



# **Softwaretechnik D**

## **Qualitätssicherung und Wartung**

Prof. Dr. Dirk Wiesmann  
Wintersemester 2024/2025

## Warum spielt Qualität von Software ein Rolle?

- Software dringt in fast alle Bereiche des täglichen Lebens vor (Bestandteil von Gebrauchsgütern) *ubiquitäres Computerzeitalter*
- Die Systemkomplexität steigt an
- Software wird in sicherheitskritischen Bereichen eingesetzt (Luftfahrt, Auto, Medizintechnik, Kraftwerke)
- Die spezielle Sicht von Informatikern, die in Deutschland tätig sind
  - Softwareentwicklung ist personalintensiv
  - Die Arbeitskosten sind in Deutschland vergleichsweise hoch
  - Dadurch entsteht ein Wettbewerbsnachteil
  - Man muss sich von den billigeren Mitbewerbern absetzen
  - Möglichkeit: Eine hohe Qualität bieten

*Erfordert qualifizierte Mitarbeiter*

*Software Made in Germany*

## Was sind Merkmale einer qualitativ hochwertigen Software?

Hohe Rechengenauigkeit

Gute Testbarkeit

Ausfallsicherheit

Korrektheit

leichte Bedienbarkeit

Einfachheit

Gute Dokumentation

...

Geringer Ressourcenverbrauch

gute Portierbarkeit

Spezifikationsvollständigkeit

## Wie entsteht Qualität?

„Von nichts kommt nichts“  
oder  
„Qualität entsteht nicht von alleine“

- Wir müssen etwas tun!
  - Was ist zu tun?
  - Wie ist es zu tun?
  - Wann ist es zu tun?
  - Wer hat es zu tun?

## Mit welchen Mitteln können wir qualitativ hochwertige Software erstellen?

Dokumentation      Integrationstest

Durchsicht      Zweigüberdeckung

Review      Blackbox-Test      Standards

Typisierung      Unit-Test      Richtlinien

**Konstruktive Maßnahmen**

Äquivalenzklassenbildung      Programmdokumentation      Grenzwertanalyse

Organisatorische Maßnahmen      **Schulungen**

Whitebox-Test      ...      Werkzeuge

Planung

Metriken      Lasttest      Prozessqualität

## Ziel

- Vermittlung der erforderlichen Kenntnisse, um bei der Softwareentwicklung ein definiertes Qualitätsniveau zu erzielen

Zielgerichtet, unter Einsatz der geeigneten Methoden, Prinzipien und Werkzeuge, das benötigte Qualitätsniveau erreichen

- Methodisches Vorgehen bei der Software-Wartung

## Weg

- Operationalisierung des Qualitätsbegriffs über Qualitätsmodelle
- Differenzierung in organisatorische, analytische und konstruktive Maßnahmen

## Themen

- 1) Qualität und Qualitätsmodelle
- 2) Typische Fehlerquellen
- 3) Bedeutung der Prozessqualität
- 4) Konstruktive Maßnahmen
- 5) Manuelle Prüfmethoden
- 6) Dynamische Prüfmethoden (Test)
- 7) Testautomatisierung
- 8) Testen objektorientierter Programme
- 9) Statische Prüfmethoden / Metriken
- 10) Werkzeuge / Build-Automatisierung
- 11) Wartung

## Disclaimer

- Die vorliegende Präsentation reicht **nicht** aus, um den Stoff der Vorlesung SWT D zu erlernen!
- Neben dem Durcharbeiten der Präsentation empfehle ich Ihnen
  - ① den aktiven Besuch der Vorlesung
  - ② die Lösung der Übungsaufgaben
  - ③ die Bearbeitung der Praktikumsaufgaben
  - ④ das Durcharbeiten von zusätzlicher Literatur
- Bitte bringen Sie zu den Vorlesungen, Übungen und Praktika Stift und Papier mit!
- Bitte bringen Sie zu den Übungen und Praktika Ihre Vorlesungsnotizen und die Folien mit!

