

Willkommen zur Vorlesung  
*Methodische Grundlagen  
des Software-Engineering*  
im Sommersemester 2012  
Prof. Dr. Jan Jürjens

TU Dortmund, Fakultät Informatik, Lehrstuhl XIV

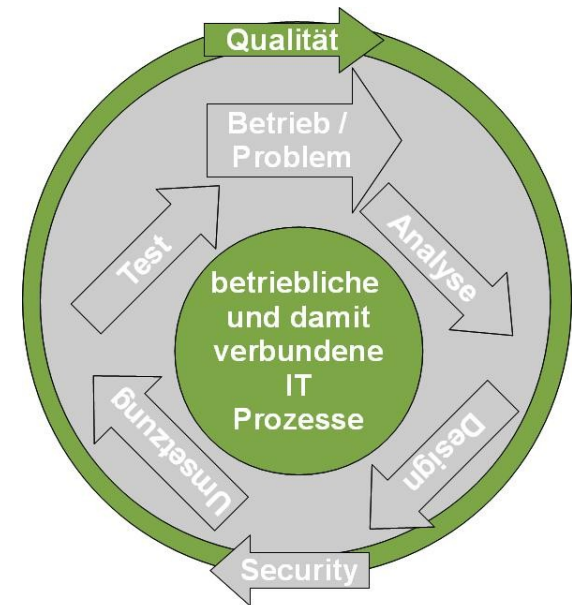
## 3.0 Qualitätsmanagement: Grundlagen

[inkl. Beiträge von Prof. Sommerville, St. Andrews University  
und Prof. Martin Glinz, Universität Zürich]

# Einordnung

## Qualitätsmanagement: Grundlagen

- Anwendungsbeispiel Finanz- und Versicherungsdomäne
- Geschäfts-Prozesse
- **Qualitätsmanagement**
  - **Grundlagen**
  - Prozessqualität
  - Softwaremetriken
- Testen
- Sicherheit
- Sicheres Software Design



- Software-Qualität
  - Was ist Qualität?
  - Qualitätsmerkmale
  - Qualitätsmanagement
- Software-Standards




## Qualität:

- **Hochwertige** Produkte oder Arbeitsweise.
- Langlebige Produkte **hoher Güte**.

- **Qualität (quality):** der Grad, in dem ein Satz **inhärenter Merkmale Anforderungen** erfüllt. [ISO 9000:2000, vgl. Vorlesung Abschnitt 3.1]
- **Inhärentes Merkmal (inherent characteristic):** eine **kennzeichnende Eigenschaft** einer Einheit (Produkt, Dienstleistung, Prozess, System, Person, Organisation, etc.), welche diese aus sich selbst heraus hat und die ihr nicht explizit zugeordnet ist.
- „**Qualität** ist die Gesamtheit von Eigenschaften und Merkmalen eines Produkts oder einer Tätigkeit, die sich auf deren Eignung zur Erfüllung gegebener Erfordernisse bezieht.“ (DIN 55350 Teil 11)

Beispiel:  
Ein Medikament



- **Inhärentes Merkmal:** Zusammensetzung
- **Explizit zugeordnet:** Preis

- Qualität ist **Zielerfüllung**. Die Ziele (Anforderungen) können **explizit** festgelegt oder **implizit** durch gemeinsame Vorstellungen der Beteiligten gegeben sein.
- Eine Auffassung von Qualität als reine **Zweckeignung** oder **Kundenzufriedenheit** greift zu kurz. Sie erfasst den Qualitätsbegriff nicht in seiner Gesamtheit.
- Qualität ist kein absolutes Maß für die **Güte** einer Einheit.
- Qualität entsteht nicht von selbst. Sie muss **definiert** und **geschaffen** werden.
- Qualität bezieht sich sowohl auf **Produkte** als auch auf **Prozesse** und **Projekte**.

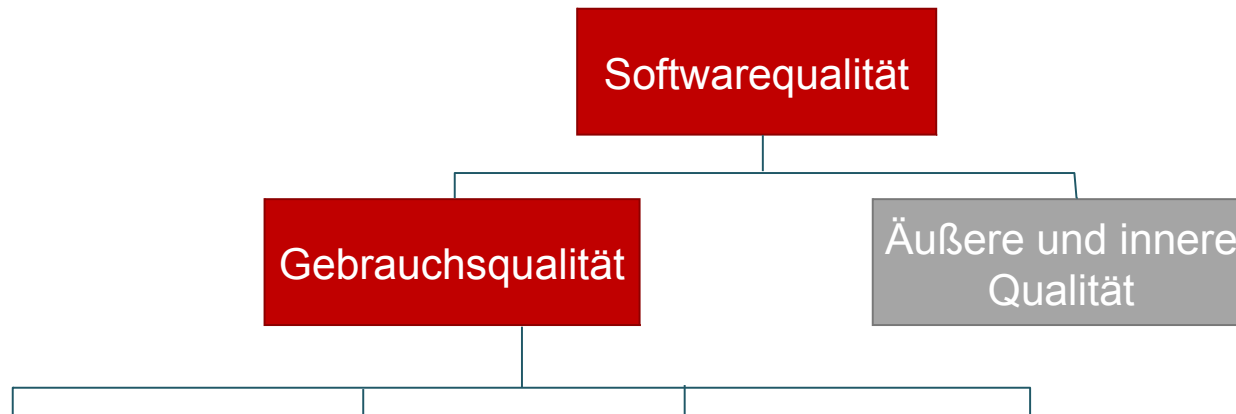
- Der Grad, in dem ein System, eine Komponente oder ein Prozess die Kundenerwartungen und -bedürfnisse erfüllt.  
[nach IEEE 610: „Standard Glossary of Software Engineering Terminology“]
- Qualitätsanforderungen geben vor, welche Qualitätsmerkmale das Produkt in welcher Güte aufweisen soll (Qualitätsniveau).
- Gesamtheit aller Qualitätsmerkmale und deren geforderte Ausprägung.
- Qualitätsmerkmale beziehen sich auf Anforderungen:
  - Funktionale Anforderungen  
(Fachlichkeit, Funktionen, Schnittstellen, ...)
  - Nicht-funktionale Anforderungen  
(Qualitäts- und Realisierungsanforderungen, Projektspezifische Anforderungen, ...)



