

- Hilfreich ist es, schon während der Analyse ein **Begriffslexikon** anzulegen; dieses wird während der gesamten Software-Entwicklung verwendet und ergänzt.
- In den frühen Phasen wird dieses Dokument **aufgebaut und weiterentwickelt**.
- Das Begriffslexikon enthält solche Begriffe, die
  - **wichtig** sind und
  - von verschiedenen Leuten, v.a. von Klienten, Analytikern und Entwicklern, **unterschiedlich ausgelegt werden** könnten.
- Dazu gehören häufig **Begriffe, die auf den ersten Blick völlig klar zu sein scheinen**.
- Beispiel: Informationssystem für die Prüfungsdaten von Studenten
  - „Student“, „Prüfung“, „Note“, „Erstprüfung“



- a) Begriff und Synonyma (im Sinne der Spezifikation)
  - b) Bedeutung (Definition, Erklärung)
  - c) Abgrenzung (wo ist dieser Begriff nicht anzuwenden?)
  - d) Gültigkeit (zeitlich, räumlich, sonst)
  - e) Fragen der Bezeichnung, Eindeutigkeit u. A.
  - f) Unklarheiten, die noch nicht beseitigt werden konnten
  - g) verwandte Begriffe (Querverweise)
- Die Angaben werden **aus den Gesprächen und Interviews abgeleitet** oder, wenn sie von den Analytikern kommen, **sorgfältig mit den Klienten überprüft.**
  - Typische Quellen sind im genannten Beispiel die Angestellten in der Verwaltung, die Juristen der Uni, die Gesetze und Ordnungen der Universität.