## Voraussetzungen

- Qualitätssicherung und Wartung sind Bestandteil der Softwaretechnik
- Die Vorlesungen Softwaretechnik 1 und 2 werden als Basis vorausgesetzt
  - Analyse
  - Entwurf
- Zudem werden Erfahrungen in der Programmierung mit Java vorausgesetzt
  - Einführung in die Informatik
  - Programmierkurs 1, A

## Klausur

- Für die Prüfung relevante Inhalte
  - Vorlesung
    - Folien
    - Tafel
  - Übungen
  - Praktikum
- Aufgabentypen
  - Abfrage von Definitionen
  - Verständnisfragen
  - Anwendung (z.B. Herleitung von Testfällen, Aufdeckung von Qualitätsmängeln, ...)

## Literatur

[Bal98] H. Balzert. *Lehrbuch der Softwaretechnik, Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung*. Spektrum Akademischer Verlag, Heidelberg, 1998

[Bal08] H. Balzert. *Lehrbuch der Softwaretechnik, Softwaremanagement*. Spektrum Akademischer Verlag, Heidelberg, 2008

[Hof13] D.W. Hoffmann. *Software-Qualität*. Springer Vieweg, Berlin, 2013

[Lig09] P. Liggesmeyer. *Software-Qualität: Testen, Analysieren und Verifizieren von Software*. Spektrum, Heidelberg, 2009

[LL10] J. Ludewig, H. Lichter. *Software Engineering*. dpunkt.verlag, Heidelberg, 2010


[SL12] A. Spillner, T. Linz. *Basiswissen Softwaretest*. dpunkt.verlag, Heidelberg, 2012

[Wag13] S. Wagner. *Software Product Quality Control*. Springer, Berlin, 2013

## Software

- Was ist Software?

Programme, Abläufe, gegebenenfalls Dokumentation und Daten, die mit dem Betrieb eines Rechnersystems zu tun haben

- Damit ist Software mehr als nur ein ausführbares Programm
  - In einigen Aspekten unterscheidet sich Software von üblichen technischen Produkten
  - Diese Besonderheiten von Software haben Einfluss auf die Entwicklung von Software
- erschweren die Entwicklung fehlerfreier Software*
- 

- Spezielle Eigenschaften von Software:
  - ① Software ist ein immaterielles Produkt
  - ② Software benötigt einen Träger (Papier, „Silizium“)
  - ③ Software ist leicht änderbar
  - ④ Schwache Kausalität zwischen Änderung und Wirkung
  - ⑤ Ein Programm realisiert keine stetige Funktion
  - ⑥ Software ist schwer zu vermessen
  - ⑦ Software wird nicht gefertigt, sondern nur entwickelt
  - ⑧ Software verschleißt nicht
  - ⑨ Software altert über die Umgebung

Was ist

# Qualität

## Qualität

*Gesamtheit von Eigenschaften und Merkmalen eines Produktes oder einer Tätigkeit, die sich auf die Eignung zur Erfüllung gegebener Erfordernisse beziehen.*

