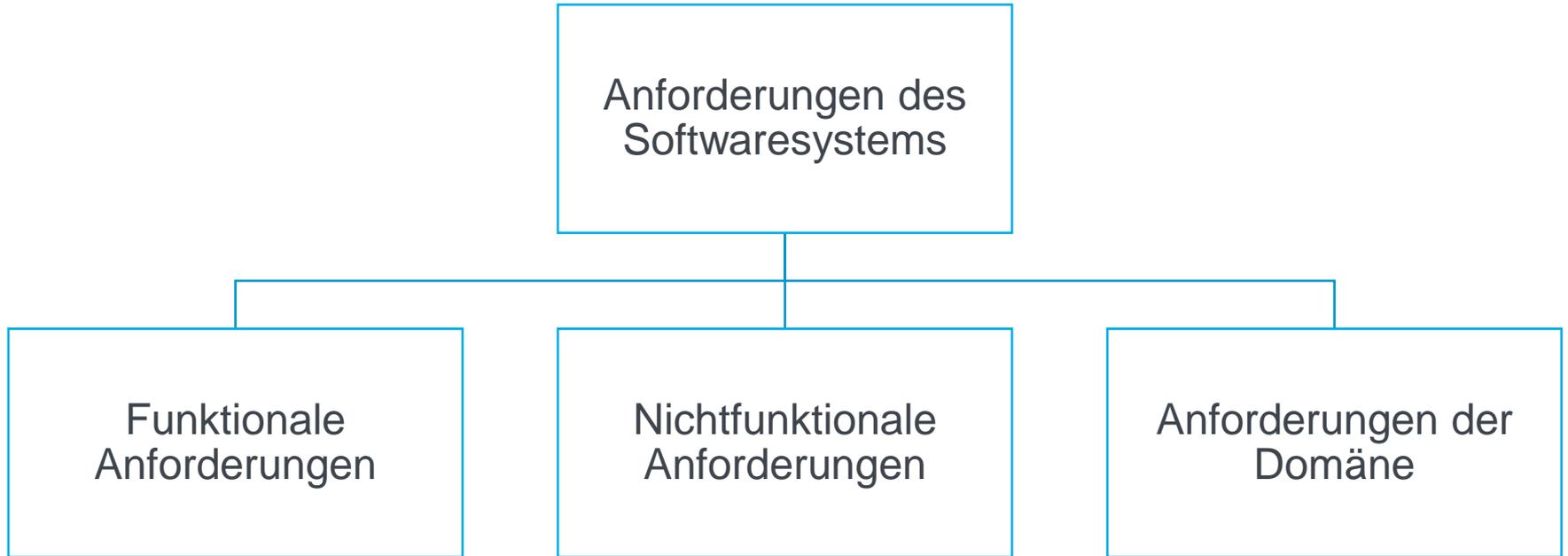


Einführung in das Requirements Engineering

Teil 3 – Anforderungsklassifikation

Arten von Anforderungen



Arten von Anforderungen

- Rahmenbedingungen, die das System einhalten muss
- Funktionalität, die das System bereitstellen muss
- Anforderungen, die in dieser Domäne oft stillschweigend erwartet werden

Nichtfunktionale
Anforderungen

Funktionale
Anforderungen

Anforderungen
der Domäne

Beispiel: Recommender System Szenario

- Metadaten: Das System soll aktuelle und genaue Informationen über Filme zur Verfügung stellen. Das beinhaltet Bewertungen, Verfügbarkeit und Kritiken. (Anforderung der Domäne)
- Nutzbarkeit: Das System soll von Nutzern mit Behinderungen benutzt werden können. Dafür soll es gängigen Kriterien folgen und passende Assistenzsysteme anbinden. (Nichtfunktionale Anforderung)
- Ausleihfunktion: Das System soll einen Ausleihprozess unterstützen. Ein Benutzer kann verfügbare Filme suchen und ausleihen. Das System soll die Ausleihfristen beobachten und ausgeliehene Filme automatisch zurück geben. (Funktionale Anforderung)

