



□ Dr. Kim Lauenroth

(kim.lauenroth@adesso.de)

ist Competence Center Leiter und Chief Requirements Engineer bei der adesso AG. Nach dem Studium der Informatik, BWL und Psychologie an der Universität Dortmund zog es ihn 2003 an die Universität Duisburg-Essen, dort promovierte er 2009 im Bereich Requirements Engineering. Außerdem engagiert er sich im Vorstand des IREB e.V. für die Aus- und Weiterbildung im Requirements Engineering.

## Die Legende von der lösungsneutralen Anforderung oder warum das Requirements Engineering eine Gestaltungsdisziplin ist!

Wenn jemand das Gebiet des Requirements Engineering (RE) betritt, dann lernt er anfangs fast immer etwas über die Trennung zwischen Problem und Lösung (oft auch als Trennung zwischen „Was?“ und „Wie?“ bezeichnet). Sie kann mit Fug und Recht als Klassiker in der RE-Literatur (vgl. z. B. [Poh08]) bezeichnet werden und gehört zu den wesentlichen Gedanken im RE. Anschließend lernt man die Anforderung kennen. Die Anforderung ist so etwas wie der Werkstoff des Requirements Engineer. Er erhebt die Anforderungen, dokumentiert sie, verwaltet sie und validiert sie. Ab und an muss er sie auch abstimmen. Kurzum im RE dreht sich alles um Anforderungen. Zwischendrin kommt aber noch ein weiterer Satz, den man als angehender Requirements Engineer oft lernen muss: „Anforderungen müssen lösungsneutral sein!“, manchmal auch so formuliert: „Die Anforderungen dürfen die Lösung nicht vorwegnehmen!“. Aus dem bisher Gesagten kann dann also geschlossen werden, dass ich mich als Requirements Engineer mit Anforderungen befasse, die die Lösungen nicht vorwegnehmen dürfen, welche man ja auch noch von Problemen trennen soll. Liebe Leserin, lieber Leser, entgegen der üblichen Konventionen in Fachartikeln spreche ich Sie immer mal wieder direkt an und werde auch über meine eigenen Gedanken schreiben. Ich hoffe, Sie sehen mir diesen Stilbruch nach, er soll darstellen, dass die Inhalte dieses Artikels meine persönliche Meinung sind, und dazu dienen, den Artikel etwas lesbarer und verdaulicher zu machen. Mal ganz unter uns ... mich verwirrt das gerade Geschriebene über Probleme, Anforderungen und Lösungen ganz schön und ich glaube auch, dass diese Sicht auf Anforderungen dem RE als Disziplin nicht gut tut. Daher würde ich Ihnen gerne eine Alternative anbieten, die zumindest in meinem Kopf etwas mehr Ordnung erzeugt und nach meiner Erfahrung auch besser funktioniert.

### Was ist eine Anforderung?

Fangen wir also noch einmal von vorn an. Die Literatur (z. B. [Gli14]) sagt uns, dass *eine Anforderung eine Bedingung oder Fähigkeit ist, die von einem Benutzer zur Lösung eines Problems oder zur Erreichung eines Ziels benötigt wird.*

Diese Definition mag auf den ersten Blick harmlos klingen, allerdings lassen sich aus dieser Definition einige fundamentale Festlegungen in Bezug auf die Begriffe Problem, Anforderung und Lösung ableiten:

- Anforderungen (Bedingungen und Fähigkeiten) sind Aussagen, die sich auf

einen (nicht notwendigerweise materiellen) Gegenstand (die Lösung) beziehen.

- Eine Anforderung wird benötigt, um ein Problem zu lösen oder ein Ziel zu erreichen: Damit geht die Formulierung des Problems der Formulierung einer Anforderung voraus.
- Zur Formulierung einer Anforderung sind drei Konzepte notwendig: das Problem, die Lösung (der Gegenstand dessen Eigenschaft beschrieben wird) und die eigentliche Bedingung oder Fähigkeit.

Die letzte Festlegung ist der Knackpunkt. Warum ist eine Lösung notwendig, um eine Anforderung zu formulieren? Sollte es nicht eher umgekehrt sein: Die Lösung wird auf Basis der Anforderungen entwickelt? Der Satz von oben über „lösungsneutrale Anforderungen“ steht ja auch noch im Raum. Wenn die letzte Festlegung stimmt, dann würde er ja die Lösungsneutralität von Anforderungen fundamental in Frage stellen.

Und genau dies ist mit der Feststellung passiert. Vor der Formulierung der Anforderung müssen wir uns für eine Lösung

des betrachteten Problems entscheiden, da wir die Eigenschaften der Lösung in Form von Anforderungen beschreiben.

