

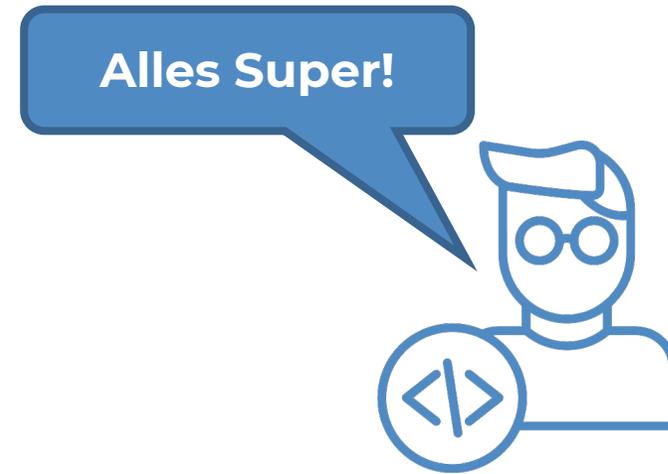
SIND SOFTWARE-METRIKEN VERMESSEN? - VOM SINNVOLLEN EINSATZ VON METRIKEN UND DEREN EFFIZIENTER ERHEBUNG

Stephan Pirnbaum

15.03.2023

- ▲ Dresdner IT-Consulting-Unternehmen
 - ▲ Zu Hause in der Java-Welt verbunden mit neuesten Technologien und Altbewährtem (z. B. React, Angular, Python, Neo4j, etc.)
 - ▲ Langjährige Forschungsk Kooperationen mit Hochschulen und Engagement in Communities (JUG Saxony)
- ▲ Unsere Schwerpunkte
 - ▲ Strategische, zielgerichtete und nachhaltige Entwicklung von Geschäftsanwendungen
 - ▲ Software-Qualitätsanalysen und -Sicherung
 - ▲ Köpfe hinter jQAssistant: von Entwicklung über Workshops bis hin zu Consulting
- ▲ Unsere Kunden
 - ▲ Kleine Auswahl: ITZBund, Sächsische Aufbaubank-Förderbank, ASML, GlobalFoundries, Thyssenkrupp Steel, Telekom, COOP Schweiz



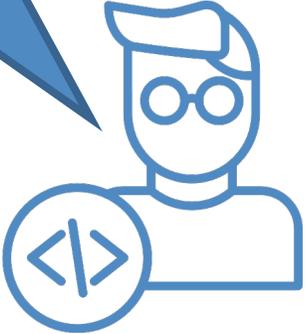




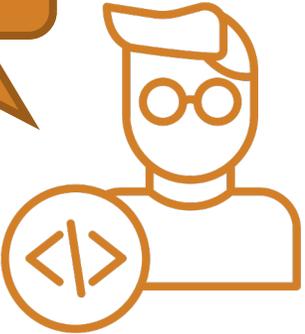


Wie gut ist eigentlich unsere Software?

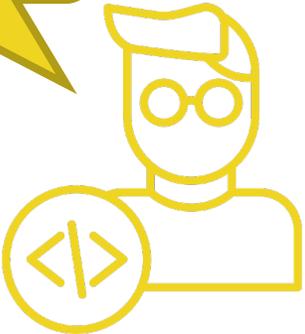
Alles Super!



Naja, könnte besser sein...