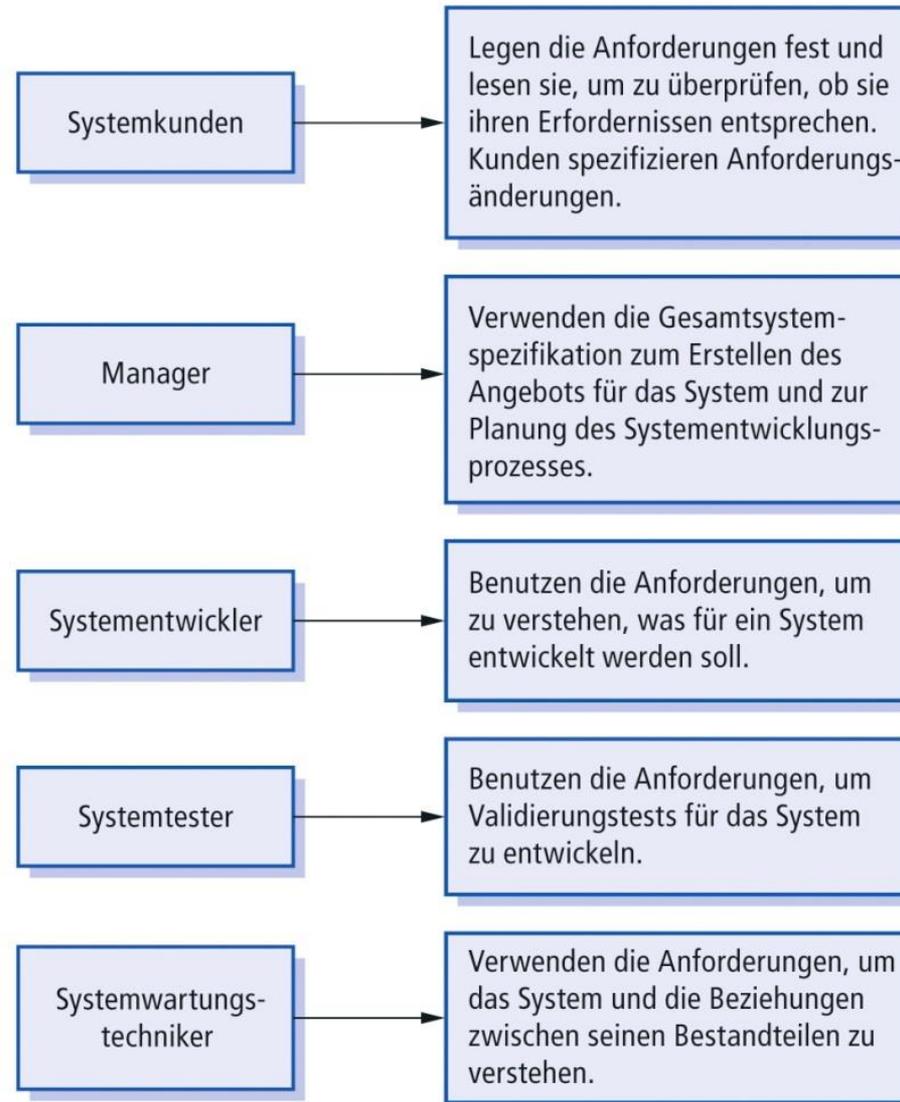
<b>Begriff</b>	<b>Student</b> , synonym Studentin, Studierender, Studierende
<b>Bedeutung</b>	Eine Person, die an der Universität Stuttgart immatrikuliert ist und noch nicht exmatrikuliert wurde, die folglich legal einen Studentenausweis der Universität Bremen hat oder haben könnte.
<b>Abgrenzung</b>	Gasthörer und Studierende anderer Hochschulen sind im Sinne dieses Systems keine Studenten.
<b>Gültigkeit</b>	Mit der Immatrikulation an der Universität Bremen entsteht ein neuer Student; er existiert bis zur Exmatrikulation, gleichgültig, wie sie zustande kommt. Ein Fachwechsel oder eine Namensänderung implizieren keine Exmatrikulation. Hat sich eine Person im Laufe ihres Lebens mehrfach an der Universität Bremen immatrikuliert, so handelt es sich um mehrere, nicht identische Studenten.
<b>Bezeichnung</b>	Ein Student ist durch die Matrikelnummer und einen Zeitpunkt (zu dem die Matrikelnummer gültig war oder ist) eindeutig bestimmt, alle anderen Attribute, insbesondere der Name, können mehrfach vorkommen.
<b>Unklarheiten</b>	Es ist noch ungeklärt, wie Namen aus anderen Schriftsystemen (z. B. Russisch, Arabisch, Chinesisch) dargestellt werden.
<b>Querverweise</b>	Gasthörer, Matrikelnummer, Studentenausweis



