

Grundlagen / Begriffe:

Requirement = Anforderung = eine Aussage über eine zu erfüllende Eigenschaft oder zu erbringende Leistung eines Produktes, Systems oder Prozesses. [Wikipedia]

Sie können in **Anforderungsspezifikationen** (oft auch **Lastenheft**), in Datenbanken oder anderen Dokumenten zusammengefasst oder auch völlig undokumentiert sein (was oft zu Problemen führt!).

Stakeholder = Projektbeteiligter oder Projektbetroffener. Es ist wesentlich für den Projekterfolg, dass für die Anforderungsspezifikation möglichst alle Stakeholder einbezogen werden!

Requirements-Provider = Der/die Stakeholder, die eine bestimmte Anforderung definiert haben bzw. das Entstehen dieser Anforderung bewirkt oder verantwortet haben.

Arten von Requirements

Funktionale Anforderungen: legen fest, was das System tun soll und wie es sich darstellt.

Beispiele: „Das System soll die Stamm-Daten von Patienten verwalten.“ oder „Das System soll einen Report über die durchgeführten Untersuchungen ausgeben.“

Nichtfunktionale Anforderungen: legen fest, welche Eigenschaften das System haben soll.

Beispiele: „Das System soll dem Anwender innerhalb von 2 Sekunden antworten.“ oder „Die Daten müssen vor unberechtigten Zugriffen geschützt sein.“

Zusätzlich werden oft auch **Randbedingungen** als eigene Art von Anforderungen beschrieben.

Beispiele: Obergrenze für die Kosten, einzuhaltende Termine, eine bestimmte zu verwendende Technologie.

Hinweis: Die hier angeführten Beispiele sind nur grobe Anforderungen, die in Projekten noch weitere verfeinert werden müssen!

Prozess / Vorgehen:

Es gibt 2 Bereiche, die für die Requirements-Spezifikation von zentraler Bedeutung sind:

1. Requirements-Engineering Prozesse: In diesen (Teil-)Prozessen entsteht die Requirements-Spezifikation.
2. Requirements-Management Prozesse: In diesen (Teil-)Prozessen wird die Requirements-Spezifikation im Laufe des Projekt weiterentwickelt bzw. kontrolliert und nachvollziehbar verwaltet und verändert.

Die Teilprozesse dieser beiden Prozess-Bereiche sind nachfolgend im Überblick dargestellt:

Requirements-Engineering-Prozesse

Ziel-Definition, Scope, Abgrenzung, Stakeholder, ...

Requirements-Übersicht (Feature-Lists)

Requirements-Verfeinerung

Requirements-Qualitäts-Sicherung (Reviews)

Requirements-Management-Prozesse

Requirements-Bewertung (Risiko, Kosten, Zeiten, Machbarkeit, ...)

Requirements-Change- & Konfigurations-Management

Wie formulieren ?

Requirements gut zu formulieren, ist keine leichte Aufgabe und bedarf in vielen Fällen auch intensiver Übung mit einem erfahrenen Requirements-Engineer oder Analytiker.

Nachfolgend sind einige Eigenschaften und Richtlinien angegeben, die bei der Formulierung von Requirements beachtet werden sollten:

Jedes einzelne Requirement muss ...

- mindestens einem Requirements-Provider zugeordnet sein. Wenn es keinen gibt, warum sollte diese Anforderung dann realisiert werden?
- klar und verständlich formuliert sein. Wie soll sonst eine klare Auftragsvergabe bzw. ev. sogar ein Fixpreis möglich sein?
- prüfbar sein. Auf welcher Basis soll sonst eine Abnahme des Systems erfolgen?
- verstanden werden - sowohl vom Auftraggeber als auch vom Auftragnehmer und den betroffenen Stakeholdern.
- identifizierbar sein (wenn möglich eindeutig nummerieren, damit die Anforderung in anderen Dokumenten (z.B. Umsetzungsdokumente, Testspezifikation, ...) referenziert werden kann).
- möglichst atomar formuliert sein (kurze Sätze, Und-Verknüpfungen auflösen, ...)
- positiv formuliert sein (negative Formulierung wie Nicht, Nie, ... können oft nicht geprüft werden).
- Füllwörter, Weichmacher und Konjunktive vermeiden (eventuell, soll, sollte, könnte, wenn möglich, ...)

Die Requirements-Spezifikation insgesamt muss ...

- in sich konsistent sein (keine Widersprüche zwischen einzelnen Anforderungen bzw. Widersprüche sollten vor Freigabe klar aufgelöst werden)
- vollständig sein. Es sollte möglichst keine „impliziten“ Anforderungen mehr geben (die einzelne Stakeholder im Kopf haben und die noch nicht erfasst wurden).

Es gibt noch weitere Kriterien und Richtlinien, die jedoch hier aus Platzgründen nicht angeführt werden.