Tool XY sagt 4,5/5

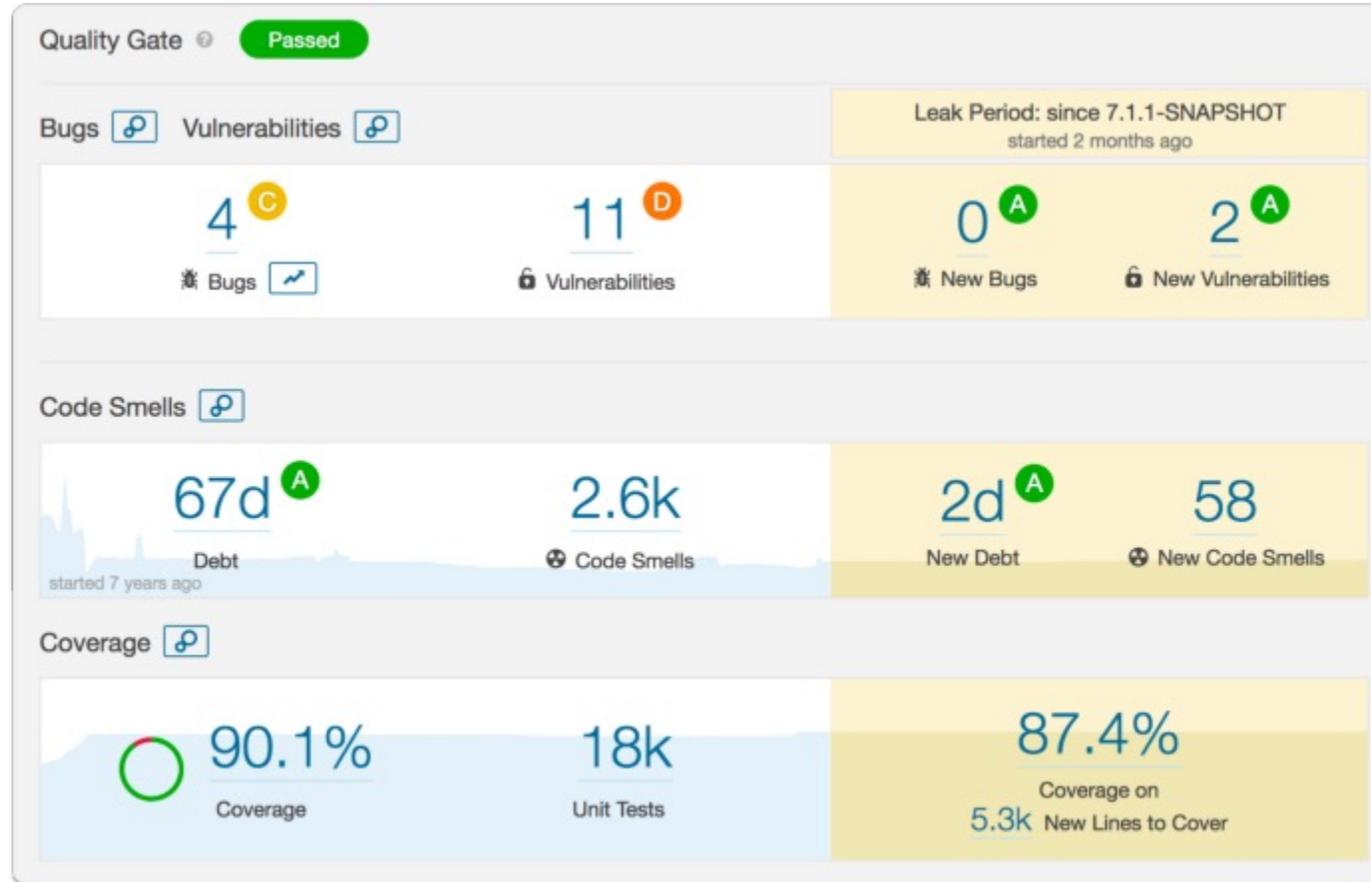




Ist das hilfreich für euch?



Warum genau diese Zahlen?



<https://medium.com/red6-es/go-for-sonarqube-ffff5b74f33a>

Was macht ihr mit den Zahlen?



Was wenn Grenzwerte verletzt werden?



- ▲ Metriken durch Tools häufig „da“
 - ▲ Kein Problem, solange nicht beliebige Quality Gates etabliert werden
 - ▲ Kein Nutzen, da keine Projekt-intrinsische Motivation vorhanden

- Auf was soll optimiert werden? Und warum?