Ungenauere Anforderungen

- Bei ungenauer Formulierung von Anforderungen können leicht Probleme entstehen.
- Ungenauere Anforderungen können von Entwicklern und Nutzern unterschiedlich interpretiert werden.
- Was ist die Zielgruppe der Anforderungsdokumente?
 - “Die Anwendung muss Dokumente der Datentypen X, Y und Z darstellen”
 - Anwenderabsicht – spezialisierte Anzeige für alle Dokumentenarten
 - Entwicklerinterpretation – Textanzeige des Dokumenteninhalts

Ungenauere Anforderungen

- **Benutzerfreundliche Oberfläche (Was heißt das?)**
 - **Besser: Ein Nutzer soll sich die Funktionen der Software ohne Handbuch erarbeiten können**
- **Ständiges Lernen und Weiterentwicklung (Wie genau? Woran wird das festgemacht?)**
 - **Besser: Der Empfehlungsalgorithmus soll sein Empfehlungsmodell fortlaufend mit den gesehenen Filmen eines Nutzers updaten**
- **Akzeptable Antwortzeiten (Wie lange ist das? Unter welcher Systemlast? Mit welcher Toleranz?)**
 - **Besser: Das System soll bei einer mittleren Last von 200 gleichzeitigen Nutzern auf einen Mausklick zu 95% innerhalb von 700ms antworten.**

Vollständigkeit und Konsistenz

- Generell sollten Anforderungen vollständig und konsistent sein.
- Vollständig
 - Sie sollten alle notwendigen Informationen enthalten.
- Konsistent
 - Anforderungen sollten sich nicht gegenseitig widersprechen
- In der Praxis ist es meist nahezu unmöglich, Anforderungen konsistent und vollständig zu formulieren.

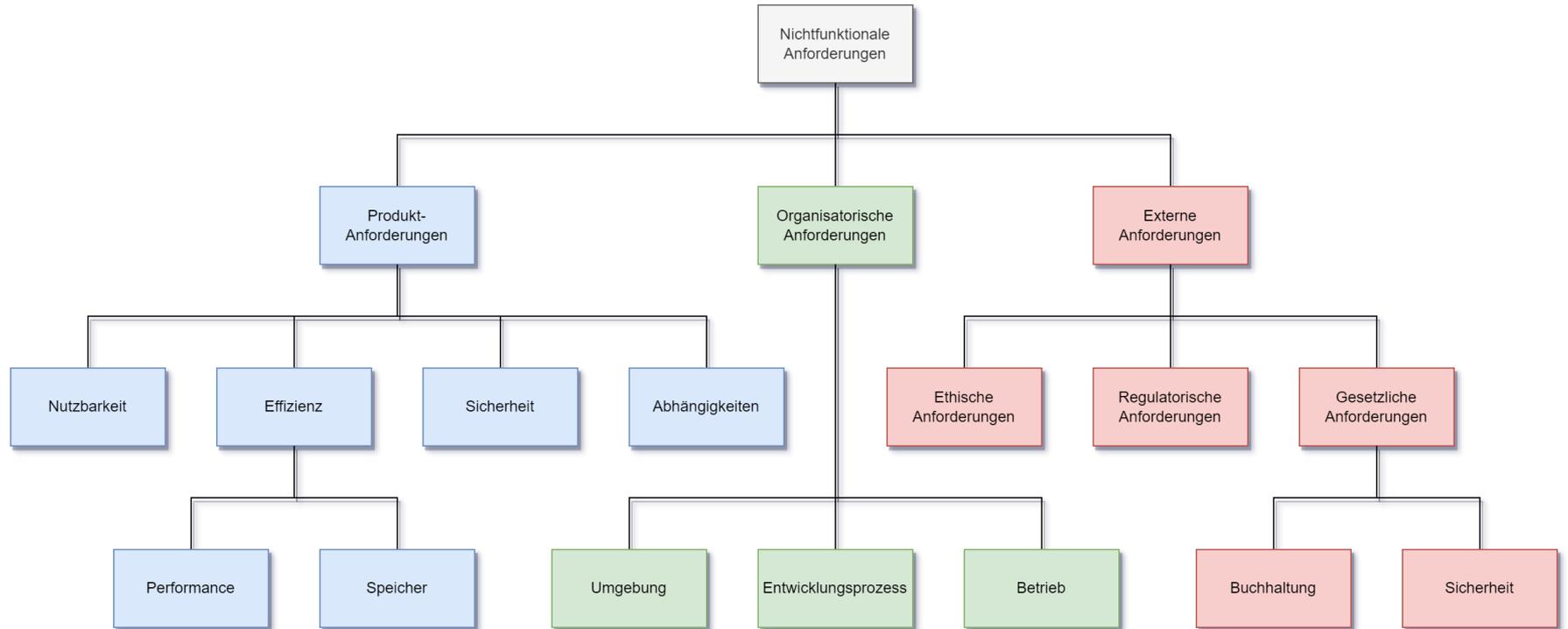
Klassifikation von funktionalen Anforderungen

