



Maik Pfingsten

# Erfolgreich Lastenhefte schreiben

Eine Schritt für Schritt  
Anleitung aus der Praxis  
eines Troubleshooters



Systems Engineering Akademie

# Erfolgreich Lastenhefte schreiben

---

Eine Schritt für Schritt Anleitung  
aus der Praxis eines Troubleshooters

Geschrieben von Maik Pfingsten



## **Erfolgreich Lastenhefte schreiben**

Erste Edition

Veröffentlichungsdatum August 2014

geschrieben von Maik Pfingsten

([www.systemsengineeringakademie.de](http://www.systemsengineeringakademie.de))

Produzent: Jumoliev Press

Graphic Design: Jumoliev Press

©Copyright 2014 blueshaper UG (haftungsbeschränkt)

### Wichtiger Hinweis an die Benutzer

Lesern dieses Handbuchs wird empfohlen, ihre eigene Sorgfalt walten zu lassen, wenn es zu geschäftlichen Entscheidungen kommt. Alle Vorgehensweisen, Informationen, Produkte oder Dienstleistungen, die mitgeliefert werden, müssen unabhängig durch Ihre eigenen qualifizierten Spezialisten bewertet werden. Obwohl ich bei der Vorbereitung dieses E-Books entsprechend sorgfältig vorgegangen bin, besteht keine Verpflichtung gegenüber Personen oder Unternehmen in Bezug auf Verlust oder Schaden, die direkt oder indirekt durch den Einsatz der Methode oder der Informationen aus diesem E-Book entstehen

können. Mit dem Lesen dieses E-Books stimmen Sie zu, dass weder ich noch mein Unternehmen für Erfolg oder Misserfolg Ihrer Entscheidungen aufgrund der hier veröffentlichten Informationen verantwortlich sind.

Copyright - Sie können gerne alles kopieren, aber ...

... nur für Ihren persönlichen Zweck als Spezialist (also nicht z.B. für gewerbliche Zwecke als Trainer). Die im E-Book beschriebene Vorgehensweise ist und bleibt geistiges Eigentum von blueshaper und steht jenen Personen zur Verfügung, die das Buch gekauft haben oder Teilnehmer des Trainings sind. Nicht erlaubt ist der Auszug von Kopien, sowie die Verwendung von Bestandteilen dieses E-Books für eigene kommerzielle Verwendung.

## Marken und Drittanbieter

Der Inhalt in diesem Handbuch enthält möglicherweise Informationen, Produkte oder Dienstleistungen von Drittanbietern. Einige Links sind unter Umständen Affiliatelinks. Inhalte von Drittanbietern schließen Produkte und Meinungen ein, die durch ihren Besitzer angeführt werden. Das bedeutet: Ich übernehme keine Verantwortung oder Haftung für Inhalte oder

Meinungen von Drittanbietern. Das Veröffentlichen der Inhalte von Drittanbietern begründet keine Garantie bezüglich Informationen, Anweisungen, Meinungen, Produkten oder Dienstleistungen, die diese beinhalten. Der Einsatz von empfohlenen Inhalten der Drittanbieter garantiert keinen Erfolg oder Gewinn bezogen auf Ihr Geschäft. Die Veröffentlichung solcher Inhalte von Drittanbietern ist einfach eine Empfehlung und damit Ausdruck meiner eigenen Meinung zu diesem Inhalt.

# Über den Autor

*„Was Ingenieure antreibt, ist, mit Stolz und die Leidenschaft etwas Sinnvolles zu schaffen.“*

Maik Pfingsten

**Maik Pfingsten** ist internationaler Speaker, Podcaster, Dozent und Mentor mit dem Schwerpunkt Systems Engineering. Mit einem Praxishintergrund von über 13 Jahren als Systemingenieur und Troubleshooter in der Automobilentwicklung hat er sich zu einem inspirierenden Vordenker im Systems Engineering entwickelt. Unter Berücksichtigung pragmatischer Lösungsansätze ist sein Ziel, einen nachhaltigen Nutzen zu erschaffen. Entscheider in unterschiedlichen Industriezweigen nutzen seine Denkansätze, um einen Wettbewerbsvorteil durch den Einsatz von Systems Engineering zu erlangen. Unternehmen, die seine Betrachtungsweisen nutzen, sind Zeiss, Fresenius Medical Care, B.Braun, HILTI, Valeo, Kostal, Daimler, ThyssenKrupp, T-Systems, Hella, JohnsonControl und viele mittelständische Technologieunternehmen.