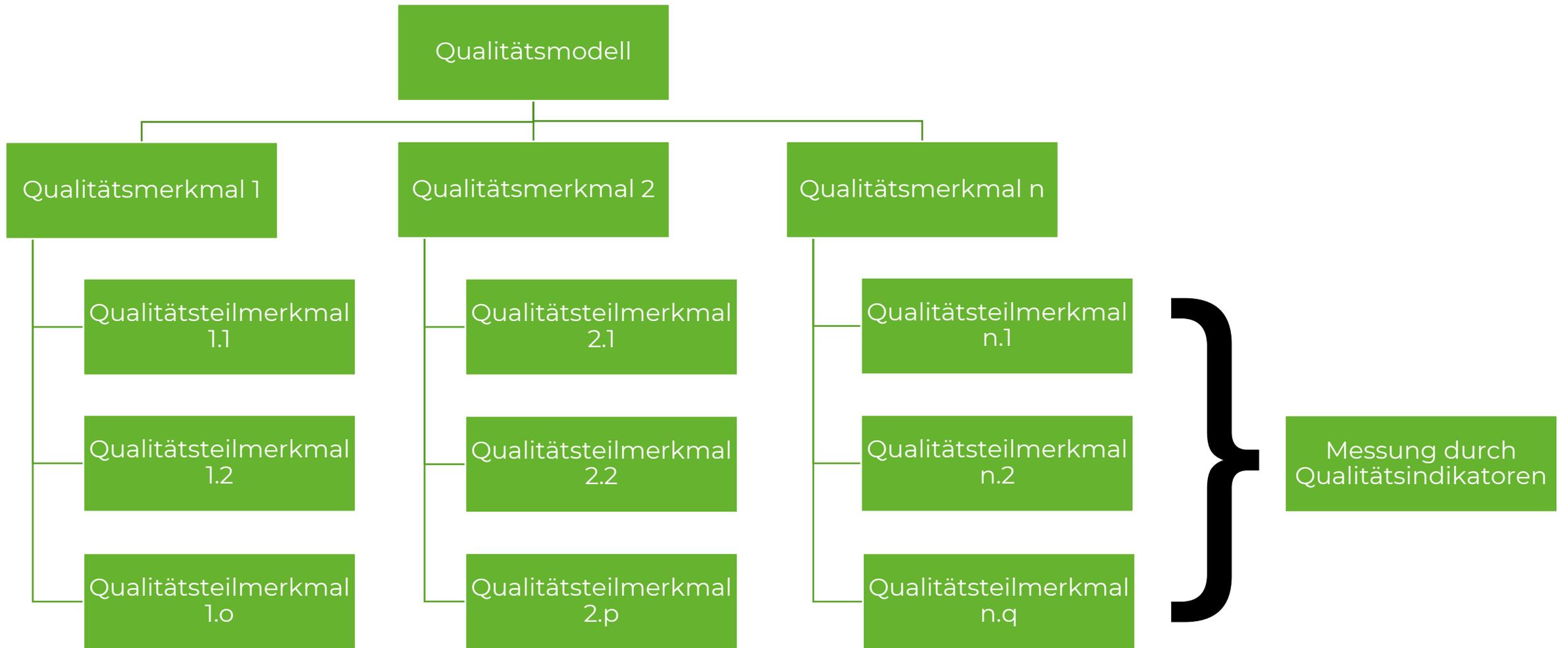
EIN KURZER AUSFLUG IN DIE WELT DER SOFTWAREQUALITÄT

... und warum diese wichtig ist

Unter **Softwarequalität** versteht man die Gesamtheit der **Merkmale** und **Merkmalswerte** eines Softwareprodukts, die sich auf dessen **Eignung** beziehen, festgelegte oder vorausgesetzte **Erfordernisse** zu erfüllen.