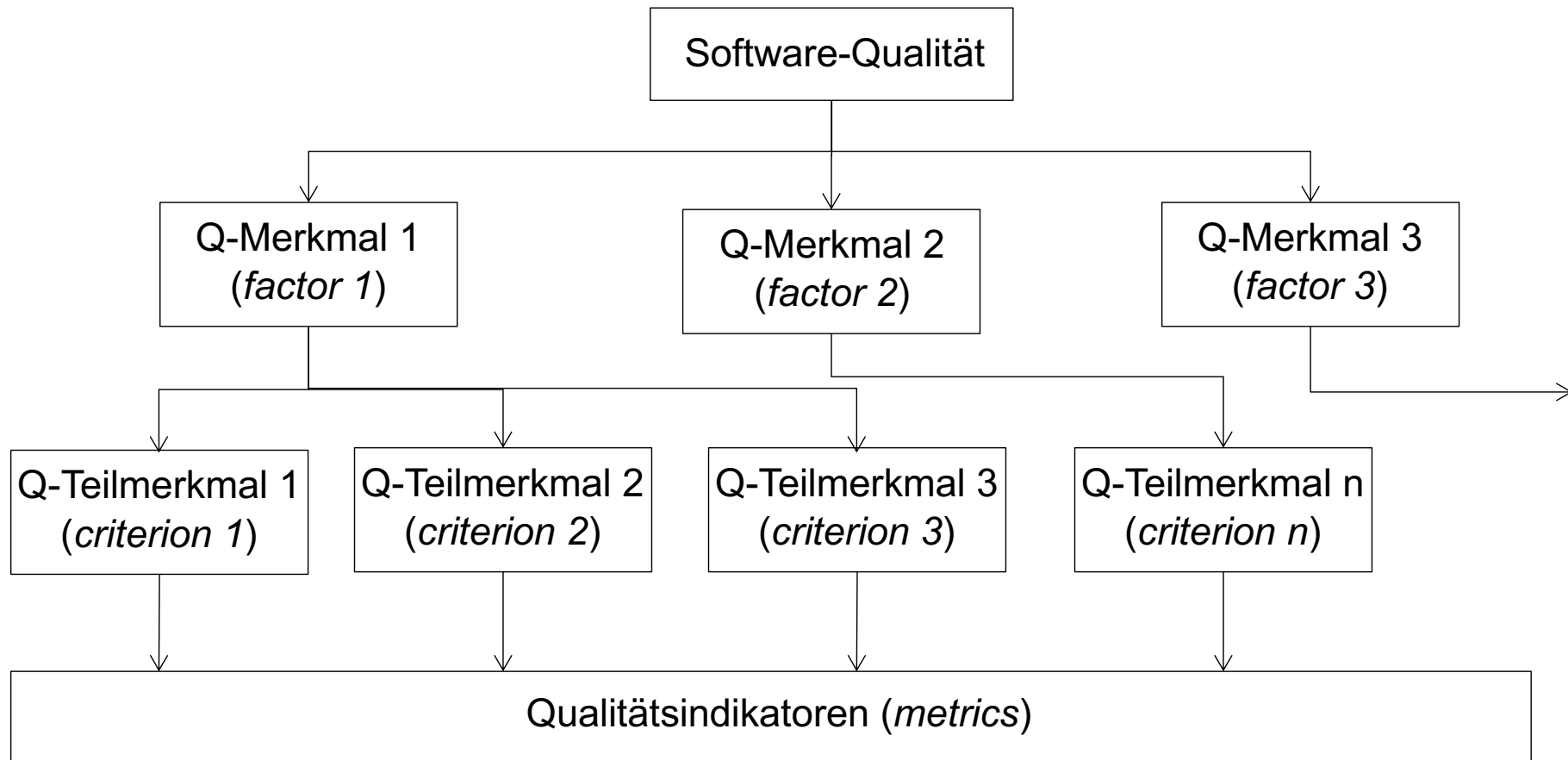
aus DIN 55350 – 11:1995-08

- Der Begriff Qualität ist
  - vielfältig
  - relativ
  
- In der Praxis
  - ist eine allgemeine Qualitätsdefinition nicht ausreichend
  - müssen die relevanten Qualitätsmerkmale identifiziert und benannt werden