- Qualität bedeutet vereinfacht, dass ein Produkt seine Spezifikation erfüllen sollte.
- Dies ist problematisch für Software-Systeme:
  - Es gibt eine Spannung zwischen Qualitätsanforderungen des Kunden (Effizienz, Zuverlässigkeit, etc.) und Qualitätsanforderungen des Entwicklers (Wartbarkeit, Wiederverwendbarkeit, etc.).
  - Einige Qualitätsanforderungen sind schwierig, in eindeutiger Weise festzulegen.
  - Software-Spezifikationen sind meist unvollständig und oft widersprüchlich.
- Der Fokus könnte "Zwecktauglichkeit" statt Spezifikations-Konformität sein.

- Kann die Software ausreichend zuverlässig in Betrieb genommen werden ?
- Ist die Leistungsfähigkeit der Software akzeptabel für den normalen Gebrauch ?
- Ist die Software nutzbar ?
- Ist die Software gut strukturiert und verständlich ?

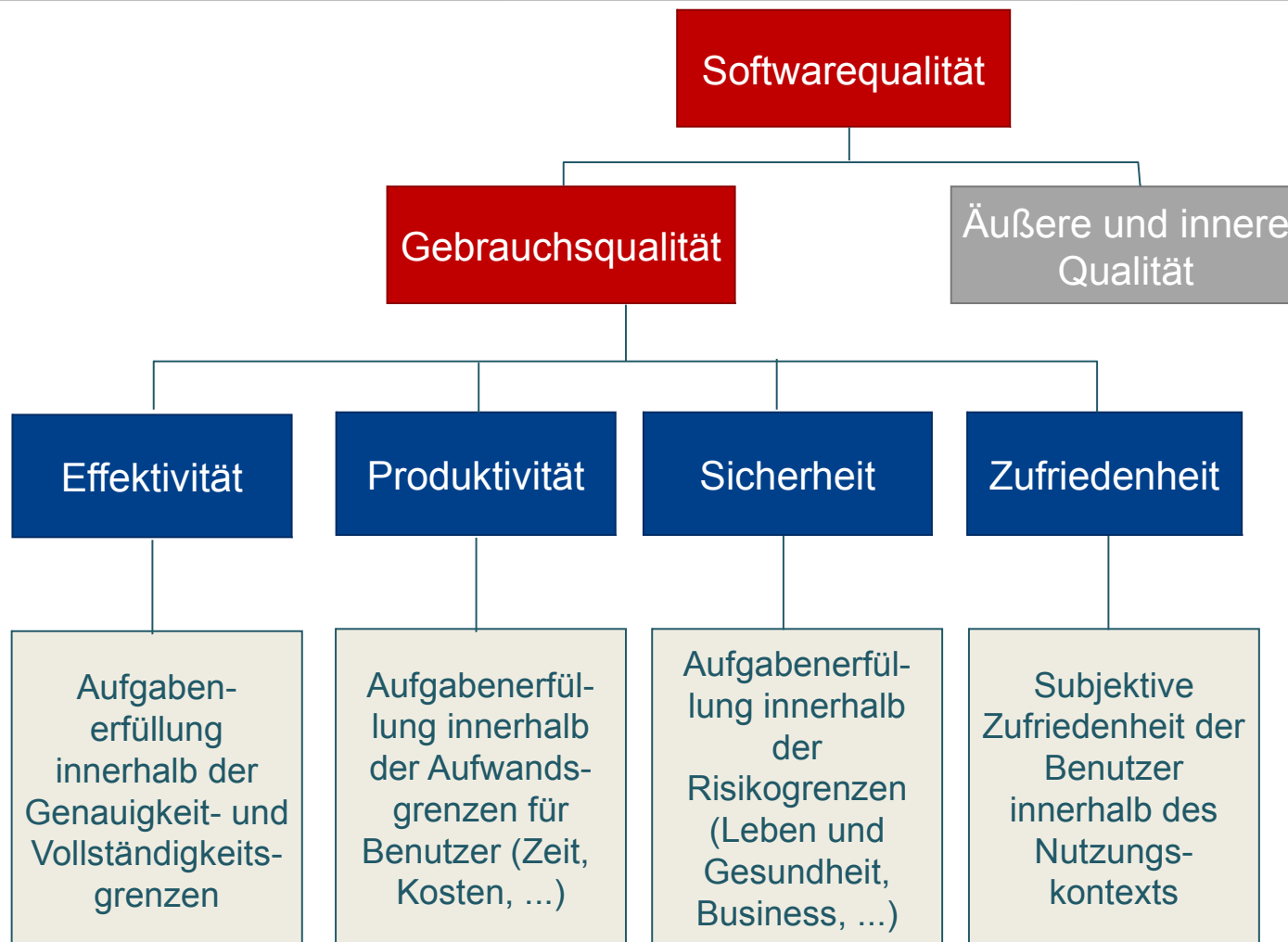
# Software-Qualität nach ISO/IEC 9126 (1)