Neben seinen Vortragsengagements fokussiert er sich auf die Weitergabe von Methodenwissen rund ums Systems Engineering. Durch seine Zeit als externer Troubleshooter hat er für verschiedenste Unternehmen und Industriezweige gearbeitet. Er ist zertifizierter Trainer und gibt als Dozent für Systems Engineering sein Wissen in internationalen Masterprogrammen an die nächste Ingenieurgeneration weiter.

# Inhalt

<b>ÜBER DEN AUTOR .....</b>	<b>6</b>
<b>INHALT .....</b>	<b>8</b>
<b>VORWORT .....</b>	<b>11</b>
<b>HINTERGRUNDWISSEN .....</b>	<b>13</b>
<b>KAPITEL 1: DIE VORBEREITUNGEN TREFFEN.....</b>	<b>18</b>
<b>DAS VORGEHEN PLANEN .....</b>	<b>19</b>
<b>ÜBERSICHT ERLANGEN .....</b>	<b>29</b>
<b>KAPITEL 2: DIE ANFORDERUNGEN ERHEBEN .....</b>	<b>35</b>
<b>ANFORDERUNGEN ANALYSIEREN .....</b>	<b>36</b>
<b>SYSTEMABGRENZUNG ERZEUGEN .....</b>	<b>38</b>
<b>SYSTEM FOOTPRINT ERSTELLEN.....</b>	<b>43</b>
<b>FEHLENDE DOKUMENTE BESORGEN .....</b>	<b>56</b>
<b>OFFENE FRAGEN KLÄREN .....</b>	<b>57</b>
<b>SYSTEM FOOTPRINT AKTUALISIEREN .....</b>	<b>59</b>
<b>KAPITEL 3: DAS LASTENHEFT ERSTELLEN.....</b>	<b>60</b>
<b>DAS LASTENHEFT SCHREIBEN .....</b>	<b>61</b>
<b>MIT ATTRIBUTEN ARBEITEN .....</b>	<b>72</b>
<b>KAPITEL 4: LASTENHEFT ÜBERPRÜFEN UND FREIGEBEN .....</b>	<b>75</b>

<b>PEER-REVIEWS DURCHFÜHREN .....</b>	<b>76</b>
<b>FORMALES REVIEW DURCHFÜHREN .....</b>	<b>78</b>
<b>FREIGABE DURCHFÜHREN .....</b>	<b>80</b>
<b>KAPITEL 5: AUS ERFAHRUNG LERNEN .....</b>	<b>84</b>
<b>RETROSPEKTIVE DURCHFÜHREN .....</b>	<b>85</b>
<b>ABSCHLUSS .....</b>	<b>89</b>
<b>WIE GEHT ES WEITER?.....</b>	<b>90</b>

für Julia, Eva, Mona und Linus

# Vorwort

Sie möchten Ihr Lastenheft verbessern, wissen aber nicht, wo Sie beginnen sollen? Vielleicht haben Sie in Ihrem Entwicklungsprojekt festgestellt, dass Sie an Ihrem Lasterhaft noch arbeiten müssen. Aus welchem Grund auch immer, genau dafür ist dieses Buch geschrieben.

Im Februar 2012 begann ich mit meinen Podcast über Systems Engineering. Von Beginn an waren Themen wie Lastenhefte, Pflichtenhefte, Spezifikationen und Systemanforderungen von den Hörern oft nachgefragte Gebiete. Während mehrerer Vorträge fragten mich dann Teilnehmer, wie ich mit der Erfahrung als Troubleshooter in der Automobilentwicklung vorgehen würde, um Lastenhefte zu erstellen.

Bis heute hat sich wenig geändert. Alle Unternehmen wollen für ihre Entwicklungsprojekte gute Lastenhefte, aber viele kommen einfach nicht dazu, sich die Zeit zu nehmen und die wichtigen Schritte zu tun. In diesem Buch geht es nicht darum, theoretisches Wissen zu vermitteln, sondern Ihnen konkrete Schritte an die Hand zu geben, in kürzester Zeit ein Lastenheft zu erstellen.

Aus der Praxis für die Praxis.