### Ein Beispiel

So abstrakt könnte man die gerade getroffene Feststellung noch als graue Theorie abtun und so weiter machen wie bisher. Ich möchte nun aber ein sehr einfaches Beispiel vorlegen, das meine Feststellung von gerade ziemlich deutlich macht.

Nehmen wir an, dass unser Problem darin besteht, den Fahrer eines PKW vor zu geringem Luftdruck in den Reifen seines PKWs zu warnen. Die Lösung dieses Problems ist naheliegend. Wir messen regelmäßig den Reifendruck und sobald der Druck eine definierte Schwelle unterschreitet, wird der Fahrer gewarnt. Daraus ergeben sich eine Reihe von Anforderungen, die beispielhaft so aussehen könnten:

- Der Sensor muss den Druck im Bereich zwischen 0 und 5 Bar mit einer Genauigkeit von  $\pm x\%$  messen.
- Der Reifendruck muss alle  $y$ -Sekunden gemessen werden.
- Zu geringer Reifendruck liegt vor, wenn der gemessene Druck unter den Grenzwert  $z$  sinkt.

Schaut man sich die Anforderungen genau an, so ist zu erkennen, dass Druckmessung bereits in den Anforderungen enthalten ist. Dies würde bedeuten, dass zur Formulierung der Anforderungen Wissen über die Gestalt der Lösung notwendig ist oder anders formuliert: Die Anforderungen hängen von der Gestalt der Lösung (hier Drucksensoren) ab.

Um diese Erkenntnis zu überprüfen, wechseln wir versuchsweise das Lösungskonzept in unserem Reifendruckbeispiel. Eine andere Lösung? Kann man denn den Druckverlust überhaupt erkennen, ohne den Luftdruck direkt mit einem Sensor zu messen? Man kann!

Anstatt eines Drucksensors kann die Tatsache genutzt werden, dass ein Reifen, der Luft verliert, auch gleichzeitig seinen Durchmesser verringert. Dies hat zur Folge, dass ein Reifen mit geringerem Luftdruck im Vergleich zu den anderen am PKW montierten Reifen mehr Umdrehungen vollzieht. Genau diese Veränderung kann bspw. durch ESP-Sensorik gemessen werden (vgl. [REI]).

Ersetzen wir also den *Drucksensor* durch *Auswertung der Reifendrehzahl* und formulieren erneut Anforderungen:

- Das ESP-System muss kontinuierlich die Reifendrehzahl aller vier Räder überwachen.
- Die Reifendrehzahl muss mit einer Genauigkeit von  $\pm x\%$  gemessen werden.
- Zu geringer Reifendruck liegt vor, wenn über einen Zeitraum von  $y$ -Sekunden eine Abweichung in der Drehzahl eines Reifens zu den anderen Reifen von  $z\%$  vorliegt.

Das Lösungskonzept und die daraus resultierenden Anforderungen unterscheiden sich substantiell von den Anforderungen für das Lösungskonzept „Drucksensor“.

### Drei Denkkategorien

Aus der zuvor beschriebenen Untersuchung des Wortes „Anforderung“ können wir drei Denkkategorien ableiten:

- Das *Problem* beschreibt, was in der Welt verändert oder erreicht werden soll. Häufig wird für das Wort „Problem“ auch das Wort „Ziel“ oder „Zweck“ verwendet.
- Die *Anforderung(en)* beschreiben notwendige Eigenschaften eines Systems zur Lösung eines Problems.
- Die *Lösung* beschreibt das System, mit dem das Problem gelöst, bzw. das Ziel erreicht oder der Zweck realisiert wird.

Zur Vereinfachung bezeichnen wir diese drei Denkkategorien im Folgenden mit PAL (für Problem, Anforderung, Lösung). Betrachtet man PAL genauer, so erkennt man besondere Eigenschaften:

- Probleme sind unabhängig und hängen damit nicht von Anforderungen oder Lösungen ab. Dies zeigt sich im betrachteten Beispiel daran, dass das Problem durch verschiedenste Lösungskonzepte gelöst werden kann. Der Klassiker der RE-Literatur zur Trennung von Problem und Lösung hat also weiterhin Bestand.
- Die Wahl des Lösungskonzeptes bestimmt die Anforderungen, d. h. ändert man das Lösungskonzept, so ändern sich auch die Anforderungen. Oben haben wir das am Beispiel der Lösungskonzepte „Drucksensor“ und „Auswertung der Reifendrehzahl“ gezeigt. Anforderungen sind damit nicht lösungsneutral.
- Anforderungen sind Beziehungen zwischen Problem und Lösung. Anforderungen beschreiben die notwendigen Eigen-

schaften, die ein System aufweisen muss, um die Lösung für ein Problem zu sein. Damit stellen Anforderungen eine Beziehung zwischen Problem und Lösung her. Oder anders herum, Anforderungen können erst formuliert werden, wenn Problem und Lösung definiert wurden.