Welche Software-Qualitätseigenschaften kennen Sie ?  
(Wiederholung von SWT / SWK)

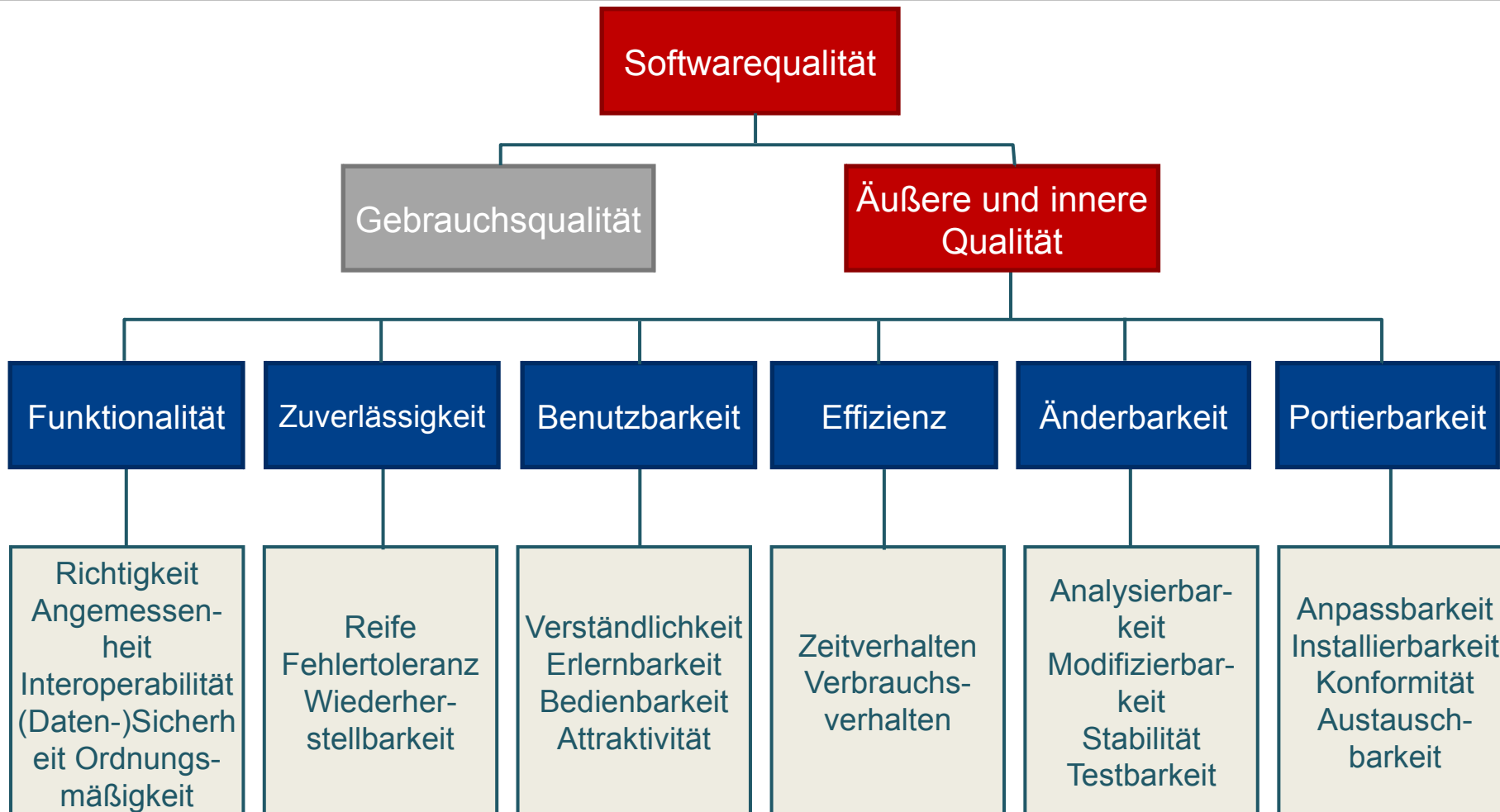
ISO/IEC 9126: Bewerten von Softwareprodukten, Qualitätsmerkmale und Leitfaden zu ihrer Verwendung. (Seit 2005 ersetzt durch ISO/IEC 25000 Software engineering – Software product Quality Requirements and Evaluation (SQuaRE).)