## Danke sagen

Dieses Buch wäre nie entstanden, wenn mich nicht ein paar entscheidende Menschen in meinem Umfeld unterstützt hätten. Zunächst möchte ich erst mal meiner Frau und den drei Kindern danken, dass es sie gibt und sie mich bei meinem Treiben als Podcaster und Autor so unterstützen. Dann geht mein Dank an die Hörer-Community des Podcast, ohne deren beharrliche Nachfrage dieses Buch nie entstanden wäre.

# Hintergrundwissen

*Wer die Geometrie begreift, vermag in dieser Welt alles zu verstehen.*

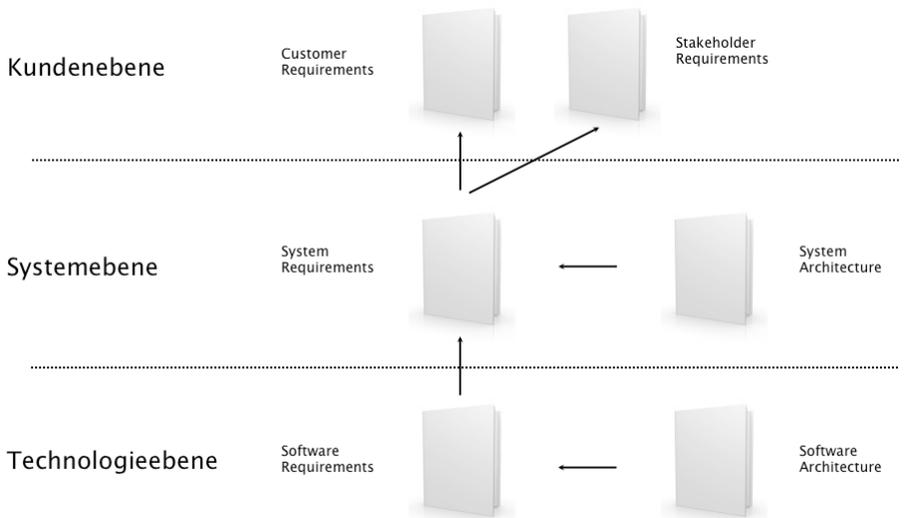
Galileo Galilei

Lastenhefte vs. Pflichtenhefte: Ein Thema, das zur Zeit eine Menge Leute beschäftigt und bei dem wir einige Zusammenhänge besprechen müssen, um später die Umsetzung besser zu verstehen.

Die oberste Ebene ist die Kundenebene. Dort gibt es die sogenannten `Customer Requirements` oder Kundenanforderungen. Das sind beispielsweise ganz klassische Lastenhefte, die sie vom Kunden oder dem Produktmanagement bekommen. Dann gibt es zusätzliche auch noch `Stakeholder requirements`. Das sind Anforderungen von Interessensgruppen, beispielsweise fixiert in internen oder externen Normen. In einer Normenkommission wird festgelegt, dass ein gewisses Thema von den Anforderungen her immer gleich definiert wird.

Diese beiden Artefakte sind die Spezifikation auf der Kundenebene, klassisch die Lastenhefte im allgemeinen Sprachgebrauch.

Dann haben wir bei komplexeren Systemen noch eine direkt darunter liegende Ebene, die Systemebene. Auf diese nächsten Ebene - klassisch ist das immer so im allgemeinen Sprachgebrauch das Pflichtenheft – gibt es im englischen Sprachgebrauch die System Requirements Specification (SRS). Das ist die Spezifikation der Systemanforderungen. Dazu kommt ein zweites Dokument, die System Architecture Specification (SAS), beziehungsweise die Architekturbeschreibung des Systems auf dieser Ebene.



Ebenen der Spezifikationen

...

# Kapitel 1: Die Vorbereitungen treffen

# Das Vorgehen planen

Die Herausforderung in heutigen Entwicklungsprojekten ist die Planung und Steuerung von Aufgaben. Da wir oft während der Arbeit in den Projekten gestört werden und auch häufig spontan mit unerwarteten Zusatzaufgaben überschüttet werden, brauchen wir ein System, mit dem wir unsere Arbeit und unseren Durchsatz besser planen können. Wenn wir dabei auch noch unsere aktuelle Situation für uns und andere visualisieren können und darauf basierend vernünftige Entscheidungen fällen, haben wir viel gewonnen.