In der Buchreihe „Per Anhalter durch die Galaxis“ von Douglas Adams bezeichnet die Abkürzung PAL den Begriff „Problem Anderer Leute“. Eigentlich auch ganz passend, da sich das RE typischerweise mit den Problemen anderer Leute befasst und auch das Ergebnis (die Anforderungen) das Problem anderer Leute ist.

### Konsequenzen für das Selbstbild

Unsere bisherigen Erkenntnisse in Bezug auf PAL scheinen auf den ersten Blick wieder akademischer Natur zu sein und nicht unbedingt von praktischem Wert. Genau so könnte man jetzt sagen, dass die Legende von den lösungsneutralen Anforderungen auch keine besondere Bedeutung hat. Meine persönliche Erfahrung an Hochschulen und in der Industrie sagt mir aber etwas ganz anderes.

Requirements Engineering wird als Disziplin und als Tätigkeit maßlos unterschätzt. Nicht selten höre ich Aussagen wie „Ihr macht doch nur Papier schwarz, das niemand lesen will“. Auf viele Anforderungsdokumente mag dies auch tatsächlich zutreffen, da ihr Inhalt oft keinen Mehrwert für den Leser oder das gesamte Projekt bringt. Ich bin aber fest davon überzeugt, dass dies nicht immer an der mangelnden Kompetenz der handelnden Personen liegt.

Es liegt vielleicht auch daran, dass die Aufgabe des Requirements Engineering durch Aussagen wie „Anforderungen müssen lösungsneutral sein“ derart verunstaltet wurde, dass die handelnden Personen vielleicht nicht anders konnten. Nach meiner Erfahrung führt der Versuch „lösungsneutrale“ Anforderungen zu formulieren unter anderem dazu, dass „wachweiche Aussagen“ in einem Anforderungsdokument formuliert werden, die weder umsetzbar noch testbar sind oder überhaupt einen Mehrwert bieten.

Mir ist klar, dass ich gerade maßlos überzeichnet habe. Aber ich hoffe, Sie teilen meine Wahrnehmung in Ansätzen. Die Konsequenz ist in jedem Fall, dass das Requirements Engineering nicht als Gestaltungsdisziplin in der Softwaretechnik wahrgenommen wird. Die Tätigkeit eines

RElers wird oft als langweilige Verwaltungstätigkeit abgetan, da man das RE viel mehr mit seinen Artefakten assoziiert (den dicken langweiligen Dokumenten oder den Tools voller Anforderungen) als der Tätigkeit und der Bedeutung der Artefakte.

## RE ist Gestaltung

Die wichtigste Schlussfolgerung aus den Ausführungen zu PAL ist, dass ich es in meiner Tätigkeit als Requirements Engineer mit Problemen, Anforderungen und Lösungen zu tun habe. Insbesondere bin ich fest davon überzeugt, dass ich im Requirements Engineering eine gestaltende Tätigkeit vollziehe, bei der die zentralen Weichen für die Gestaltung einer Software gestellt werden, was ja auch zahlreiche Studien belegen.

Mit Gestaltung meine ich explizit nicht die softwaretechnische Gestaltung (oft auch als Software-Design oder -Architektur bezeichnet), ich meine die Gestaltung der Software oberhalb der Technologie. Beispielsweise beschreiben wir in Anforderungsdokumenten durch Use Cases die spätere Benutzung der Software durch den Endnutzer und gestalten dadurch die Arbeit des Endnutzers mit der Software.

Die Disziplin heißt nun mal Requirements Engineering und trägt damit das Wort „Anforderung“ im Titel. Die Anforderungen sind mitnichten das einzige „Material“, das ein Requirements Engineer bearbeitet. Die Ausführungen in diesem Artikel zu PAL machen es für mich sehr deutlich. Erst müssen das Problem und die Lösung definiert werden, erst dann kann eine echte Anforderung formuliert werden. Somit bin ich vor der Formulierung von Anforderungen gezwungen, mich mit Problemen und Lösungen auseinanderzusetzen.

Dieser Veränderung im Verständnis des Begriffs Anforderung ist für mich der Kern einer geänderten Sichtweise auf das Requirements Engineering, weg von einer eher verwaltenden Disziplin hin zu einer gestaltenden Disziplin. Die Bedeutung der Trennung von Problem und Lösung ist im Übrigen nicht nur in der Softwareentwicklung von Bedeutung, sie wird auch intensiv in der Design-Literatur diskutiert.

[Pye78] beleuchtet dieses Thema und gibt eine spannende Einführung in die Design-Theorie. Das Buch macht insbesondere sehr schön deutlich, dass das Problem durchdrungen werden muss, um eine gute Lösung zu gestalten.