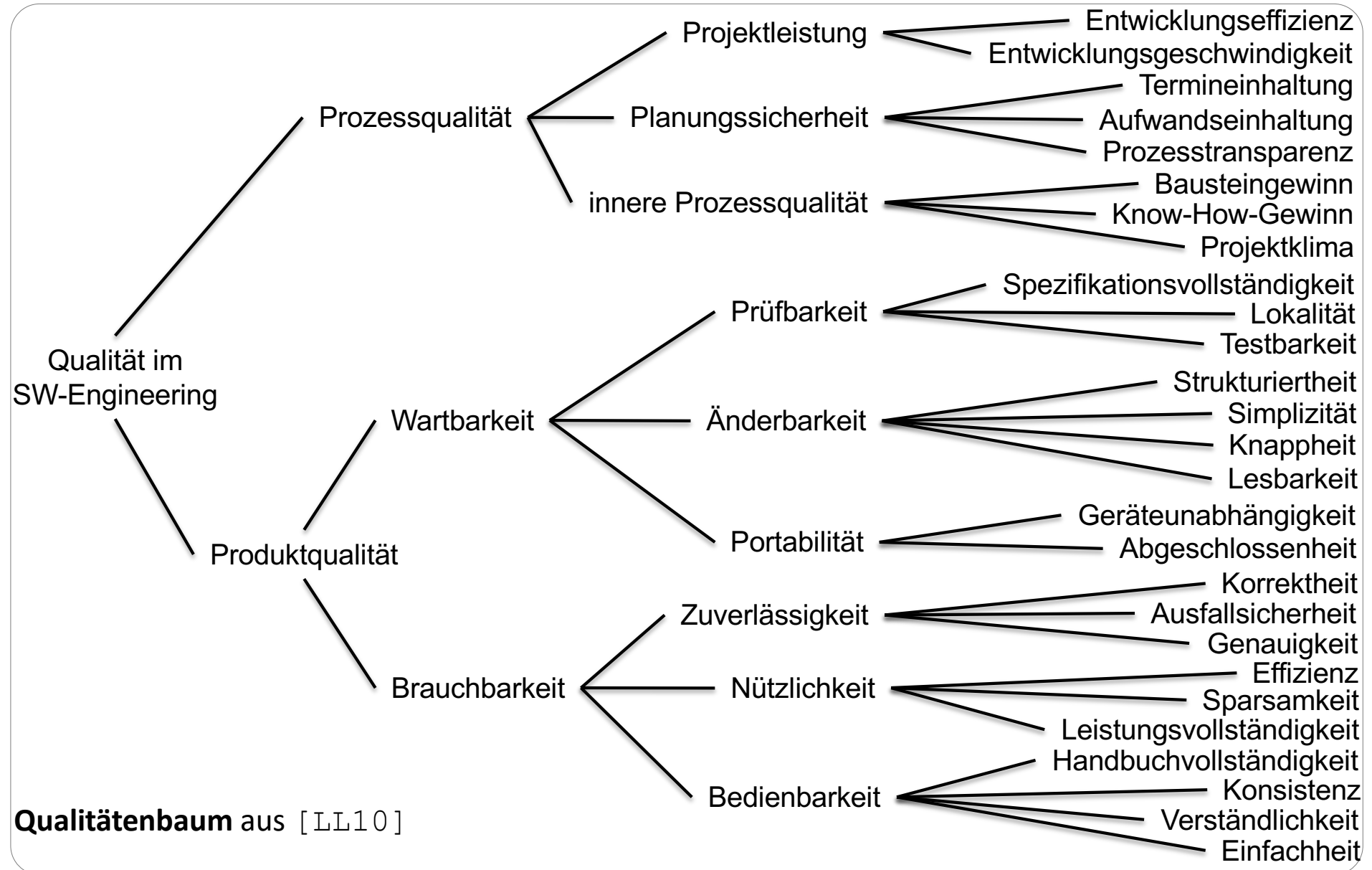
## Qualitätsmodell

- Man erstellt ein Qualitätsmodell, indem ausgehend vom allgemeinen Qualitätsbegriff Unterbegriffe (Qualitätsmerkmale, *factors*, *characteristics*) abgeleitet werden
- Diese Qualitätsmerkmale werden in einem weiteren Schritt in Teilmerkmale (atomare Qualitätsmerkmale, *criteria*, *subcharacteristics*) verfeinert
- Das Ziel ist die Messbarkeit/Überprüfbarkeit von atomaren Qualitätsmerkmalen durch Qualitätsindikatoren (Attributen)

- Schematischer Aufbau eines FCM-Qualitätsmodells  
(*factor-criteria-metrics-models*)