- Helmut Balzert -

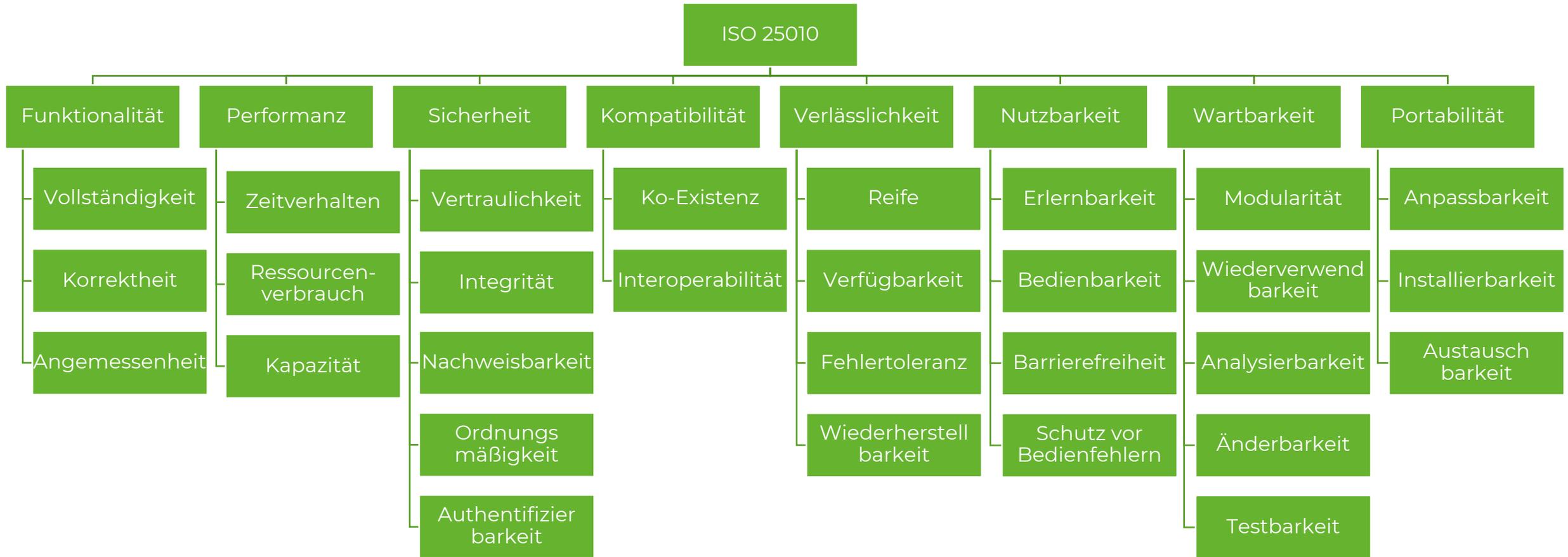
- ▲ Softwarequalität ist ein abstrakter, nicht direkt anwendbarer Begriff
- ▲ Qualitätsmodelle konkretisieren den Begriff „Softwarequalität“ mit
 - ▲ Qualitätsmerkmalen (*factor*)
 - ▲ Qualitätsteilmerkmalen (*criterion*)
 - ▲ Qualitätsindikatoren (*metric*)
- ▲ Qualitätsmodelle machen Softwarequalität mess- oder beobachtbar

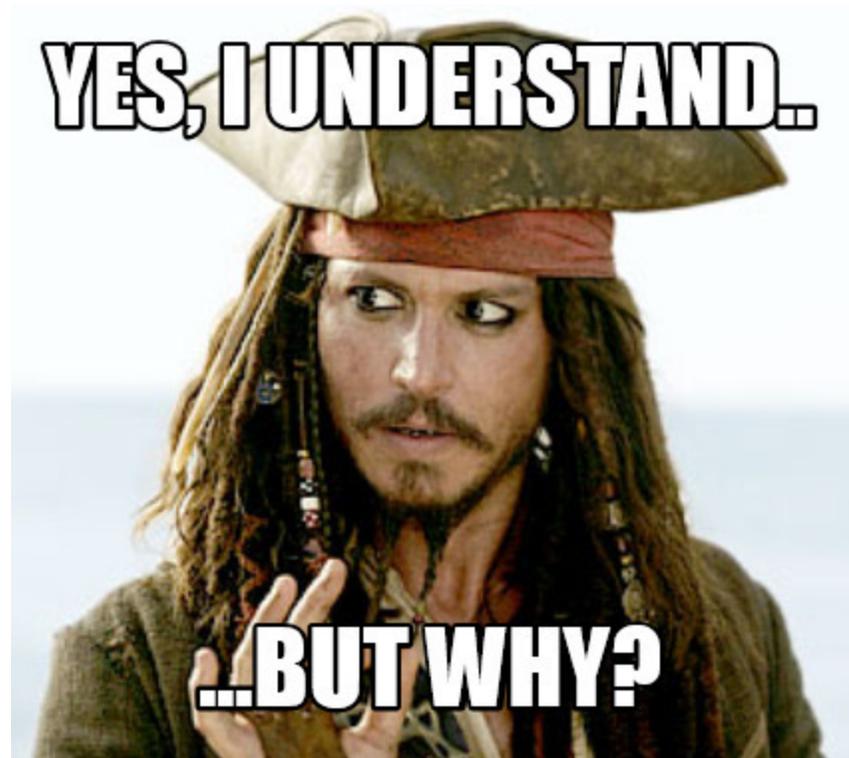


ISO 25010

–

„Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models“





<https://www.mememaker.net/meme/yes-i-understand.-...but-why/>

Software wird entwickelt, um
Geschäftsmodelle effizient zu
ermöglichen und Prozesse zu
optimieren

Im Vordergrund der Entwicklung
steht also die fachliche Domäne.
Diese sowie geschäftliche
Vorgaben bestimmen, welche
Qualitätsmerkmale relevant sind