# Software-Qualität nach ISO/IEC 9126 (1)



ISO/IEC 9126: Bewerten von Softwareprodukten, Qualitätsmerkmale und Leitfaden zu ihrer Verwendung. (Seit 2005 ersetzt durch ISO/IEC 25000 Software engineering – Software product Quality Requirements and Evaluation (SQuaRE).)

# Software-Qualität nach ISO/IEC 9126 (2)



ISO/IEC 9126: Bewerten von Softwareprodukten, Qualitätsmerkmale und Leitfaden zu ihrer Verwendung. (Seit 2005 ersetzt durch ISO/IEC 25000 Software engineering – Software product Quality Requirements and Evaluation (SQuaRE).)

Vorhandensein von Funktionen mit festgelegten Eigenschaften. Diese Funktionen erfüllen die festgelegten oder vorausgesetzten Anforderungen.

## Angemessenheit

- Merkmale von Software, die sich auf das Vorhandensein und die Eignung einer Menge von Funktionen für spezifizierte Aufgaben beziehen.
- Die Fähigkeit eines Softwareprodukts für spezifizierte Aufgaben und Zielsetzungen der Benutzer einen geeigneten Satz Funktionen zu liefern.

## Richtigkeit

- Merkmale von Software, die sich auf das Liefern der richtigen oder vereinbarten Ergebnisse oder Wirkungen beziehen.
- Die Fähigkeit eines Softwareprodukts, die richtigen oder vereinbarten Ergebnisse oder Wirkungen mit dem benötigten Grad an Genauigkeit zu liefern.

## Interoperabilität

- Merkmale von Software, die sich auf ihre Eignung beziehen, mit vorgegebenen Systemen zusammenzuwirken.
- Die Fähigkeit eines Softwareprodukts, mit einer oder mehreren spezifizierten Komponenten zusammenzuwirken.

## Ordnungsmäßigkeit

- Merkmale von Software, die bewirken, dass die Software anwendungsspezifische Normen oder Vereinbarungen oder gesetzliche Bestimmungen und ähnliche Vorschriften erfüllt.

## Sicherheit

- Merkmale von Software, die sich auf ihre Eignung beziehen, unberechtigten Zugriff, sowohl versehentlich als auch vorsätzlich, auf Programme und Daten zu verhindern.
- Eigenschaften der Software, die sich auf die Fähigkeit beziehen, nicht autorisierte Zugriffe auf Programme oder Daten zu verhindern, unabhängig davon, ob diese versehentlich oder vorsätzlich erfolgen.

# Äußeres Qualitätsmerkmal

## Zuverlässigkeit

Merkmale, die sich auf die Fähigkeit der Software beziehen, ihr Leistungsniveau unter festgelegten Bedingungen über einen festgelegten Zeitraum zu bewahren.

### Reife

- Merkmale von Software, die sich auf die Häufigkeit von Versagen durch Fehlzustände in der Software beziehen.
- Die Fähigkeit eines Softwareprodukts, Fehlerwirkungen aufgrund von Fehlerzuständen in der Software zu vermeiden.

### Fehlertoleranz

- Merkmale von Software, die sich auf ihre Eignung beziehen, ein spezifiziertes Leistungsniveau bei Software-Fehlern oder Nicht-Einhaltung ihrer spezifizierten Schnittstelle zu bewahren.
- Die Fähigkeit eines Softwareprodukts, ein spezifiziertes Leistungsniveau auch bei Fehlfunktionen oder trotz Fehleingaben (z.B. falsche Bedienung) aufrecht zu erhalten.

### Wiederherstellbarkeit

- Merkmale von Software, die sich beziehen auf die Möglichkeit, bei einem Versagen ihr Leistungsniveau wiederherzustellen und die direkt betroffenen Daten wiederzugewinnen und auf die dafür benötigte Zeit und den benötigten Aufwand.
- Die Fähigkeit eines Softwareprodukts, bei einer Fehlerwirkung das spezifizierte Leistungsniveau des Systems wiederherzustellen und die direkt betroffenen Daten wiederzugewinnen.



Merkmale, die sich auf den zur Benutzung erforderlichen Aufwand beziehen, und auf die individuelle Bewertung einer solchen Benutzung durch eine festgelegte oder vorausgesetzte Gruppe von Benutzern.

### **Verständlichkeit**

- Merkmale von Software, die sich auf den Aufwand für den Benutzer beziehen, das Konzept und die Anwendung zu verstehen.
- Die Fähigkeit eines Softwareprodukts, den Benutzer in die Lage zu versetzen zu verstehen, ob die Software geeignet ist, und wie sie für eine bestimmte Aufgabe und Benutzungsbedingungen brauchbar ist.

