

# Qualitätsanforderungen an Embedded-Software – Teil 1: Anforderungen und Mythen

Anforderungen zu erfassen und zu verwalten ist ein wesentlicher Schlüssel zum Projekterfolg. Die Embedded-Software-Funktionalität lässt sich einfacher in Anforderungen beschreiben als die Qualitätsmerkmale. Dennoch: Qualitätsmerkmale lassen sich nicht am Ende einfach „hineintesten“. Je abstrakter sie sind, desto aufwendiger ist ihre Erfassung. Welche Herausforderung stellt das für ein Projekt dar?

Auch Qualitätsanforderungen unterliegen der Mythenbildung. Deshalb ist es wichtig, sich die unterschiedlichen Arten von Anforderungen genau anzuschauen.

Drei Arten von Anforderungen lassen sich unterscheiden:

**Funktionale Anforderungen** beschreiben die nach außen hin sichtbaren Funktionalitäten der Embedded-Software. Beispiel für eine funktionale Anforderung: *Die Kaffeefullautomat-Software muss fähig sein, sich zu aktualisieren.*

**Zusicherungen als Anforderungen:** Zusicherungen (Constraints) sind allgemeine, globale Zusicherungen, die für das Produkt und/oder für die Produktentwicklung gelten. Auch die Projekt-Randbedingungen gehören zu den Zusicherungen.

Wir differenzieren darüber hinaus Unterarten von Zusicherungsanforderungen:

- **Technische Zusicherungen** sind Anforderungen, die technische Vorgaben zur Realisierung festlegen, z.B. Bestimmung der Programmiersprache oder des Prozessors. Beispiel einer technischen Zusicherung: *Die Kaffeefullautomat-Software muss mit der UML (Unified Modeling Language) modelliert sein.*
- **Projektbezogene Zusicherungen** sind Anforderungen, die sich auf das gesamte Projekt beziehen, z.B. Lieferumfang, Liefervarianten, Lieferzeiten, Liefermengen und Transport.
- **Rechtliche und vertragliche Zusicherungen** sind Anforderungen, die die Abwicklung der Entwicklung betreffen, z.B. Kosten, Termin der Fertigstellung, Zahlungsmeilensteine, Vertragsstrafen, Umgang mit neuen Anforderungen, Umgang mit Anforderungsänderungen und Eskalationspfade.
- **Prozessbezogene Zusicherungen** sind Anforderungen bezüglich des Entwicklungsprozesses (Vorgehensmodell), z.B. ob die Entwicklung basierend auf dem V-Modell XT durchgeführt werden muss.

**Nichtfunktionale Anforderungen** reduzieren sich mit der obigen Gliederung auf Qualitätsmerkmale. **Qualitätsmerkmale** differenzieren sich in:

- Qualitätsmerkmale, welche **eine Funktionalität** anhand einer funktionalen Anforderung näher beschreiben (z.B. ein Zeitverbrauch). In diesem Fall ist klar,

was bei der Implementierung zu tun ist, um den Zeitverbrauch einzuhalten. Der Zeitverbrauch lässt sich einfach testen, sofern eine konkrete Zeitangabe bestehend aus Wert und Einheit in der Qualitätsanforderung spezifiziert ist.

→ **konkretes Qualitätsmerkmal**

- Qualitätsmerkmale, welche für die **gesamte Software** relevant sind (z.B. Wiederverwendbarkeit). In diesem Fall ist nicht eindeutig klar, was bei der Implementierung zu tun ist, um Wiederverwendbarkeit in der Software zu erreichen; Wiederverwendbarkeit ist nicht „einfach so“ testbar

→ **abstraktes Qualitätsmerkmal**

Dieses **Bewusstsein und Verständnis** der Qualitätsanforderungen sollte bei allen Mitarbeitern aus den Unternehmensbereichen Vertrieb, Marketing, Produktmanagement, Projektmanagement, Entwicklung, Test und Produktion vorhanden sein.