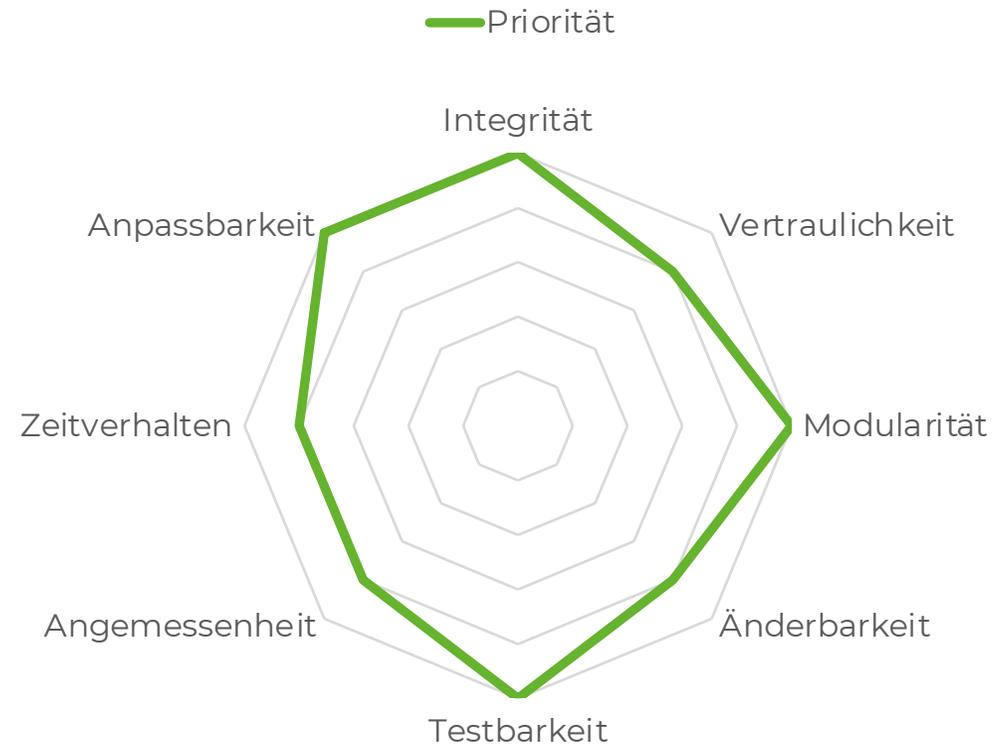
Sind die zu erreichenden **Qualitätsziele** unbekannt oder die Architektur und Umsetzung der Software nicht an diesen orientiert, hat dies weitreichende Folgen auf z.B.

- ▲ Entwicklungskosten und -dauer
- ▲ Abschätzbarkeit und Verlässlichkeit der Planung
- ▲ Laufzeiteigenschaften wie Response Time oder Ressourcenverbrauch
- ▲ Nutzer-/Kundenzufriedenheit
- ▲ Sicherung des Geschäftsmodells

- ▲ Jede Software ist individuell, daher muss auch das Qualitätsmodell individuell erstellt werden
 - ▲ Welche Vorgaben und (Geschäfts-)Ziele definiert das Unternehmen?
 - ▲ Was ist der Kontext und die Vision für das Softwareprojekt?
 - ▲ Welche Ansprüche haben die Nutzer bzw. Kunden?
 - ▲ Welche technischen Rahmenbedingungen existieren?
 - ▲ Was sind die Bedürfnisse der Softwareentwickler, QA/QS-Abteilung, etc.?