### **Erlernbarkeit**

- Merkmale von Software, die sich auf den Aufwand für den Benutzer beziehen, die Anwendung zu erlernen.
- Die Fähigkeit eines Softwareprodukts, einem Benutzer das Erlernen der Anwendung leicht zu machen.

### **Bedienbarkeit**

- Merkmale von Software, die sich auf den Aufwand für den Benutzer bei der Bedienung und Ablaufsteuerung beziehen.

Merkmale, die sich auf das Verhältnis zwischen dem Leistungsniveau der Software und dem Umfang der eingesetzten Betriebsmittel unter festgelegten Bedingungen beziehen.

- **Zeitverhalten**

Merkmale von Software, die sich beziehen auf die Antwort- und Verarbeitungszeiten und auf den Durchsatz bei der Ausführung ihrer Funktionen.

- **Verbrauchsverhalten**

Merkmale von Software, die sich darauf beziehen, wie viele Betriebsmittel bei der Erfüllung ihrer Funktionen benötigt werden und wie lange.

Merkmale, die sich auf den Aufwand für Durchführung vorgegebener Änderungen beziehen.

## **Analysierbarkeit**

- Merkmale von Software, die sich auf den Aufwand beziehen, der notwendig ist, um Mängel oder Ursachen von Versagen zu diagnostizieren oder um änderungsbedürftige Teile zu bestimmen.
- Die Fähigkeit eines Softwareprodukts, die Diagnose von Mängeln oder Ursachen von Fehlerwirkungen zu ermöglichen oder änderungsbedürftige Teile zu bestimmen.

## **Modifizierbarkeit**

- Merkmale von Software, die sich auf den Aufwand beziehen, der zur Ausführung von Verbesserungen, zur Fehlerbeseitigung oder zur Anpassung an Umgebungsänderungen notwendig ist.
- Die Fähigkeit eines Softwareprodukts, die Durchführung spezifizierter Änderungen zu ermöglichen.

## **Stabilität**

- Merkmale von Software bezüglich des Risikos unerwarteter Wirkungen von Änderungen.
- Fähigkeit eines Softwareprodukts, unerwartete Auswirkungen von Änderungen zu vermeiden.

## **Testbarkeit**

- Merkmale von Software, die sich auf den Aufwand beziehen, der zum Testen der geänderten Software notwendig ist.
- Die Fähigkeit eines Softwareprodukts für einen Test nach einer Änderung.

Merkmale, die sich auf die Eignung der Software beziehen, von einer Umgebung in eine andere übertragen zu werden.

## **Anpassbarkeit**

- Merkmale von Software, die sich auf die Möglichkeit beziehen, sie an verschiedene festgelegte Umgebungen anzupassen, wenn nur Schritte unternommen oder Mittel eingesetzt werden, die für diesen Zweck für die betrachtete Software vorgesehen sind.
- Die Fähigkeit eines Softwareprodukts, dass sie auf verschiedene Laufzeitumgebungen angepasst werden kann und dabei nur die Anpassungen vorzunehmen sind, die genau diesem Zweck dienen.

## **Installierbarkeit**

- Merkmale von Software, die sich auf den Aufwand beziehen, der zur Installierung der Software in einer festgelegten Umgebung notwendig ist.
- Die Fähigkeit eines Softwareprodukts, in einer spezifizierten Umgebung installierbar zu sein.

## Konformität

- Merkmale von Software, die bewirken, dass die Software Normen oder Vereinbarungen zur Übertragbarkeit erfüllt.
- Die Fähigkeit eines Softwareprodukts, anwendungsspezifische Normen oder Vereinbarungen oder gesetzliche Bestimmungen und ähnliche Vorschriften zu erfüllen.

## Austauschbarkeit

- Merkmale von Software, die sich beziehen auf die Möglichkeit, diese anstelle einer spezifizierten anderen Software in der Umgebung jener Software zu verwenden, und auf den dafür notwendigen Aufwand.
- Die Fähigkeit eines Softwareprodukts an Stelle einer anderen spezifizierten Software zum selben Zweck in der gleichen Umgebung genutzt zu werden.

- Nicht alle Qualitätsmerkmale lassen sich gleichzeitig gleich gut erfüllen.
- Ein Qualitätsplan sollte deshalb für die zu entwickelnde Software die wichtigsten Qualitätsmerkmale herausstellen.
- Prioritäten festlegen
  - In engster Absprache mit Auftraggebern und Anwendern.
  - Qualitätsanforderungen sind Bestandteil der nicht-funktionalen Anforderungen im Pflichtenheft.
- Der Qualitätsplan sollte auch vorgeben, wie die Qualitätsbewertung der Software ablaufen soll.

Welche Qualitätsmerkmale können Ihrer Ansicht nach im Konflikt zueinander stehen ?

Effektivität

Produktivität

Sicherheit

Zufriedenheit

Funktionalität

Zuverlässigkeit

Benutzbarkeit

Effizienz

Änderbarkeit

Portierbarkeit

Welche Qualitätsmerkmale können Ihrer Ansicht nach im Konflikt zueinander stehen ?