Kanban ist eine sehr wirkungsvolle Methode, um unsere Tätigkeiten zur Erstellung eines Lastenhefts zu planen. Ich habe das erste Mal Ende 2010 in einem Troubleshootingprojekt Kanban eingesetzt, um für mich und das Management den Fortschritt zu dokumentieren.

Mein Problem war, dass ich als Troubleshooter die Systemanforderungen und die Systemarchitektur dokumentieren musste. Gleichzeitig war ich auch in diverse Projektmeetings eingebunden. Das führte dazu, dass ich nicht am Platz war, wenn mich ein Manager suchte. Irgendwann kam die Frage auf "Herr Pflingsten, wo sind sie eigentlich immer? Wie kann ich sicher

sein, dass Sie voran kommen?" Meine Antwort darauf war mein erstes Kanban-Board.

Diese Methode des Kanban nutze ich seitdem regelmäßig in Projekten und habe damit extrem viel positive Erfahrung gemacht. Die hier beschriebene Vorgehensweise ist das Ergebnis und die Erfahrung eines jahrelangen Einsatzes.

Ich habe mir aus der Konstruktion eine lange Bahn Papier vom Plotter besorgt und auf die Rückseite des Aktenschanks gehängt. Diese habe ich dann in "Input-Queue", "Lastenheft besorgen", "Lastenheft analysieren", "Systemanforderungen beschreiben", "Systemanforderungen reviewen", "Systemarchitektur beschreiben", "Systemarchitektur reviewen" und "Fertig" eingeteilt. Später haben wir darunter ein Kanban-Board für den Software-Requirements-Ingenieur angehängt. In diesem Bild ist mein erstes Kanban-Board während dem Einsatz zu sehen:

...

# Kapitel 2: Die Anforderungen erheben

...

# Systemabgrenzung erzeugen

Ein entscheidender Schritt zur Entwicklung eines Lastenhefts ist die Definition der Systemgrenzen. Ein System existiert niemals alleine, sondern interagiert immer mit anderen Systemen oder Menschen. Die Grenzen zeigen uns, was im System drinnen und draußen ist. Gleichzeitig ermöglicht uns die Systemgrenze auch, die Schnittstellen des Systems zu identifizieren und festzulegen, was über die Systemgrenzen hinweg ausgetauscht wird. Schnittstellen sind der entscheidende Faktor in einem System. Oft werden Schnittstellen nicht richtig identifiziert und dann auch nicht richtig bedient. Das führt häufig zu schweren Systemfehlern, die dann aber oft auch nicht so einfach berichtigt werden können.

Ich nutze bei der Erstellung der Systemabgrenzung fünf Schritte:

1. Definiere, was draußen und was drinnen ist

Entscheidend für das Systemverständnis ist zu wissen, was zum System gehört. Auch wenn viele Entwickler meinen zu wissen, was zum System gehört, ist in den Projekten die klare

Abgrenzung nicht vorhanden.

In diesem Schritt verschaffe ich mir einen Überblick über alle Komponenten und (Teil-) Systeme auf der obersten Ebene. Dazu nutze ich ein Whiteboard und Post-its. Auf dem Whiteboard zeichne ich ein großes Rechteck und klebe die Post-its in das Rechteck, wenn ich diese Komponenten oder Teilsystem als intern definiere. Wenn das System als externes System definiert wird, dann klebe ich es außerhalb des Rechtecks auf.

Als Quellen dienen mir der Systems Footprint und die Notizen aus der Analyse der Lastenhefte. So erhalte ich langsam einen Überblick und kann entscheiden, wie die erste grobe Systemabgrenzung aussehen muss.

## 2. Gruppieren die externen Systeme

Durch den ersten Schritt habe ich nun alle externen Systeme als Post-its außen um das Rechteck aufgeklebt. Nun beginne ich, die Zettel nach einem ersten Schema zu gruppieren. Beispielsweise alle Systeme, die für die Fahrdynamik zuständig sind. Oder alle Systeme, die zur Erweiterung eines Systems hinzugekauft werden können. Wie Sie die Gruppierung durchführen, hängt von Ihrem System ab. Es gibt hier kein Richtig oder Falsch. Bei mir ist oft die zweite oder dritte Version deutlich unterschieden vom ersten

Entwurf. Wichtig ist, dass Sie mit der Gruppierung beginnen und Muster dafür finden.