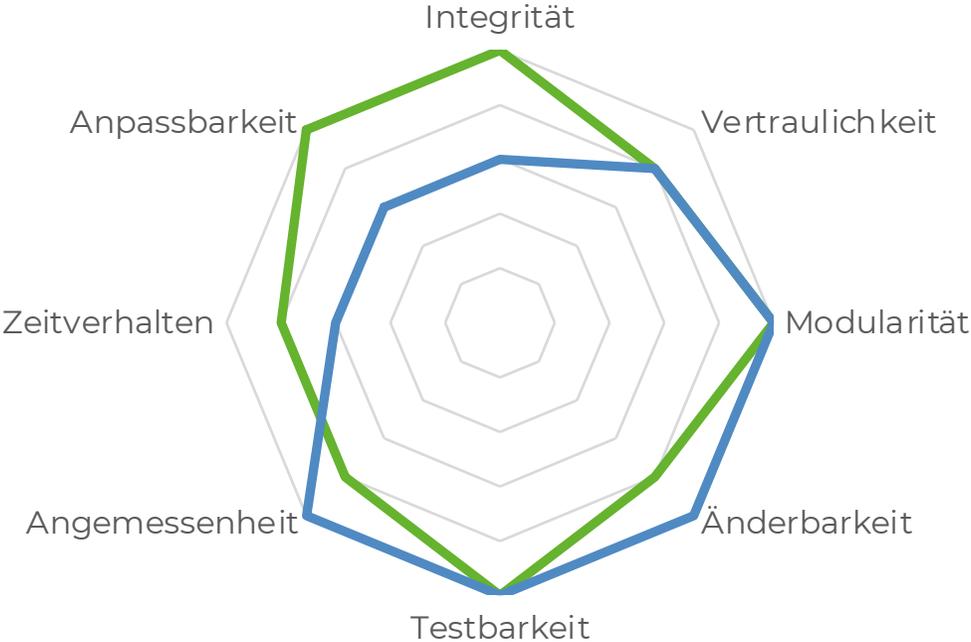
- Resultierendes Qualitätsmodell ist zumeist ein Trade-Off

Soll-Einschätzung



Soll- und Risiko-Einschätzung

— Priorität — Risiko



Du kennst jetzt die zu optimierenden Qualitätsmerkmale?

Woher weißt du, ob du die gesetzten Ziele erreichst?

Softwarequalität muss messbar werden!

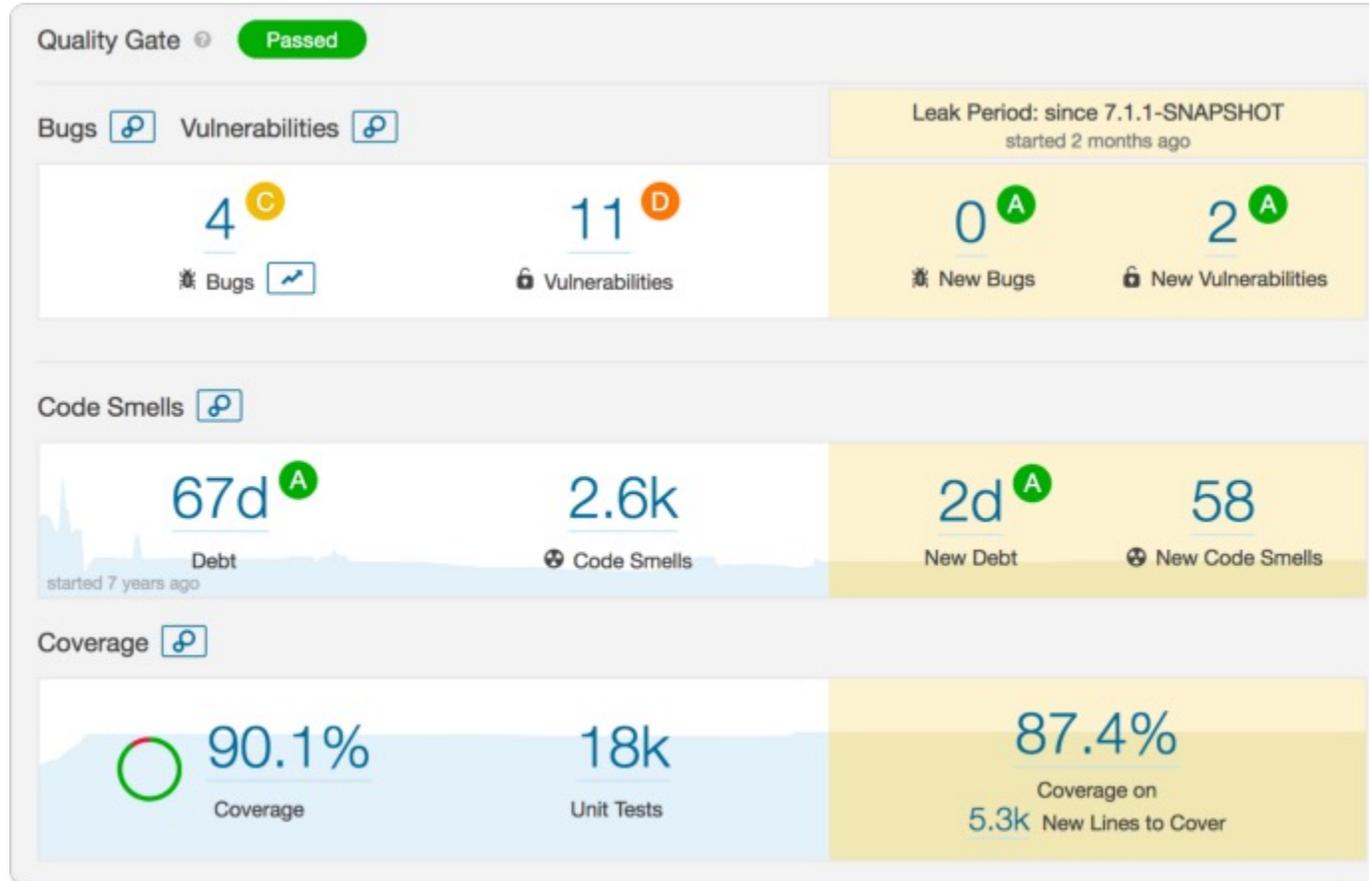
WAS SIND METRIKEN?