**Antwort:**

Zum Beispiel:

Zuverlässigkeit ↔ Effizienz

Sicherheit ↔ Benutzbarkeit

Sicherheit ↔ Effizienz

Änderbarkeit / Wartbarkeit ↔ Effizienz



- Software-Qualität
  - Was ist Qualität?
  - Qualitätsmerkmale
  - Qualitätsmanagement
- Software-Standards

Stellt sicher, dass die geforderte Qualität des Software-Produkts erreicht wird.

Drei Hauptanliegen:

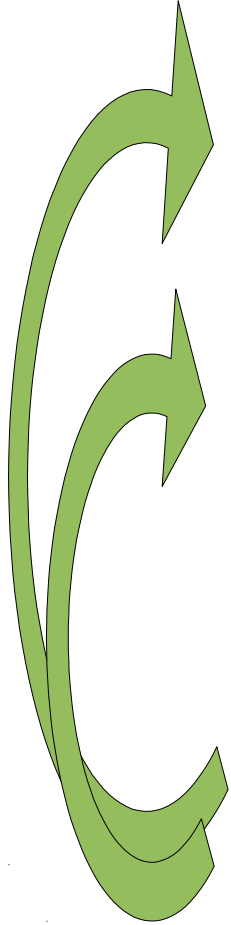
- Auf organisatorischer Ebene beschäftigt sich das Qualitätsmanagement mit Schaffung eines Rahmens für die **organisatorischen Prozesse und Standards**, welche zu qualitativ hochwertiger Software führen.
- Auf der Projektebene beinhaltet Qualitätsmanagement die Anwendung bestimmter **Qualitäts-Prozesse** und die Überprüfung, ob die geplanten Prozesse befolgt wurden.
- Auf Projektebene beinhaltet das Qualitätsmanagement auch die Festlegung eines **Qualitätsmanagement-Plans** für ein Projekt. Der Qualitätsmanagement-Plan sollte eine Menge von Qualitätszielen für das Projekt definieren und festlegen, welche Prozesse und Standards zu verwenden sind.

- Qualitätsmanagement ist besonders für große und komplexe Systeme wichtig. Die Qualitäts-Dokumentation ist eine Aufzeichnung der Fortschritte und unterstützt die Kontinuität der Entwicklung, als auch die Entwicklerteam-Änderungen.
- Für kleinere Systeme muss Qualitätsmanagement weniger Dokumentation enthalten und sollte sich auf die Einführung einer Qualitätsmanagements-Kultur konzentrieren.

- Qualitätsmanagement bietet eine unabhängige Überprüfung im Software-Entwicklungsprozess.
- Der Qualitätsmanagement-Prozess prüft die Projektergebnisse, um zu gewährleisten, dass sie im Einklang mit den organisatorischen Standards und Zielen sind.
- Das Qualitäts-Team sollte vom Entwicklungsteam unabhängig sein, sodass es ein objektives Bild von der Software erhält. Das erlaubt ihm, ohne Beeinflussung durch die Software-Entwicklung über die Software-Qualität zu berichten.

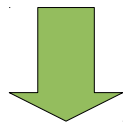
## Generell: planen – lenken – verbessern

Qualitätssicherung



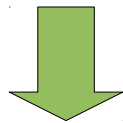
### Qualitätsplanung

Definition der Qualitätsziele: Das wollen wir erreichen !



### Qualitätslenkung

Konstruktiv: so müssen wir arbeiten !



### Qualitätsprüfung/-verbesserung

Analytisch: haben wir richtig gearbeitet ?

- Wie gut funktioniert was ?
- Wo gibt es Schwachstellen ?

Das wollen wir verbessern !

- **Qualitätsplanung (quality planning)** – Teil des Qualitätsmanagements, der auf das Festlegen der Qualitätsziele und der notwendigen Ausführungsprozesse sowie der zugehörigen Ressourcen zur Erfüllung der Qualitätsziele gerichtet ist.  
[ISO 9000:2000]

=> Qualitätsziele bestimmen

=> Kein Qualitätsmanagement ohne eine saubere, quantifizierte Spezifikation der Anforderungen.

- **Qualitätsplanung** heißt:
  - Im Allgemeinen: Aufbau und Dokumentation des QM-Systems, allgemeine Qualitätsziele
  - Im Speziellen: Festlegung der Qualitätsziele für individuelle Projekte



- Ein Qualitäts-Plan legt die gewünschten Produktqualitäten und deren Bewertung fest und definiert die wichtigsten Qualitätsmerkmale.
- Der Qualitäts-Plan sollte den Qualitätsbeurteilungsprozess definieren.
- Es sollte festgelegt werden, welche organisatorischen Standards angewendet werden und, wenn nötig, definieren, welche neuen Standards verwendet werden.

## Qualitäts-Plan-Struktur:

- Produkt-Einführung;
- Produkt-Pläne;
- Prozessbeschreibungen;
- Qualitätsziele;
- Risiken und Risikomanagement.

Qualitäts-Pläne sollten kurze und prägnante Dokumente sein.

- Wenn sie zu lang sind, wird sie niemand lesen.