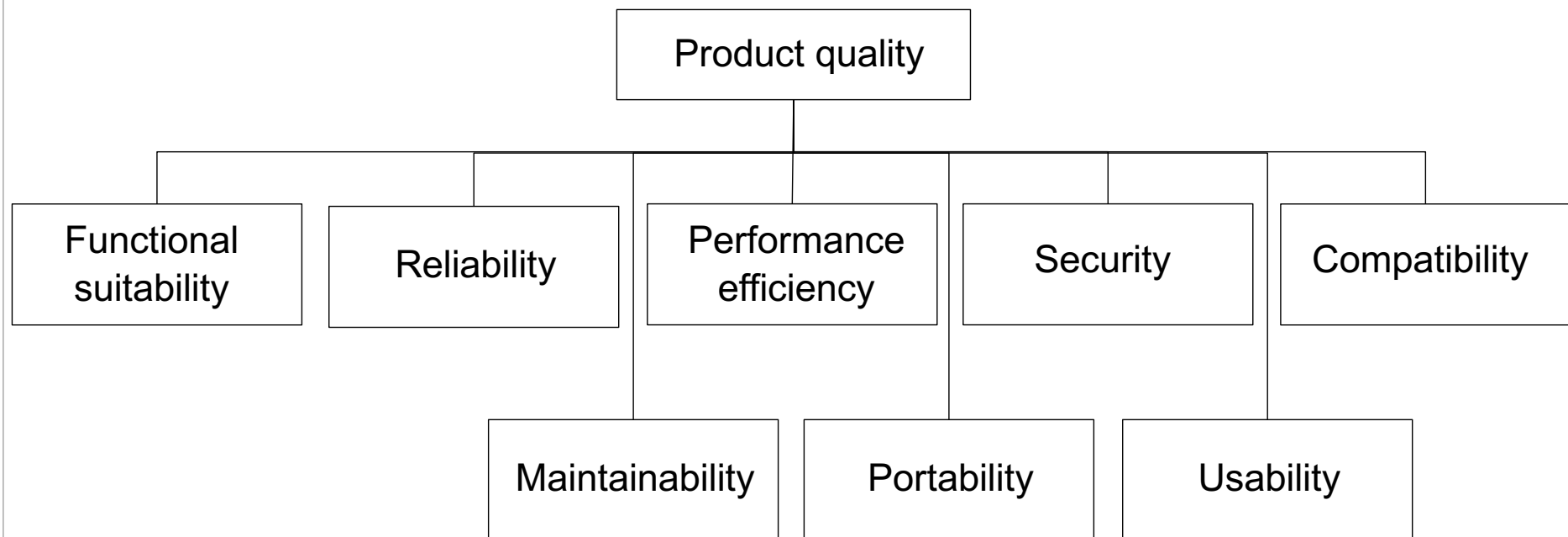


- Im Software-Engineering gibt es zwei Sichtweisen auf die Qualität:
  - Produktqualität
  - Prozessqualität (siehe Bewertung von Prozessmodellen)
- Historisch sind verschiedene FCM-Qualitätsmodelle entstanden
  - Qualitätenbaum nach Boehm, Brown und Lipow (1976) aus [LL10]
  - ISO/IEC 9126
  - ISO/IEC 250nn (ab 2005 Nachfolge von 9126)



- Die Norm ISO/ICE 25010 betrachtet nur die Produktqualität
- Hier werden zwei Qualitätsmodelle definiert
  - *Product quality*
  - *Quality in use*
- Definitionen aus ISO/IEC FCD 25010 aus 2010 (*final draft*)

- Das Qualitätsmodell *Product quality* gliedert sich in 8 Qualitätsmerkmale



- Functional suitability (funktionale Eignung)  
*the degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions*
- Reliability (Ausfallsicherheit)  
*the degree to which a system or component performs specified functions under specified conditions for a specified period of time*
- Performance efficiency (Leistungseffizienz)  
*the performance relative to the amount of resources used under stated conditions*
- Security (Sicherheit)  
*the degree of protection of information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them*

- Compatibility (Kompatibilität)

*the degree to which a product, system or component can exchange information with other products, systems or components, and/or perform its required functions, while sharing the same hardware or software environment*

- Maintainability (Wartbarkeit)

*the degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers*

- Portability (Portabilität)

*the degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another*

– Das Qualitätsmodell *Product quality* in der Übersicht

Qualitätsmerkmal	Teilmerkmale
Functional suitability	Functional appropriateness Functional correctness Functional completeness
Reliability	Maturity Availability Fault tolerance Recoverability
Performance efficiency	Time behaviour Resource utilisation Capacity

