

Anforderung und Spezifikation

Anforderung

1. Eine Bedingung oder Fähigkeit, die von einem Benutzer zur Lösung eines Problems oder zur Erreichung eines Ziels benötigt wird.
2. Eine Bedingung oder Fähigkeit, die ein System erfüllen oder besitzen muss, um einen Vertrag, eine Norm, eine Spezifikation oder andere, formell vorgegebene Dokumente zu erfüllen.
3. Eine dokumentierte Repräsentation einer Bedingung oder Fähigkeit nach (1) oder (2).

Anforderungsspezifikation

Zusammenstellung (aller) Anforderungen an ein System.

Institute of Electric and Electronic Engineers: IEEE Standard Glossary of Software Engineering Terminology (IEEE Std. 610.12-1990)

Anforderungsspezifikation heißt auch Anforderungsdokument

Bei Software insbesondere auch Software Requirements Specification

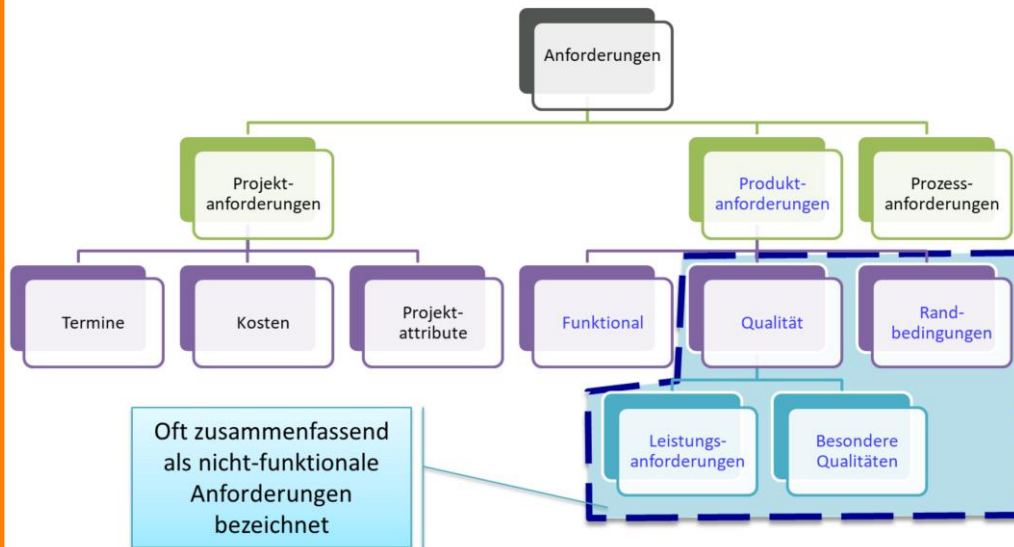
„Die Spezifikation“ ist als Begriff nicht immer eindeutig:

Prozess oder Dokument

Anforderungs-, Lösungs- oder Produktspezifikation

Aus der Definition für Anforderung folgt, dass sowohl nicht dokumentierte als auch dokumentierte Bedingungen und Eigenschaften im Sinne von (1) und (2) als Anforderungen bezeichnet werden. Daher verwenden bestimmte Bücher (z.B. Pohl 2008) für **dokumentierte Anforderungen den Begriff „Anforderungsartefakt“**

Kategorisierung von Anforderungen



19.10.2018

Software Engineering 2

17

Produktanforderungen heißen in unserem Kontext auch Systemanforderungen

Bei den funktionalen Anforderungen werden typischerweise drei Perspektiven betrachtet, die einander ergänzen: **Funktionsperspektive, Datenperspektive und Verhaltensperspektive**

Funktion: Manipulation der Daten; Eingabe, Verarbeitung und Ausgabe

Daten: Statische Struktur; Verwendung, Erzeugung und Speicherung

Verhalten: Wie reagiert das System auf externe Stimuli

Bei der Einteilung in funktionale und nicht-funktionale Anforderungen legen die funktionalen Anforderungen fest, was das Produkt tun muss, also welche Aktionen es ausführen muss, um den fundamentalen Gründen für seine Existenz zu genügen. Die nicht-funktionalen Anforderungen beschreiben, welche Qualitäten das Produkt haben muss oder wie „gut“ es seine Aufgaben erledigt. Sie machen das Produkt attraktiv, benutzbar, schnell, verlässlich oder sicher.

HINWEIS:

Die Unterscheidung in funktionale und nicht-funktionale Anforderungen ist umstritten, es gibt Fachleute im RE die von dieser Unterscheidung abraten. Besser sei die auf dieser Folie auch dargestellte Dreiteilung der Produkthanforderungen. Begründung der Vertreter dieser Sichtweise :

Nicht-funktionale Anforderungen sind typischerweise 1) Qualitätsanforderungen oder 2) unspezifizierte funktionale Anforderungen. Gefahr ist, dass eine als nicht-funktionale Anforderung eingestufte Anforderung nicht weiter hinterfragt wird und somit unspezifizierte Anforderungen in die Spezifikation eingehen.

Beispiel: „Das System soll sicher sein.“

Was bedeutet hier sicher? Welche Eigenschaften muss das System haben, damit es „sicher“ ist? Wie kann überprüft werden, ob das erstellte System die Eigenschaft „sicher“ hat?