### 3. Definiere die Schnittstellen auf einer groben Ebene

In diesem Schritt zeige ich die Schnittstellen auf, die zwischen dem eigenen System und den externen Systemen existieren. Dazu nutze ich Boardmaker in drei verschiedenen Farben, um die drei Grundtypen für Systemschnittstellen (Austausch von Information, Energie oder Stoffen) darzustellen. Ich zeichne eine Linie zwischen dem eigenen und dem externen System in der jeweiligen Farbe. Gibt es mehrere unterschiedliche Schnittstellen, dann zeichne ich mehrere unterschiedlich farbige Linien. Am Ende der Linien zeichne ich Pfeilspitzen. Je nach Richtung der Schnittstelle zeichne ich eine oder zwei Pfeilspitzen an die Linie.

### 4. Visualisiere das Ergebnis

Im letzten Schritt geht es darum, das Ergebnis vom Whiteboard zu übertragen und für die nächsten Schritte nutzbar zu machen. Ich übertrage das Ergebnis gerne in ein SysML Schaubild. Wer kein SysML kennt oder kein Werkzeug dafür hat, kann auch hervorragend Powerpoint dafür nutzen. Welches Werkzeug Sie nutzen, ist am Ende des Tages egal. Auf den Inhalt kommt es an.

## 5. Systemabgrenzung reviewen

Reviews sind ein wichtiger Schritt während des Entwicklungsprojekts. Sie sind meiner Erfahrung nach eine Geheimwaffe, mit der früh Fehler oder Verständnisprobleme abgefangen werden können.

Nehmen Sie sich Ihre Systemabgrenzung und suchen Sie sich ein oder mehrere Kollegen, die Ihnen ein Feedback geben können.

Seien Sie nicht überrascht, wenn Ihre späteren Versionen der Systemabgrenzung anders aussehen als die erste Version. Die Rückmeldungen durch die Kollegen sind wichtig und führen bei mir oft zu einer neuen Version, mit denen ich mich Schritt für Schritt einem guten Ergebnis nähere. Ich habe noch nie erlebt, dass meine dritte oder vierte Version der Systemabgrenzung noch genau so aussah, wie mein erster Wurf.

In der Episode #13 [Geheimwaffe Review](#) bin ich auf die Reviews und ihre verschiedenen Formen eingegangen.

## Tipps & Tricks

Diese Aufgabe fordert eine sehr hohe geistige Denkleistung. Ich habe die Erfahrung gemacht, dass es für mich ein paar

Rahmenbedingungen gibt, die meine Arbeit an dieser Aufgabe effektiv unterstützen.

- Ich arbeite alleine an der Systemabgrenzung

Ich arbeite sehr gerne in Teams und mit Menschen zusammen. Aber für die heutige Aufgabe ist es für mich deutlich effektiver, wenn ich mich zunächst alleine durch die Materie denke. In der Aufgabe für den nächsten Tag binde ich dann die Kollegen ein.

- Ich schaffe mir eine ruhige Umgebung

Die Aufgabe ist kreativ und denkintensiv zugleich. Ich schaffe mir immer eine ruhige Umgebung, in der ich ungestört arbeiten kann. Dazu eignet sich in der Regel ein kleines Besprechungszimmer, in dem ein Whiteboard oder ein Flipchart vorhanden ist.

- Ich visualisiere erst auf Papier und dann im Rechner

Für mich ist die kreative Arbeit schneller, wenn ich erst auf Papier, mit Stift und Post-its bewaffnet, arbeite. Dann muss ich mich nicht darum kümmern, ob ich irgendwelche Server-Lizenzen habe, der Virenschoner rattert oder die Software abstürzt. Erst wenn ich meine Gedanken soweit auf Papier gebracht habe, dann übertrage ich das Ergebnis in eine elektronische Form.

# System Footprint erstellen

Bei diesem Schritt geht es darum, das große Bild eines Systems zu erfassen. Aber warum benötigen wir dieses Bild? Wenn wir ein Lastenheft erstellen und darin Anforderungen beschreiben, werden wir irgendwann an den Punkt kommen, wo wir entscheiden müssen, ob eine Anforderung für das System sinnvoll ist oder nicht. Diese Entscheidung können wir einfacher fällen, wenn wir den Kernnutzen des Systems und die wesentlichen Bestandteile im Blick haben. Den Systems Footprint habe ich als Troubleshooter entwickelt, da ich immer sehr schnell verstehen muss, wie die wichtigen Zusammenhänge des System sind.