Das Requirements Engineering als ge-

staltende Disziplin ist aus meiner Sicht dafür verantwortlich, dass mit allen relevanten Stakeholdern

- die Problemstellungen, die eine Software lösen soll, definiert und abgestimmt werden,
- Lösungen für die identifizierten Problemstellungen definiert und abgestimmt werden,
- Anforderungen an die Lösungen definiert und abgestimmt werden.

Das explizite Auseinandersetzen mit Problemen, Anforderungen und Lösungen im Requirements Engineering ist keine großartige Veränderung – sämtliche Tätigkeiten, Methoden und Techniken bleiben unverändert. Probleme und Lösungen werden einfach mit Anforderungen auf eine Stufe gestellt und als explizite und gleichwertige Einheiten behandelt und bearbeitet. Vorschläge, wie dies gelingen kann, finden Sie beispielsweise in [Lau13] und [Lau14].

Die Wirkung dieser Veränderung ist aus meiner Sicht jedoch fundamental. Ich setze mich explizit mit Problemen und ihren Lösungen auseinander und habe somit die Chance, verschiedene Lösungsansätze zu diskutieren und dann bewusst eine – möglicherweise sogar innovative – Lösung auszuwählen.

Ein letzter Gedanke ist mir an dieser Stelle noch wichtig. Dieser Artikel ist kein Plädoyer für ausufernde Requirements Engineering-Projekte oder gar für die von den Vertretern der agilen Bewegung vielfach bemühten Waterfall-Projekte. Mir geht es einzig und allein darum, dass das Requirements Engineering als das wahrgenommen wird, was es ist: eine Gestaltungsdisziplin mit einer großen Verantwortung für das Endergebnis eines Softwareprojektes. Wer, wann, wie und wie viel Requirements Engineering betreibt, ist eine ganz andere Frage.

## Requirements Engineering mit Industriedesign gleichsetzen

Man könnte es auch so ausdrücken: wir schreiben das Requirements in RE etwas kleiner und das Engineering etwas größer. Denn Engineering bedeutet im Grunde genommen nichts anderes als das „Erfinden“ oder „Gestalten“.

Wichtig ist aber, dass sich das Engineering nicht auf die Requirements bezieht. Die Requirements sind also nicht der Gegenstand der Gestaltung, die Requirements sind neben der Beschreibung von

Problemen und Lösungen ein *Mittel zur Gestaltung einer Software*.

Andere Branchen haben da für mich Vorbildcharakter. Sie entdecken, wie wichtig gutes Design (oder Industriedesign im Bereich der Massenprodukte) für die Entwicklung erfolgreicher Produkte ist. Daniel H. Pink fasst aus meiner Sicht diese Entwicklung hin zu mehr Design und insbesondere auch die Bedeutung von Design für erfolgreiche Produkte sehr gut zusammen (siehe [Pin05]).

Genauso wie Konzepte, Skizzen und Prototypen im Industriedesign kein Selbstzweck sind, sondern ein wichtiges Mittel zur Gestaltung eines Produktes, sollten Anforderungsdokumente und ihre verschiedensten Stilrichtungen nicht als Selbstzweck, sondern als ein Mittel zur Gestaltung einer Software angesehen werden. Auf eine prägnante Formel reduziert: *Requirements Engineering ist das Industriedesign der Softwareentwicklung* und sollte auch mit einem entsprechenden Selbstverständnis und Selbstvertrauen betrieben werden. ■

## Literatur & Links

[Glin14] M. Glinz, A Glossary of Requirements Engineering Terminology, Version 1.6, International Requirements Engineering Board, 2014.

[Lau13] K. Lauenroth, Starten Sie mit einem Blatt Papier: Anforderungen mit dem PAL-Modell strukturieren und analysieren, OBJEKTSpektrum, Nr. 3, 2013.

[Lau14] K. Lauenroth, What does it mean to say „requirement“? An inquiry into the abilities of the human mind and the meaning of the word „requirement“. RE-Magazine No. 1, 2014, <http://re-magazine.ireb.org/issues/2014-1-learning-to-fly/what-does-it-mean-to-say-requirement/>

[Pin05] D. H. Pink, A whole new mind – why right-brainers will rule the future. Riverhead Books, 2005.

[Poh08] K. Pohl, Requirements Engineering – Grundlagen, Prinzipien, Techniken. dpunkt.verlag, 2008.

[Pye78] D. Pye, The Nature & Aesthetics of Design – A Design Handbook, the Herbert Press, 1978.

[REI] Reifendruckkontrollsystem, siehe <http://de.wikipedia.org/wiki/Reifendruckkontrollsystem>