# Qualitätslenkung: Konstruktive Maßnahmen

- Fehlerverhindernde / fehlervermeidende Prozesse definieren.
- Prüf- und Korrekturverfahren in die Prozesse integrieren.
- Prüfergebnisse zur Verbesserung des Prozesses verwenden.
- Eine systematische, ingenieurmäßige Vorgehensweise, welche die Erreichung gegebener Qualitätsanforderungen garantiert, gibt es für Software bis heute nicht.
- Konstruktive Maßnahmen werden so weit als möglich eingesetzt, um das generelle Qualitätsniveau zu heben.
- Rigorose Qualitätsprüfung (und Behebung der festgestellten Mängel) während aller Phasen der Entwicklung ist heute das Mittel zur Sicherstellung der konkreten Qualitätsanforderungen an Software.



- Eine Gruppe von Qualitätsprüfern prüft teilweise oder ganz das Softwaresystem und die dazugehörige Dokumentation.
- Code, Entwurf, Spezifikationen, Testentwürfe, Standards, etc. können alle überprüft werden.
- Die Software oder Dokumente sollten als freigegeben markiert werden, damit das Fortfahren zum nächsten Schritt durch das Management abgesegnet werden kann.
- Es gibt drei verschiedene Prüfungen mit verschiedenen Zielen:
  - Inspektion zur Mängelbeseitigung (Produkt)
  - Prüfung der Fortschrittsbewertung (Produkt / Abläufe)
  - Prüfung der Qualität (Produkt / Standards)



# Qualitätsprüfung: Analytische Maßnahmen

- Prüfung der Produkte
  - Statische Prüfung
    - ?
  - Dynamische Prüfung
    - ?
- Prüfung der Prozesse
  - Audits
  - Prozessbeurteilung

- Prüfung der Produkte
  - Statische Prüfung
    - Review
    - Statische Analyse
    - Formale Programmverifikation
    - Model Checking
  - Dynamische Prüfung
    - Test
    - Simulation
    - Prototypen
- Prüfung der Prozesse
  - Audits
  - Prozessbeurteilung

## Qualitätsverbesserung (quality improvement):

Teil des Qualitätsmanagements, der auf die Erhöhung der Fähigkeit zur Erfüllung von Qualitätsanforderungen gerichtet ist. [ISO 9000:2000]

- Behebung der bei der Produktprüfung gefundenen Qualitätsmängel
  - Notwendig zur Erreichung von Produktqualität.
  - Häufig jedoch nur Symptombekämpfung.
- Modifikationen im Entwicklungsprozess und im Qualitätsmanagementsystem aufgrund von:
  - Auswertung von Fehlerursachen
  - Resultaten von Audits
  - Messungen