Qualitätsmerkmal	Teilmerkmale
Security	Confidentiality Integrity Non-repudiation Accountability Authenticity
Compatibility	Co-existence Interoperability
Maintainability	Modularity Reusability Analysability Modifiability Testability

Qualitätsmerkmal	Teilmerkmale
Portability	Adaptability Installability Replaceability
Usability	Appropriateness recognizability Learnability Operability User error protection User interface aesthetics Accessibility

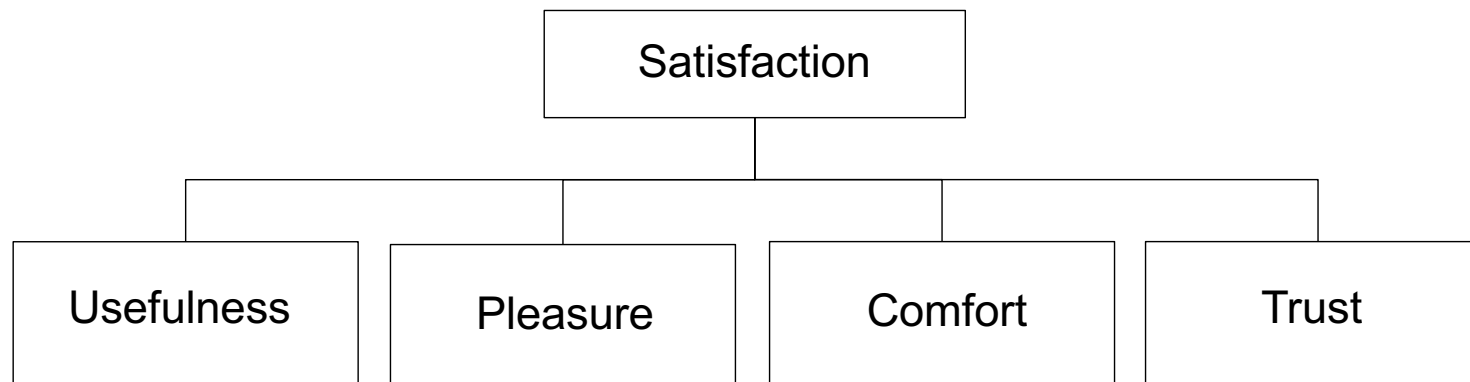
– *Quality in Use:*

Zu welchem Grad kann ein spezifischer Nutzer spezifische Ziele unter Berücksichtigung von Effektivität, Effizienz, Zufriedenheit, Sicherheit und Brauchbarkeit erreichen (bezogen auf einen spezifischen Anwendungskontext)



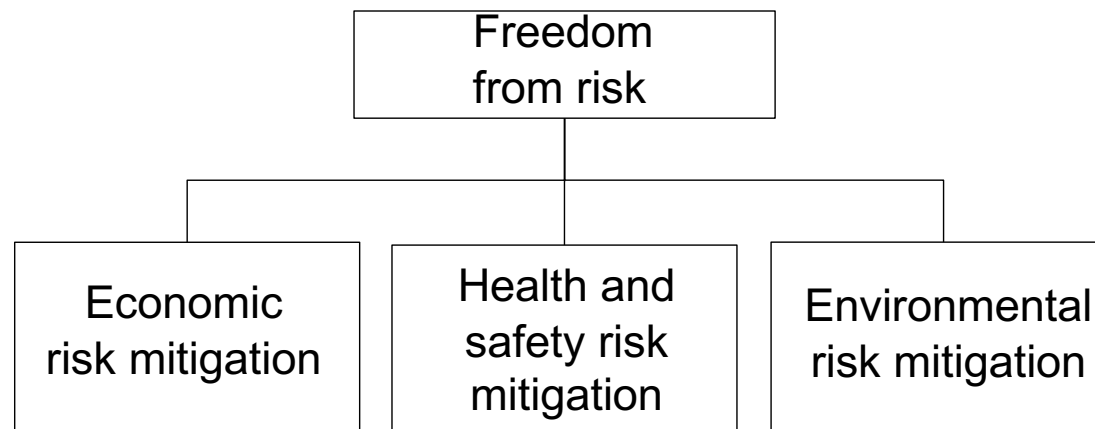
- Effectiveness (Effektivität)  
*the accuracy and completeness with which users achieve specified goals*
- Efficiency (Effizienz)  
*the resources expended in relation to the accuracy and completeness with which users achieve goals*
- Satisfaction (Zufriedenstellung)  
*the degree to which users are satisfied with the experience of using a product in a specified context of use*
- Freedom from risk (Risikofreiheit)  
*the degree to which a product or system mitigates the potential risk to economic status, human life, health, or the environment*

