel zu technologische) Anforderung „Der WebShop soll auf zwei parallelen Installationen betrieben werden“, so kommt man vielleicht zuerst zur Aussage „Der WebShop soll 24 Stunden am Tag verfügbar sein“. Auch das ist eine gewünschte Eigenschaft des WebShops und noch immer nicht ein Ziel, das mit Hilfe des WebShops erreicht werden soll. Erst eine weitere Frage nach dem Warum fördert zutage, dass die Versicherung folgendes Ziel hat: „Kunden sollen auf der ganzen Welt und somit zu jeder Tages- und Nachtzeit die Möglichkeit haben, Versicherungen abschließen.“

Der Painpoint „Die Verkäufer brauchen zu viel Zeit, um Standard-Policen für Kunden auszufertigen“ beschreibt einen unerwünschten Zustand. Das dazugehörige Ziel, also der *erwünschte Zustand*, ist: „Die Verkäufer sind von Routineaufträgen entlastet (und haben mehr Zeit für die Beratung der Kunden).“

**Rahmenbedingungen nicht vergessen**

Ob man nun direkt nach Zielen fragt oder nach Painpoints sucht, in beiden Fällen

besteht die Gefahr, wichtige Aspekte aus den Augen zu verlieren. Das Konzept des Ziels als ein „veränderter und angestrebter Zustand“ in der Zukunft birgt nämlich ein Problem in sich: Der Fokus liegt auf der gewünschten Veränderung. Neben angestrebten Veränderungen gibt es aber immer Aspekte des Zustands, die sich gerade *nicht verändern*, sondern *erhalten* bleiben sollen. Diese Rahmenbedingungen werden aber meist nicht explizit gemacht, was dazu führt, dass sie in der Lösungsfindung zu wenig beachtet werden. Die Lösung bringt dann gegenüber den formulierten Zielen zwar Verbesserungen, schafft aber gleichzeitig neue Painpoints, also Abweichungen vom gewünschten Zustand in Aspekten, die man nicht beachtet hat. Die Formulierung von Rahmenbedingungen, also wichtigen Zuständen, die auch nach der neuen Lösung immer noch bestehen sollen, kann das Entstehen neuer Painpoints wenn nicht verhindern so doch wenigstens einschränken.

**Beispiel WebShop**

Im Kontext des Systems WebShop gibt es die Rahmenbedingung „unveränderter Anteil der Rücktritte von Vertragsabschluss“. Wird diese Rahmenbedingung nicht beachtet, so läuft man Gefahr, dass ein neuer Painpoint entsteht: „höherer Anteil von Vertragsrücktritten nach dem Abschluss über den WebShop“.

**Zielkonflikte**

Innerhalb einer Organisation gibt es – selbst wenn sie eine klare Zielhierarchie hat – immer Teilorganisationen und Individuen, die ihre eigenen Ziele haben. Die Ziele des Unternehmens sind nicht die gleichen wie die der Abteilung und der Abteilungsleiter hat wiederum seine eigenen persönlichen Ziele. Diese verschiedenen Ziele müssen zwar nicht miteinander in Konflikt stehen, aber oft tun sie es. Die Maßnahmen zur Erreichung eines Zieles können die Erreichung eines anderen Zieles beeinträchtigen. Sich in diesem Wirrwarr von Zielen zu

rechtzufinden, ist eine der Herausforderungen des Requirements-Engineers. Nicht erkannte Zielkonflikte führen dazu, dass nicht alle Ziele erreicht werden können. Das Vorhaben ist damit von Anfang an zum (teilweisen) Scheitern verurteilt. Ein erster Schritt ist daher immer, die verschiedenen expliziten und impliziten Ziele zu erkennen und zu dokumentieren. Nur so lassen sich Zielkonflikte erkennen und lösen.

**Beispiel WebShop**

Zu den Zielen der Versicherung gehört vielleicht auch, dass die betrieblichen Kosten gesenkt werden müssen. Die Erreichung dieses Ziels wird, zumindest kurzfristig, durch das neue System WebShop gefährdet. Es wird aber erwartet, dass die Kosten der Versicherung langfristig gesehen durch den WebShop gesenkt werden können. Im Kontext des Systems WebShop gibt es außerdem nicht nur die offiziellen Ziele der Versicherung, sondern auch private Ziele von Stakeholdern. So hat der Verkaufsleiter das zusätzliche Ziel „einen Verkäufer einsparen“.

Demgegenüber haben die Verkäufer folgende Ziele: „Arbeitsstelle behalten“ und „Provisionsniveau halten“. Diese Ziele stehen teilweise im Konflikt zueinander.

**Lösungsideen: die Wie-Frage**

Ziele helfen nicht nur bei der Validierung des fertigen Systems, sie sind auch, wie gezeigt wurde, bei der Spezifikation Quelle und Verankerung der Anforderungen. Eine Quelle sind sie, weil aus den Zielen durch initiale Lösungsideen Anforderungen entstehen, und sie sind Verankerung, weil die Ziele die Antwort geben auf die Frage nach dem Warum einer Anforderung.