### **Fakten zu Qualität**

- Unterscheidung der Produkt- und Prozessqualität – beide beeinflussen sich gegenseitig.
- Unterscheidung der inneren (Entwicklersicht) und äußeren Qualität (Kundensicht)
- Qualität kann nicht später „hineingetestet“ werden.
- Qualitätsmerkmale müssen vor Entwicklungsbeginn definiert sein.
- Manche Qualitätsmerkmale und deren Maßnahmen zur Umsetzung unterstützen sich gegenseitig, z.B. beeinflussen sich Änderbarkeit und Wiederverwendbarkeit gegenseitig positiv.
- Manche Qualitätsmerkmale und deren Maßnahmen zur Umsetzung sind gegensätzlich, z.B. beeinflussen sich Performance und Zuverlässigkeit gegenseitig negativ.

***Richtlinie: Wählen Sie vier aus sechs Qualitätsmerkmale zur Realisierung aus.***

Stellen wir uns ein Beispiel einer Anwendung und ihre Umsetzung vor: Schaffen Sie es, einen Formel-1-Rennwagen so zu realisieren, dass er als Drei-Liter-Auto und zum Containertransport geeignet ist?

### **Relevante Standards**

Zu Software-Qualitätsmerkmalen sowie deren Spezifikation und Test existieren die folgenden **Standards**:

- ISO/IES 9126-1:2001 Software Engineering – Product Quality – Part 1: Quality Model

Dieser Standard ist abgelöst durch:

- ISO/IEC 25010:2011 Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) – System and Software Quality Models

Qualitätsmerkmal	Subqualitätsmerkmal
<b>Funktionalität</b>	Angemessenheit, Richtigkeit, Interoperabilität, Ordnungsmäßigkeit
<b>Zuverlässigkeit</b>	Reife, Fehlertoleranz, Wiederherstellbarkeit
<b>Benutzbarkeit</b>	Verständlichkeit, Erlernbarkeit, Bedienbarkeit
<b>Effizienz</b>	Zeitverhalten, Verbrauchsverhalten
<b>Änderbarkeit</b>	Analysierbarkeit, Modifizierbarkeit, Stabilität, Prüfbarkeit
<b>Übertragbarkeit</b>	Anpassbarkeit, Installierbarkeit, Konformität, Austauschbarkeit, Wiederverwendbarkeit
<b>Sicherheit</b>	Funktionale Sicherheit, Angriffssicherheit

*Bild 1: Exemplarische Übersicht zu Qualitäts- und Subqualitätsmerkmalen*

Bitte beachten Sie dazu immer die für Ihre Branche und Produkte gültigen Standards.

## Alles zu Requirements Engineering und Qualitätsanforderungen

Holen Sie sich im [MicroConsult Seminar zu Requirements Engineering und Management](#) das notwendige Wissen, um qualitativ hochwertige Anforderungen und dazu passende Abnahmekriterien zu entwickeln und zu dokumentieren. Die aktuell überarbeitete Version des Seminars fußt auf 10 Jahren Coaching- und Seminarerfahrung aus der Praxis.

**Autor: Thomas Batt**

Thomas Batt studierte nach seiner Ausbildung zum Radio- und Fernsehtechniker Nachrichtentechnik. Seit 1994 arbeitet er kontinuierlich in verschiedenen Branchen und Rollen im Bereich Embedded-/Realtime-Systementwicklung. 1999 wechselte Thomas Batt zur MicroConsult GmbH. Dort verantwortet er heute als zertifizierter Trainer und Coach die Themenbereiche Systems/Software Engineering für Embedded-/Realtime-Systeme sowie Entwicklungsprozess-Beratung.

**Wollen Sie sich auf den aktuellen Stand der Technik bringen?**

[MicroConsult Training & Coaching zum Thema Prozessmanagement](#)

[MicroConsult Training & Coaching zum Thema Requirements Engineering und Requirements Management](#)

[MicroConsult Fachwissen zum Thema Prozessmanagement](#)