Randbedingungen sind Restriktionen, die durch die Projektbeteiligten schwer oder gar nicht verändert werden können. Sie schränken entweder das zu entwickelnde System oder den Entwicklungsprozess ein. Ein Beispiel sind Vorgaben zur Implementierung aber auch gesetzliche Vorgaben.

Prozessanforderungen sind Anforderungen an den Erstellungsprozess des Produkts, nicht das Produkt selbst

Abschnitt 1.3

VARIANTEN VON SOFTWAREPROJEKTEN

- **Neuentwicklung:** System wird komplett neu konzipiert und realisiert: Keine Altlasten, keine Datenmigration
- **Reengineering-Projekt:** Verbesserung oder Weiterentwicklung eines bestehenden Systems, möglicherweise Plattformwechsel
- **Migrationsprojekt:** Übertragung von Daten aus einer (alten) Datenquelle in eine andere. Reines Migrationsprojekt hat nur geringe Implementierungsanteile
- **Weiterentwicklungsprojekt:** Erweiterung oder Verbesserung eines Systems im Produktiveinsatz

Neuentwicklung: RE-Herausforderung ist, die Anforderungen zu finden, die den tatsächlichen Erfordernissen der späteren Anwender entsprechen.

Reengineering: Extrahieren der Anforderungen aus dem bestehenden System (manchmal Software-Archäologie genannt): Freilegen der Funktionalität aus dem Quelltext (und wenn man viel Glück hat, aus früheren RE-Dokumenten). Je länger ein System schon im Einsatz war, desto unwahrscheinlicher ist eine Spezifikation/Dokumentation, die mit der Systemrealität konsistent ist.

Migration: RE-Herausforderung hier ist die Analyse der alten Daten sowie die technischen Randbedingungen der eigentlichen Datenübertragung. Gibt es VIEWS, TRIGGER, STORED PROCEDURES? Können diese abgebildet werden, werden sie überhaupt noch benötigt?

Weiterentwicklung: Aus RE-Sicht eher unkritisch, sofern ein geordneter Änderungsmanagementprozess existiert und die Systemartefakte (auch RE-Dokumente) aktuell sind. Wichtig ist vor allem eine ausreichende Zahl von Regressionstests.

- **Individualsoftware:** Wird im Auftrag eines einzelnen Kunden speziell für diesen entwickelt, Kunde zahlt für die Entwicklungsarbeit
- **Software-Produkt:** Wird für eine beliebige Anzahl potentieller Kunden entwickelt, Kunden Zahlen für die Benutzung des Produkts (über unterschiedliche Lizenzformen)

Individualsoftware: **Zukünftige Anwender des Produkts stehen seitens des Auftraggebers in der Regel für das Requirements Engineering zur Verfügung.** Aus diesen (und anderen) Stakeholdern sind möglichst effizient die Anforderungen zu erfahren

Software-Produkt: **Zugriff auf zukünftige Anwender eher schwierig**, möglich sind aber Umfragen oder Fokusgruppen. Letztlich liegt die Verantwortung für die Festlegung der Funktionalität beim Hersteller des Produkts

- **Datenzentriertes System:** Primäraufgabe des Systems ist die Verwaltung von Daten als zentraler Ressource: Erzeugen, Ändern, Persistieren, Löschen und Transformieren
- **Eingebettetes System:** Systeme aus Hard- und Softwarekomponenten, mit Sensoren und Aktuatoren zur Untersuchung bzw. Veränderung der Umgebung. Reagiert auf Ereignisse. Softwareanteil für den Anwender nicht unmittelbar erkennbar

Datenzentriertes System: Im RE spielt hier die Datenmodellierung die zentrale Rolle. Beispiele: Entleihsystem für Bibliothek, ERP-Systeme, Webanwendungen, Onlineshops, Software in Banken und Versicherungen. Datenbegriff ist sehr weit gefasst. Daten repräsentieren meist irgendwelche Entitäten der realen Welt.

Eingebettetes System: Sehr eng gefasster Aufgabenbereich, Hauptaufgabe ist die zuverlässige Steuerung des umgebenden Geräts. Beispiele: Komponenten in Flugzeugen und Autos, Steuerelektronik der Waschmaschine oder Heizungssteuerung. **RE-Herausforderung: System ist verhaltensgesteuert: Es muss auf Ereignisse reagieren: Zustandsmodellierung/State-machines** kommen hier häufig vor oder auch Matlab-Simulink-Modelle

[-> Beispiele-Systemtypen]

- **Inhouse-Projekt:** Kunde und Ersteller des Systems sind aus derselben Organisation, Systemerstellung erfolgt für den Eigenbedarf. Systemerstellung durch IT-Abteilung, Auftraggeber ist ein Fachbereich. Meist keine formalisierte Vertragssituation zwischen Auftraggeber und -nehmer
- **Vergabeprojekt:** Systemersteller nicht aus der Organisation des Auftraggebers (Fremdvergabe), explizite Vertragssituation zwischen Auftraggeber und -nehmer
- **Ausschreibungsprojekt:** Variante des Vergabeprojekts, Auftraggeber holt im Rahmen einer Ausschreibung Angebote verschiedener Dienstleister ein. Zuteilung nach Bewertung der Angebote anhand definierter Kriterien. Explizite Vertragssituation zwischen Auftraggeber und -nehmer, typischerweise Festpreisprojekt

[-> Beispiele-Projektvarianten-neu, Lösung-Beispiele-Projektvarianten]