- Produkt-Anforderungen
 - Anforderungen, die das Verhalten des Produkts festlegen, z.B. Geschwindigkeit, Zuverlässigkeit, etc.
- Organisatorische Anforderungen
 - Anforderungen, die aus organisatorischen Regeln oder Prozessen folgen, wie bspw. Anforderungen an den verwendeten Prozesse oder die Implementierung.
- Externe Anforderungen
 - Anforderungen von Extern, wie z.B. Dateiformate oder gesetzliche Anforderungen

Beispiele

- **Produkt-Anforderungen:**
 - **Performance-Anforderung:** Die Software muss bis zu 200 gleichzeitige Nutzer unterstützen.
- **Organisatorische Anforderungen:**
 - **Budget-Limit:** Die Kosten für Implementierung, Hardware und Wartung sollen 50 000 € nicht übersteigen.
- **Externe Anforderungen:**
 - **Regulatorische Anforderung:** Die Software muss sich an die DSGVO und weitere relevante Datenschutzrichtlinien halten, um die Datensicherheit und -vertraulichkeit zu gewährleisten.

Arten Nichtfunktionaler Anforderungen



Konflikte zwischen Anforderungen

- In komplexen Systemen kommt es häufig zu Konflikten zwischen verschiedenen nichtfunktionalen Anforderungen.
- **Beispiel:**
- Performance-Anforderung vs Datensicherheit



Fallen dir weitere Konflikte ein?

Probleme von Anforderungen

- Verständlichkeit
 - Anforderungen werden oft in Fachsprache ausgedrückt;
 - Wenn sich die Softwareentwickler nicht im Fachbereich auskennen, verstehen sie die Anforderungen nicht richtig.
- Selbstverständlichkeiten
 - Fachexperten denken oft innerhalb des Fachgebiets, und vergessen oft, allgemeine Anforderungen aus dem Fachgebiet explizit aufzuschreiben.

Tobias Eisenreich

Universität Stuttgart
Institut für Software Engineering
Empirisches Software Engineering

Umm-e-Habiba

Universität Stuttgart
Institut für Software Engineering
Empirisches Software Engineering



Universität Stuttgart

Institut für Maschinelle Sprachverarbeitung
Institut für Software Engineering



Industrie- und Handelskammer
Reutlingen

Reutlingen | Tübingen | Zollernalb



Region Stuttgart



Industrie- und Handelskammer
Karlsruhe



LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

Lizenzbestimmungen

„Anforderungsklassifikation“ – von Umm-e-Habiba und Tobias Eisenreich

Das Werk - mit Ausnahme der folgenden Elemente:

- Logos der Verbundpartner und des Förderprogramms
- im Quellenverzeichnis aufgeführte Medien

ist lizenziert unter:

 [CC BY 4.0 \(https://creativecommons.org/licenses/by/4.0/deed.de\)](https://creativecommons.org/licenses/by/4.0/deed.de)

(Namensnennung 4.0 International)

Quellenverzeichnis

Titelfoto: <https://unsplash.com/de/fotos/braune-holzschublade-IRoX0shwjUQ>