

# LU05b - Anforderungen

## Funktionale Anforderungen

Funktionale Anforderungen beschreiben gewünschte Funktionalitäten (was soll das System tun/können) eines Systems bzw. Produkts, dessen Daten oder Verhalten. Um funktionale Anforderungen vollständig zu erheben, können verschiedene Methoden herangezogen werden. In benutzerorientierten Systemen können beispielsweise Usecases eingesetzt werden. Ziel ist es, die funktionalen Anforderungen an ein System zu erheben, um eine optimale Unterstützung der Benutzer bei der Erfüllung individueller Aufgaben zu ermöglichen. Nachfolgend einige Beispiele für funktionale Anforderungen:

- **Funktion (Was soll es grundsätzlich tun?):** Ein Bestellsystem sollte in der Lage sein, Bestellungen zu verarbeiten und den Status der Lieferung zu überwachen.
- **Verhalten:** Ein Ticketing-System sollte bei der Buchung einer Veranstaltung eine Bestätigungse-Mail senden.

## Nichtfunktionale Anforderungen

Im Gegensatz zu funktionalen Anforderungen, die beschreiben, WAS ein System leisten soll (funktional), geben nichtfunktionale Anforderungen an, WIE GUT ein System etwas leisten soll (qualitativ). Die nichtfunktionalen Anforderungen an ein System können unterschiedlichster Art sein. So wurden im Rahmen des ISO Standards 9126 unter anderem folgende Typen von nichtfunktionalen Anforderungen (Qualitätsattribute) identifiziert:

- Performanz
- Funktionalität (Feature)
- Usability
- Portabilität
- Sicherheit

Ziel ist es, klare Qualitätsvorgaben an die systemunterstützten Geschäftsprozesse und Benutzeraufgaben zu erheben und schließlich umsetzen zu können. Folglich ist die zentrale Frage:

**«Welche Restriktionen/Einschränkung muss die Lösung einhalten?».**

**Beispiel:** „Das Produkt soll dem Anwender innerhalb von einer Sekunde antworten.“

## Unterscheidung

Um funktionale und nichtfunktionale Anforderungen effektiv unterscheiden zu können, ist es wichtig, die charakteristischen Eigenschaften und die Rollen, die sie in der Systementwicklung spielen, zu verstehen. Hier sind einige Schlüsselpunkte:

## Funktionale Anforderungen

- **Was das System tun soll:** Funktionale Anforderungen beschreiben die spezifischen Funktionen, die ein System ausführen muss.
- **Datenverarbeitung:** Sie legen fest, welche Art von Daten vom System empfangen, verarbeitet und ausgegeben werden sollen.
- **Klare Handlungen:** Sie sind meist als klare Handlungen oder Aufgaben formuliert, die das System ausführen muss.
- **Testbarkeit:** Sie sollten in einer Art und Weise formuliert werden, dass sie leicht testbar sind, normalerweise auf einer Pass/Fail-Basis.

## Nichtfunktionale Anforderungen

- **Wie gut das System etwas tun soll:** Diese Anforderungen beschreiben die Qualität oder Leistung, mit der ein System seine Funktionen ausführen soll.
- **Qualitätsattribute:** Sie betreffen oft „-ilities“ wie Zuverlässigkeit, Skalierbarkeit, Leistung, Sicherheit und Kompatibilität.
- **Benutzererfahrung:** Dazu können auch Anforderungen an die Benutzeroberfläche und die Benutzererfahrung gehören.
- **Messbarkeit:** Obwohl schwieriger zu messen als funktionale Anforderungen, sollten sie dennoch quantifizierbar sein, um die Systemleistung effektiv bewerten zu können.

## Gemeinsame Merkmale

- **Dokumentation:** Beide Arten von Anforderungen sollten klar und unmissverständlich dokumentiert sein.
- **Stakeholder-Involvement:** Beide erfordern die Einbeziehung von Stakeholdern, um sicherzustellen, dass die Anforderungen sowohl machbar als auch im Einklang mit den Geschäftszielen stehen.

Durch das Verständnis dieser Unterscheidungen können Teams effektiver kommunizieren und die jeweiligen Anforderungen besser in Projektmanagement- und Entwicklungsaufgaben integrieren.

Der Begriff „-ilities“ bezieht sich auf eine Gruppe von Qualitätsattributen oder nichtfunktionalen Anforderungen, die oft auf „-ility“ enden. Diese Qualitätsattribute beschreiben, wie gut ein System seine Aufgaben ausführen soll. Sie beziehen sich auf die Qualität der Systemleistung und decken verschiedene Aspekte der Benutzererfahrung, der Zuverlässigkeit und der Effizienz eines Systems ab.



Einige gängige „-ilities“ sind:

- **Usability:** Wie einfach ist es für die Benutzer, das System zu verwenden?
- **Reliability:** Wie zuverlässig ist das System?
- **Scalability:** Wie gut kann das System mit zunehmender Last skaliert werden?
- **Availability:** Ist das System immer verfügbar, wenn es benötigt wird?
- **Maintainability:** Wie einfach ist es, das System zu warten oder zu aktualisieren?
- **Testability:** Wie einfach lässt sich das System testen?

- **Security:** Wie sicher ist das System gegenüber unbefugtem Zugriff oder Datenverlust?



Diese Qualitätsattribute werden oft in der Planungs- und Anforderungsphase eines Projekts festgelegt, um den Entwicklungsprozess zu leiten und sicherzustellen, dass das Endprodukt den Erwartungen der Stakeholder entspricht.

M431-LU05, M431-A1G, M431-A1F



Kevin Maurizi, Marcel Suter

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/modul/m431/learningunits/lu05/anforderungen>

Last update: **2024/03/28 14:07**