## Wie funktioniert der System Footprint?

Der System Footprint basiert auf der Idee einer Canvas. Das ist eine große, weiße Fläche, die in verschiedene Bereiche eingeteilt ist. Diese Flächen können später, z.B. im Rahmen eines Workshops, mit Inhalt in Form von Post-its gefüllt werden. Dies sind die beim System Footprint vorhandenen Bereiche:

...

# Kapitel 3: Das Lastenheft erstellen

# Das Lastenheft schreiben

Bei diesem Schritt geht es darum, die Strukturen in dem Lastenheft anzulegen, die wir für unsere nächsten Schritte benötigen. Je nachdem, welches Requirement-Management-Werkzeug genutzt wird, lege ich verschiedene Kapitel an.

Wenn Sie kein Requirement-Management-Tool benutzen, empfiehlt es sich, diese Kapitel in einzelnen Worddokumenten in einem Laufwerk abzubilden. In diesen Kapitel sortiere ich die gefundenen Informationen.

Die folgende Aufgabe ist nicht zu unterschätzen. Wenn Sie merken, dass Sie an dieser Aufgabe länger sitzen, dann empfehle ich Ihnen, zunächst die Grundstruktur anzulegen und grob auszufüllen. Sie können dann über mehrere Durchläufe dieser Vorgehensweise (Sprints) die Strukturen inhaltlich weiter ausfüllen. Meiner Erfahrung nach ist es effektiver, eine zeitliche Grenze zu setzen und sich dann mit dem nächsten Thema zu beschäftigen. In der nächsten Iteration habe ich dann weitere Inhalte zu den einzelnen Kapiteln im Lastenheft hinzugefügt.

Zunächst erstelle ich in dem Requirement-Management-Werkzeug (z.B. Doors ö.ä.) eine eigene Spezifikation (=Modul in Doors) und importiere mein über die Jahre entwickeltes Template für ein Lastenheft. Wenn ich mit einem anderen Werkzeug arbeiten muss, gehe ich ähnlich vor und lege ein Lastenheft an. Wenn ich schon in einem Unternehmen eine Standardvorlage finde, vergleiche ich diese Vorlage mit meinem Template und ergänze mindestens die noch fehlenden Bereiche. Oft gibt es aber keine Vorlagen, oder die Vorlagen sind so allgemein, dass ich mein Template direkt einsetze.

Als nächstes fülle ich die Versionsnummer und den Status des Dokuments aus und erzeuge eine erste Baseline, um einen Startpunkt zu schaffen. Eine Baseline ist vergleichbar mit einer Kopie oder einem Foto der aktuellen Version des Lastenhefts, um einen alten Stand wieder herzustellen oder mit einem neuen Stand zu vergleichen, wie ein Versionsmanagement meiner Lastenhefts. Alle guten Requirement-Management-Werkzeuge bieten diese Möglichkeit.

## Kapitel "Management Summary" erstellen

In diesem Kapitel beschreibe ich für das Management eine kurze Zusammenfassung des Lastenhefts. Der Zweck ist das

Verständnis bei dem Leser zu erzeugen, worum es in diesem Lastenheft geht.

## Kapitel "Einleitung" erstellen

Dieses Kapitel fülle ich mit den einführenden Informationen zu dem Dokument. Das dient dem Verständnis, warum dieses Lastenheft überhaupt existiert.

- Zweck des Dokuments:

Ich beschreibe den Zweck des Lastenheftes. In der Regel ist das die Dokumentation der Anforderungen an das System.

- Ziel und Abgrenzung des Dokuments:

Hier beschreibe ich klar, was in das Dokument gehört und was nicht. Beispielsweise gehören hier keine projektrelevanten Anforderungen hinein, sondern nur systemrelevante Anforderungen. Eine Anforderung, die nach Abschluss des Projektes irrelevant ist, beispielsweise Meilensteine, hat in Lastenheft nichts zu suchen. Dafür gibt es Projekthandbücher.

- Referenzen zu Prozessstandards:

Häufig gibt es in Unternehmen Prozess-Standards auf die ich hier verweise.

- Dokumentenmanagement:

Hier beschreibe ich in welcher Form ich das

...

# Mit Attributen arbeiten

Wenn Sie ein Requirement Management Werkzeug einsetzen, haben Sie die Möglichkeit, die Anforderungen bei der Erstellung weiter zu bewerten.