# Warum machen wir das Ganze?



# Wer liest überhaupt das Pflichtenheft?

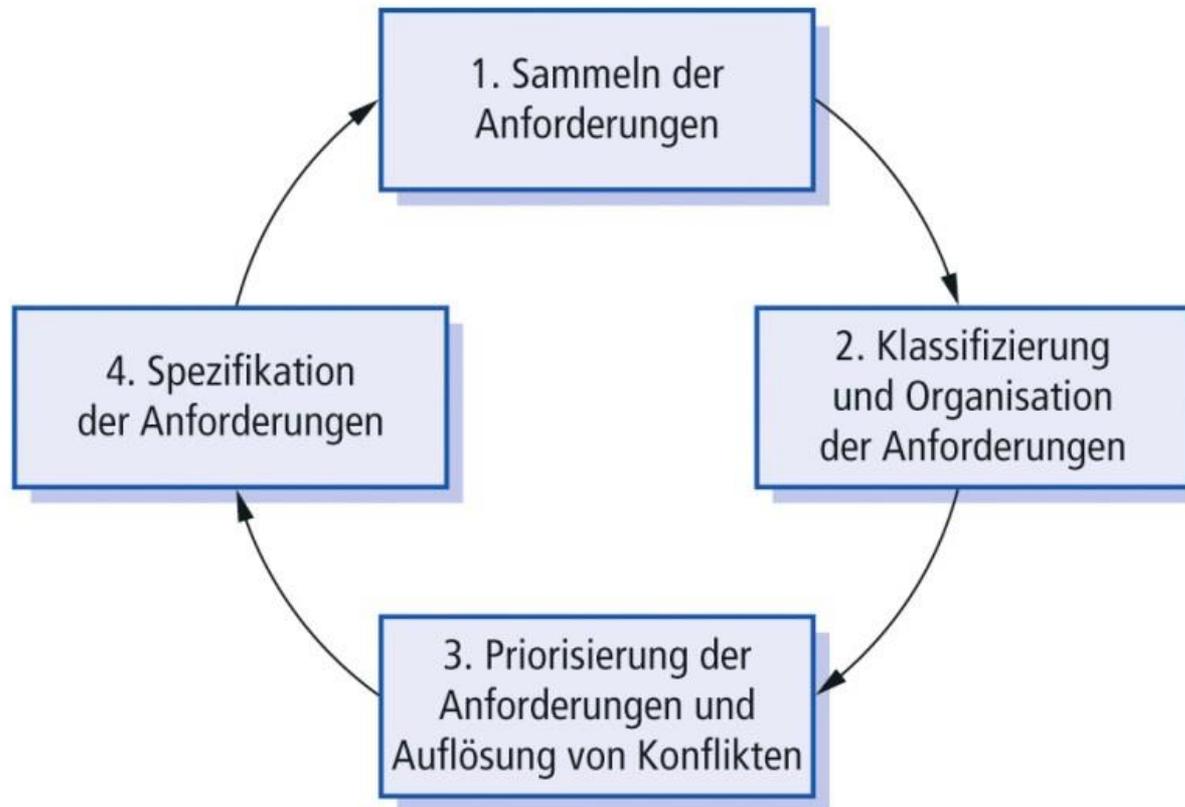


- In der Praxis gibt es viele schlechte Anforderungsanalysen; oft gibt es gar keine
- Die Anforderungsanalyse ist aber notwendig für
  1. die **Abstimmung** mit dem **Kunden** bzw. mit dem Marketing,
  2. den **Entwurf** und die **Implementierung**,
  3. das **Benutzungshandbuch**,
  4. die **Testvorbereitung**,
  5. die **Abnahme**,
  6. die **Wiederverwendung**,
  7. die **Klärung** späterer Einwände, Regressansprüche usw.,
  8. eine spätere **Re-Implementierung**.

1. Die Anforderungen bleiben **ungeklärt**, sie werden darum auch nicht erfüllt.
2. Den Entwicklern **fehlt die Vorgabe**, darum fragen sie „auf dem kurzen Dienstweg“ Bekannte, die beim Kunden arbeiten, oder sie legen mangels Kontakten die eigenen Erfahrungen und Erwartungen zu Grunde.
3. Die **Basis für das Handbuch fehlt**, es wird darum phänomenologisch, d.h. experimentell, verfasst.
4. Ein gutes Handbuch ist ein umformulierter Auszug aus der Anforderungsanalyse!
5. Ein **systematischer Test** ist ohne Anforderungsanalyse unmöglich, denn es ist nicht definiert, welche Daten das System akzeptieren muss und welche Resultate es liefern soll.

6. Wenn bei der **Abnahme** nicht entschieden werden kann, ob das System richtig arbeitet, wird die Korrektheit zur Glaubensfrage.
  7. Oft zeigen sich **echte oder vermeintliche Mängel** der Software erst nach längerem Gebrauch. Ohne Anforderungen kann diese Unterscheidung aber nicht getroffen werden.
  8. Wer eine Software(-Komponente) **wiederverwenden** will, muss wissen, was sie leistet. Das ist in der Spezifikation dokumentiert.
  9. Wenn ein System ausgemustert und ersetzt wird, ist **Aufwärtskompatibilität** gefordert (vgl. Heninger et al., 1978).
- „**Spezifikation im Kopf**“ gibt es nicht!

- Requirements-Engineering ist kein linearer Prozess, sondern ein iterativer





*"Based on our tests, the business stakeholders fall asleep around page 37 of the Functional Requirements Specification. Put the Issues Section on page 40."*