Beispiele:

Schlechtes Beispiel

- ➔ Es soll eine Befundverwaltung implementiert werden.
 - ☹ Ist sehr ungenau.
Was soll das System genau können?
 - ☹ Weichmacher „soll“ vermeiden.
Stattdessen schreiben, was der Benutzer mit dem System machen kann bzw. was das System leisten muss.
- ➔ Eine Statistik über die Befunde ist notwendig.
 - ☹ Viel zu ungenau.
Was soll in der Statistik genau enthalten sein?
 - ☹ Wie kommt der Systemanwender zu dieser Statistik?
 - ☹ Wie soll die Statistik aussehen?
 - ...
- ➔ Es wird eine Schnittstelle zum KIS benötigt.
 - ☹ Das ist viel zu ungenau.
 - ☹ Welche genauen Daten (Aufbau, Inhalt, Semantik, ...) sollen übertragen werden?
 - ☹ Sollen die Daten nur in eine Richtung übertragen werden oder in beide Richtungen?
Soll die Übertragung synchron oder asynchron erfolgen?
 - ☹ Was passiert, wenn die Daten fehlerhaft sind?
 - ☹ KIS sollte erklärt sein (im Text oder im Abkürzungsverzeichnis).
 - ...
- ➔ Das System soll benutzerfreundlich sein.
 - ☹ Was heißt „benutzerfreundlich“?
Das ist eine zu unklare Spezifikation!
- ➔ Das System soll ausreichend schnell reagieren.
 - ☹ Was heißt „ausreichend schnell reagieren“?
Das ist eine zu unklare Spezifikation!

Verbessertes Beispiel

- ➔ 10. Der Benutzer kann Befunde mit dem System verwalten:
 - 10.1. Der Benutzer kann Befunde neu anlegen.
 - 10.1.1. Beim Anlegen sind folgende Befundfelder verpflichtend auszufüllen:
 - 10.1.1. Beim Anlegen können folgende Felder optional ausgefüllt werden:
 - 10.2. Der Benutzer kann Befunde löschen.
 - 10.3. Der Benutzer kann Befunde ändern.
... (weitere Details beschreiben)
- ➔ 20. Das System muss dem Benutzer ermöglichen, eine Statistik über die Befunde zu erstellen.
 - 20.1. die Befundstatistik enthält folgenden Daten:
 - 20.2. die Befundstatistik kann vom Benutzer vorparametriert werden, wobei folgende Parameter in den angegebenen Wertebereichen möglich sein sollen: ...
 - 20.3. der Benutzer kann die für die Befundstatistik verwendeten Daten
 - auf und absteigend sortieren nach den Feldern ...
 - selektieren nach den Feldern ..., wobei die Select-Funktionalität des Standards SQL 2005 bereitgestellt werden muss.
 - ... (weitere Details beschreiben)
- ➔ 20. Das System muss eine Schnittstelle zum KIS (Krankenhaus-Informationssystem) bereitstellen.
 - 20.1. Die Schnittstelle überträgt die Patientenstammdaten (detaillierte Beschreibung mit Pflichtfeldern, Feldstruktur, Datenformaten, ... in Beiblatt S13).
 - 20.2. Die Übertragung erfolgt nur in eine Richtung vom KIS (Master) zum System.
 - 20.3. Die Übertragung erfolgt synchron, wobei das Verhalten des Systems und des KIS während der Übertragung, nach Beendigung der Übertragung oder im Fehlerfall in S14 detailliert beschrieben sind.
 - ... (weitere Details beschreiben)
- ➔ 30. Das System muss den „Styleguide xyz“ für die Gestaltung der Benutzeroberfläche einhalten.
- ➔ 40. Die Antwortzeit des Systems in der Referenz-Testumgebung und in der Referenz-Konfiguration (genau definiert in Beiblatt T17) darf für
 - die in Liste L23 angeführten Benutzereingabemasken und Reports max. 3 Sekunden
 - die in Liste L24 angeführten Benutzereingabemasken und Reports max. 10 Sekunden
 - für alle anderen Benutzereingabemasken und Reports max. 60 Sekunden betragen.
 Die Antwortzeit wird dabei wie folgt gemessen:
 - ...

Empfehlungen / Tips:

- ➔ **Erarbeiten Sie die Anforderungen in mehreren Stufen / Runden**, in denen Sie die einzelnen Anforderungen **immer weiter verfeinern** (hier sind oft Mindmaps oder Baumstrukturen für die Gliederung hilfreich), bis klar ist, wie jedes einzelne Requirement in der Abnahme geprüft werden soll. Es hat sich bewährt, zuerst die Prozesse/Detailabläufe zu überlegen und davon ausgehen die Anforderungen weiter zu spezifizieren.
- ➔ Der **Detaillierungsgrad** wird durch den **Requirements-Provider (R.P.) gesteuert. Alles was dem R.P. wichtig ist** und so realisiert werden muss, **sollte auch** als Anforderung **spezifiziert werden**. Dies kann sehr grob sein wie z.B. „Das System soll Kunden verwalten können.“ wenn es für den R.P. unwichtig ist, was bei der Verwaltung im Detail passiert (Anm. das ist meist jedoch unwahrscheinlich) oder aber auch sehr detaillierte Systemeigenschaften wie z.B. „Das Feld Vorname muss mindestens 30 Zeichen lang sein und rechtsbündig und in roter fetter Schrift dargestellt werden.“
- ➔ Die **Anforderungen**, die spezifiziert werden, immer auch von anderen Beteiligten/Betroffenen noch mal **reviewen** lassen. Möglichst viele Stakeholder in die Spezifikationserstellung einbeziehen (zumindest als Reviewer).