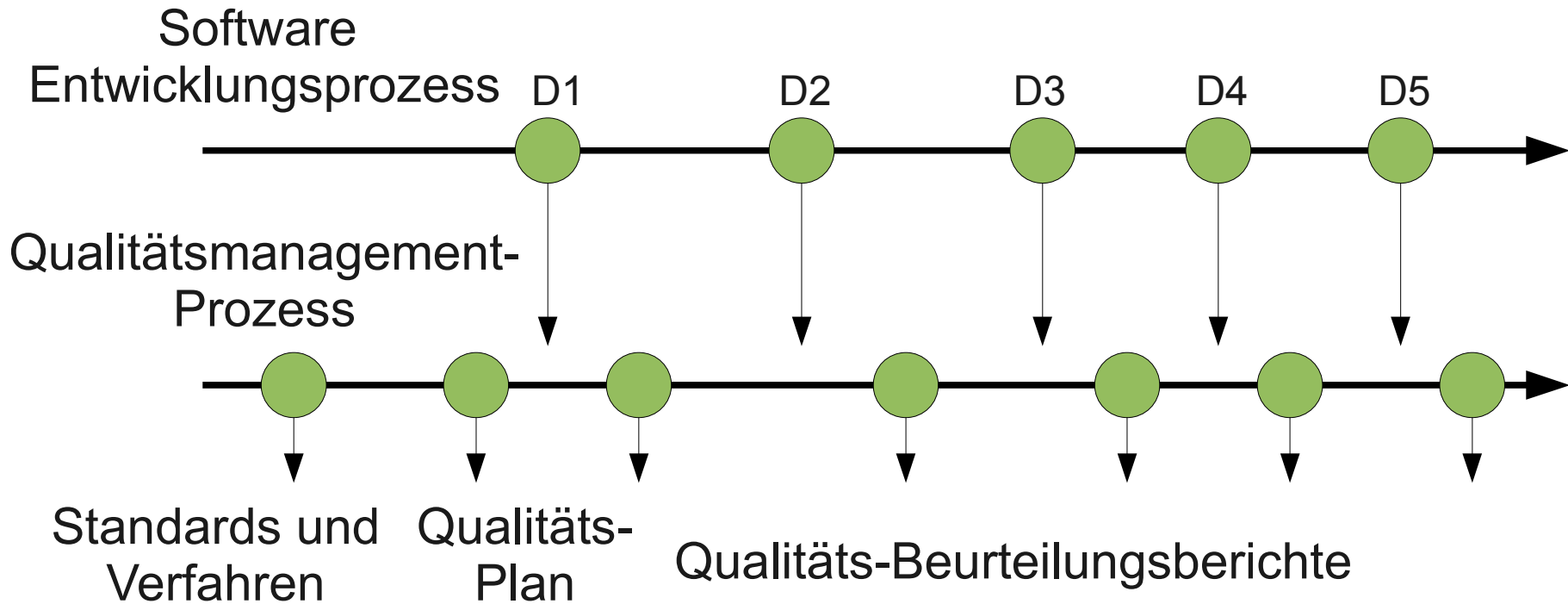
=> Prozessverbesserung



**Qualitätssicherung (quality assurance):** Teil des Qualitätsmanagements, der auf das Erzeugen von Vertrauen darauf gerichtet ist, dass Qualitätsanforderungen erfüllt werden. [ISO 9000:2000]

- Regelmäßige Überprüfung der Wirksamkeit des Qualitätsmanagementsystems durch Experten in Audits
- Publikation von qualitätsrelevanten Messgrößen
- Dokumentation (und teilweise Offenlegung) der Prozesse und Qualitätsverfahren
- Zertifizierung des Qualitätsmanagementsystems
- Aktionsprogramme zur Verbesserung der Prozesse für die Entwicklung, Pflege und Verwaltung von Software





- Software-Qualität
  - Was ist Qualität?
  - Qualitätsmerkmale
  - Qualitätsmanagement
- Software-Standards



- Standards definieren, welche Eigenschaften eines Produktes oder Prozesses benötigt werden. Sie spielen eine wichtige Rolle im Qualitätsmanagement.
- Standards können internationale oder nationale, Organisations- oder Projektstandards sein.
  - Oben bereits erwähnte internationale Standards: ISO 9000 (s.a. Abschnitt 3.1), IEEE 610, ISO/IEC 9126)
- Produktstandards definieren Charakteristiken, die alle Softwarekomponenten aufweisen müssen, z.B. ein gemeinsamer Programmierstil.
- Prozessstandards definieren, wie der Softwareprozess durchgeführt werden sollte.



Was könnten mögliche Vorteile und Ziele von Standards sein ?

Was könnten mögliche Vorteile und Ziele von Standards sein ?

- Erschließung der besten Verfahren – Vermeidung der Wiederholung vergangener Fehler.
- Standards sind ein Rahmen, um zu definieren, was Qualität in einem bestimmten Umfeld ausmacht, z.B. die Meinung eines bestimmten Unternehmens bzgl. Qualität.
- Sie bieten Kontinuität – neue Mitarbeiter verstehen das Unternehmen, indem sie die genutzten Standards verstehen.



Was könnten mögliche Probleme mit Standards sein ?

Was könnten mögliche Probleme mit Standards sein ?

- Durch die schnelle Entwicklung der IT können sie relativ schnell aufhören, relevant oder up-to-date zu sein.
- Wenn sie nicht von den Softwaretools unterstützt werden, ist das Aufrechterhalten der Dokumentation (bzgl. der Standards) oft mit langwierigem Formular-Ausfüllen verbunden.

- Involviert Fachleute in die Entwicklung. Die Entwickler sollten das Grundprinzip hinter dem Standard verstehen.
- Regelmäßige Überprüfung der Standards und deren Nutzen. Sie können schnell veralten, was ihrem Ansehen unter den Entwicklern schädigt.
- Detaillierte Standards sollten spezialisierte Programmunterstützung erhalten. Übermäßige Dokumentationsarbeit ist das größte Argument gegen Standards.

# Mögliche Inhalte für Produkt- und Prozessstandards

Was könnten mögliche Inhalte für Produkt- und Prozessstandards sein ?

# Mögliche Inhalte für Produkt- und Prozessstandards

| Produktstandards                   | Prozessstandards              |
|------------------------------------|-------------------------------|
| Form der Entwurfsprüfung           | Leitung der Designprüfung     |
| Struktur der Anforderungsdokumente | Freigabe für neuen Systemcode |
| Formatierung der Methodenköpfe     | Versionsrelease-Prozess       |
| Java Programmierstil               | Genehmigung des Projektplans  |
| Formatierung des Projektplans      | Kontrolle der Änderungen      |
| Form einer Änderungsanfrage        | Prozess der Testdurchläufe    |





- Software-Qualitätsmanagement soll sicherstellen, dass das Produkt wenige Fehler hat und die Standards bei Wartbarkeit, Zuverlässigkeit, Portabilität und weiteren Faktoren einhält.
- Software-Qualitätsmanagement beinhaltet, dass für Abläufe und Produkte Standards erstellt werden, sowie dass bei diesen Abläufen auch die Standards eingehalten werden.
- Software-Standards sind wichtig für die Qualitätssicherung, weil sie „den besten Weg“ anzeigen.

In den folgenden beiden Abschnitten geht es genauer um Prozessqualität sowie Softwarequalität (insbesondere Metriken).