- Context coverage (Nutzbarkeit, Brauchbarkeit)  
*the degree to which a product or system can be used with effectiveness, efficiency, freedom from risk and satisfaction in both specified contexts of use and in contexts beyond those initially explicitly identified*
- Das Qualitätsmerkmal Zufriedenstellung (*characteristic satisfaction*) wird in vier Teilmerkmale unterteilt

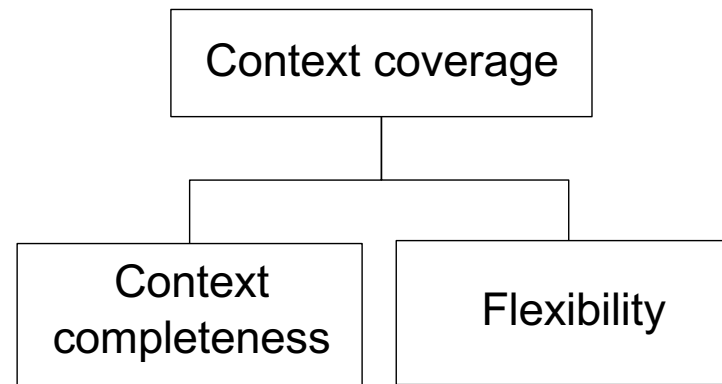


- Usefulness (Nützlichkeit)  
*the degree to which a user is satisfied with their perceived achievement of pragmatic goals, including the results of use and the consequences of use*
- Pleasure (Freude, emotionale Zufriedenheit)  
*the degree to which a user obtain pleasure from fulfilling their personal needs*
- Comfort (Komfort)  
*the degree to which the user is satisfied with physical comfort*
- Trust (Vertrauen)  
*the degree to which a user or other stakeholder has confidence that a product or system will behave as intended*

- Das Qualitätsmerkmal Risikofreiheit (*freedom from risk*) wird in drei Teilmerkmale unterteilt



- Das Qualitätsmerkmal Kontextüberdeckung (*context coverage*) wird in zwei Teilmerkmale unterteilt



- Qualitätsbewertung ist ein Soll-Ist-Vergleich
  - Soll-Wert: Vorgaben für ein Qualitätsmerkmal, abgeleitet aus den Erfordernissen
  - Ist-Wert: Tatsächliche Ausprägung des Qualitätsmerkmals
- Eine Abweichung zwischen Soll- und Ist-Wert bezeichnen wir als Fehler



*Vereinfachte Definition*

*Eine Unterscheidung zwischen  
Fehlverhalten und Fehler ist z.B. möglich*

- Zu jedem Teilqualitätsmerkmal wird ein geeigneter Qualitätsindikator (Metrik) benötigt *wird nicht zwingend durch ein Qualitätsmodell vorgegeben*
  - Vorgabe eines zu erreichenden Ziels (Soll-Wert)
  - Soll-Ist-Vergleich (Qualitätsmessung)
- Beispiele
  - Satisfaction (Zufriedenstellung)
    - Bewertung des Systems durch (Test-)Anwender bezüglich einer Ordinalskala (sehr zufrieden, zufrieden, unzufrieden, sehr unzufrieden)
    - Vorgabe eines Soll-Wertes durch eine Häufigkeitsverteilung
    - Soll-Ist-Vergleich z.B. über Median
  - Portability
    - $1 - (\text{Umgebungsabhängige Module} / \text{Module})$
  - Efficiency
    - $1 - (\text{Ist-Ressourcenverbrauch} / \text{Allokierte Ressourcen})$

*Für ein Teilqualitätsmerkmal kann es unterschiedliche Metriken geben*