Die Überlegung, wie ein System beschaffen sein soll, um in seinem Kontext einen bestimmten Effekt zu erzielen, sollte ein klassischer Produktmanager anstellen. Diese resultiert dann in konkreten Anforderungen. In diesem Sinne sind Anforderungen *nie lösungsneutral*. Zwischen der Identifikati-

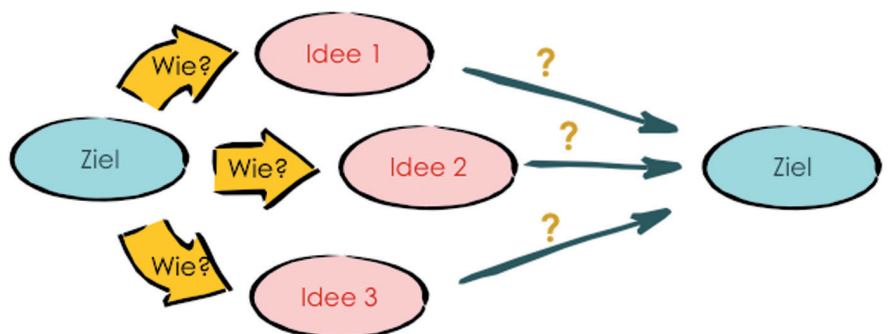


Abb. 4: Funktioniert die Lösungsidee?

1. Separiere die mit einer Lösung anzustrebenden Effekte (Ziele) von den dazu erforderlichen Eigenschaften der Lösung (Anforderungen).
2. Identifiziere wichtige Rahmenbedingungen, die durch die Einführung einer neuen Lösung beibehalten werden müssen.
3. Setze die Anforderungen und Ziele zueinander in Beziehung (Nachvollziehbarkeit/Traceability).
4. Prüfe, ob die realisierte Lösung die geforderten Anforderungen erfüllt (konventionelles Testen, Verifikation).
5. Prüfe, ob die produktiv gesetzte Lösung die gewünschten Ziele auch tatsächlich erreicht (Validierung, gegebenenfalls lange nach dem konventionellen Testen).

**Kasten 2: Fünf Punkte für die Praxis.**

on von Zielen und der Formulierung von Anforderungen liegt vielmehr eine erste Grobidentifikation einer (oder mehrerer) Lösungsideen, an die dann die Anforderungen (deren gewünschte Eigenschaften) formuliert werden (siehe **Abbildung 4**).

Ob diese Lösungsidee funktioniert, also zielführend ist, kann zu diesem frühen Zeitpunkt meist nicht überprüft werden. Umso wichtiger ist es, alle möglichen Informationsquellen wie erfahrene Spezialisten oder die Endbenutzer mit einzubeziehen.

Der Requirements-Engineer kommt in der Regel erst nach diesen Überlegungen ins Spiel, für ihn ist dann der Produktmanager mit seiner Lösungsidee eine wichtige

## Literatur & Links

[OMG15] Object Management Group (OMG), Business Motivation Model (Version 1.3), formal/2015-05-19, 2015  
 [Wei14] T. Weikiens, Systems Engineering mit SysML/UML, dpunkt.verlag, 2014  
 [Wik] Wikipedia, Ziel, siehe: [de.wikipedia.org/wiki/Ziel](http://de.wikipedia.org/wiki/Ziel)

Quelle der Anforderungen. Kennt der Requirements-Engineer die zugrunde liegenden Ziele, hat er immerhin die Chance, die Lösungsideen dahingehend zu hinterfragen, ob sie zielführend sind. Kennt er die Ziele nicht, sollte er versuchen, sie durch Hinterfragen der Anforderungen herauszufinden. Für eine bestimmte Menge von Zielen gibt es nicht nur mehrere passende Lösungsideen, diese Lösungsideen können auch auf unterschiedliche Art zu Anforderungen ausgestaltet werden. Um diese Varianten zu beurteilen, ist nicht nur der erwartete Zielerreichungsgrad relevant (die *Qualität*), sondern auch die mit der Variante verbundenen Risiken und Kosten (neue zu erwartende Painpoints).

### Beispiel WebShop

Die Lösungsidee eines WebShops für das Ziel, zusätzlichen Umsatz zu machen, stand in Konkurrenz zu anderen Lösungsideen wie der, das Werbebudget zu erhöhen. Letztlich hat der Verkaufsleiter nach Rücksprache mit seinen Verkäufern entschieden, dass der WebShop die beste Lösungsidee ist.

### Fazit

Für jeden erfolgreichen Requirements-Engineer gilt: Die Ziele nie aus den Augen

verlieren. Anforderungen sind keine Ziele, vielmehr sollen sie auf den zu erreichenden Zielen basieren. Wer also seine Anforderung auf Zielen baut, hat gute Chancen, am Ende nicht nur ein *richtig gebautes*, sondern auch das *richtige System* zu erhalten. **Kasten 2** fasst die wichtigsten Punkte zusammen, wie dies in der Praxis umgesetzt werden kann. ||

## Der Autor



|| Patrick Grässle  
 (patrick.graessle@knowgravity.com)  
 ist Mitbegründer von KnowGravity Inc., einem Beratungsunternehmen in Zürich, das sich auf modellbasiertes Engineering spezialisiert hat. Er unterstützt seine Kunden als Requirements Engineer und ist Koautor zweier Bücher zu den Themen UML und Geschäftsregeln.