Ich nutze dazu fünf Attribute. Falls ich zu viele Anforderungen zu bewerten habe, dann nehme ich mir im ersten Durchlauf die wichtigsten Anforderungen und gehe sie durch. Alternativ binde ich zur Bewertung auch die Fachkollegen mit ein. Das geht besonders gut, wenn die Artefakte nach dem fachlichen Schwerpunkt organisiert sind. Beispielsweise kann das Lastenheft für die mechanischen Komponenten auch ein Konstrukteur bewerten.

1. Typ: Mit diesem Attribut lege ich fest, welchen Typ die Anforderung besitzt. Die von mir verwendeten Typen sind:
  - Heading = Identifiziert eine Überschrift,
  - Req = Identifiziert eine Anforderung,
  - ReqInfo = Identifiziert eine Information zu einer Anforderung,

- DocInfo = Identifiziert eine Information zum vorliegenden Dokument.

2. Domäne: Mit diesem Attribut lege ich fest, in welchen fachlichen Bereich die Anforderung fällt. Dabei kann ich auch mehrere Bereiche auswählen. Die Domänen sind:

- Project = Anforderungen, die das Projekt und nicht das System betreffen,
  - System = Anforderungen, die das System als Ganzes betreffen,
  - Mechanic = Anforderungen, die die Mechanik betreffen,
  - Hardware = Anforderungen, die die Hardware betreffen,
  - Software = Anforderungen, die die Software betreffen,
  - Testing = Anforderungen, die den Testbereich betreffen.
- Status: Mit diesem Attribut lege ich fest, in welchem Status die Bearbeitung der Anforderung ist. Die Werte sind:
- n/a = noch nicht begonnen,
  - in review = in der Bewertung,
  - in Klärung = Anforderung wird gerade geklärt,
  - geschlossen = Bewertung abgeschlossen.

...

# Kapitel 4: Lastenheft überprüfen und freigeben

# Peer-Reviews durchführen

Reviews sind ein wichtiger Schritt während des Entwicklungsprojekts. Sie sind meiner Erfahrung nach eine Geheimwaffe, mit der früh Fehler oder Verständnisprobleme abgefangen werden können. Dabei gibt es vier verschiedene Typen von Reviews, die ich einsetze.

Ein Peer Review ist eine direkte Form des Reviews. Ich nutze es in zwei verschiedenen Formen. Entweder gebe ich die Informationen an einen Kollegen mit der Bitte, mir ein Feedback zu geben. Er kann sich dann in Ruhe Zeit nehmen und die Ergebnisse sichten. Anschließend setzen wir uns zusammen und ich spreche mit ihm das Feedback durch.

Mir reichen als Dokumentation auch handschriftliche Notizen auf einem Ausdruck. Alternativ setze ich mich mit dem Kollegen direkt zusammen und wir sprechen die Ergebnisse durch. Dann notiere ich mir die Anmerkungen.

Nehmen Sie sich Ihre Zwischenergebnisse des Lastenhefts und suchen Sie sich ein oder mehrere Kollegen, die Ihnen ein Feedback geben können. Vereinbaren Sie mit den Kollegen das

...

# Kapitel 5: Aus Erfahrung lernen

# Retrospektive durchführen

Es geht hier nicht um den Inhalt des Lastenhefts, sondern es ermöglicht einen Rückblick über die Vorgehensweise bei der Erstellung. Das Ziel der Retrospektive ist es, als Team in einer workshopähnlichen Form zu reflektieren, was gut funktioniert hat und was vielleicht zukünftig besser funktionieren kann.

Das Ziel der Retrospektive ist es, als Team in einer workshopähnlichen Form zu reflektieren, was gut funktioniert hat und was vielleicht zukünftig besser funktionieren kann.

Eine Retrospektive dauert durchschnittlich zwischen zwei und vier Stunden und durchläuft fünf verschiedene Phasen:

## 1. Den Boden bereiten

In der ersten Phase geht es darum, eine gute Umgebung für die Retrospektive zu bereiten. Begrüßen Sie zunächst alle Teilnehmer. Anschließend stimmen Sie die Kommunikationsregeln ab.

Klären Sie beispielsweise, dass alle Mobiltelefone leise oder ausgeschaltet sind, und vereinbaren Sie, wie mit Feedback und Kritik umgegangen wird. Abschließend ist es

...

# Abschluss

...