...und welche brauchen wir?

Eine Softwarequalitätsmetrik ist eine Funktion, die eine Software-Einheit in einen **Zahlenwert** abbildet, welcher als **Erfüllungsgrad** einer **Qualitätseigenschaft** der Software-Einheit interpretierbar ist.

- IEEE Standard 1061, 1998 -

- ▲ Metriken machen Qualitätsmerkmale messbar, z.B.
 - ▲ Zeitverhalten (Performanz) -> Response Time messen
 - ▲ Modularität (Wartbarkeit) -> Kohäsion und Kopplung messen
- ▲ ABER:
 - ▲ Eine Metrik bzw. deren Messung ist die Abbildung des Systems auf eine Quantität
 - ▲ Die Quantität stellt definitionsbedingt immer nur eine Annäherung einer Qualität dar
 - ▲ Jede Qualität ist in zahlreichen Quantitäten ausdrückbar
 - Selektion der Metriken ist maßgeblich für die Qualität der Bewertung



<https://medium.com/red6-es/go-for-sonarqube-ffff5b74f33a>

^ Das Problem:

- ^ Verwendung sehr umfangreicher Standardregelsätze
 - ^ Abschreckende Menge an „Technical Debt“ (hier: 67 Tage)
 - ^ Irrelevante, niemals behobene Findings
 - Schlechter Eindruck für Stakeholder außerhalb der Entwicklung
 - Abstumpfen der Entwickler gegenüber Findings

Das waren: Vermessene Software-Metriken

Aber wie geht es richtig?

- ▲ Metriken sind hilfreich, wenn bekannt ist
 - ▲ warum diese erhoben werden
 - ▲ für wen die Metriken erhoben werden
 - ▲ welche Auswirkungen diese in der Praxis haben
 - ▲ Stichwort: Leading vs. Trailing Metrics!
 - ▲ wie sich diese berechnen
 - ▲ welche Werte erreicht werden müssen
 - ▲ wie die Werte verbessert werden können

- Wir lernen: Auch eine Metrik wie die Lines of Code kann hilfreich sein

- ▲ Metriken sollten zusätzlich selber auf ihre Qualität geprüft werden (nach Helmut Balzert)
 - ▲ Objektivität – der Messende hat keinen subjektive Einfluss auf die Messung
 - ▲ Zuverlässigkeit – die Messung ist mit gleichem Ergebnis wiederholbar
 - ▲ Validität – die Messung misst das Beabsichtigte
 - ▲ Normierung – die Ergebnisse lassen sich auf einer Skala anordnen
 - ▲ Vergleichbarkeit – die Ergebnisse lassen sich vergleichen
 - ▲ Ökonomie – die Messung lässt sich mit geringen Kosten umsetzen
 - ▲ Nützlichkeit – die Metrik hat eine praktische Bedeutung

▲ Vorgehen zur Selektion von Metriken

1. Qualitätsmodell mit Qualitäts(teil)merkmalen festlegen
 2. Priorisierung und Risikobewertung der Merkmale
 3. Identifikation möglicher Metriken (je Merkmal)
 4. Evaluierung der Metriken nach Qualitätskriterien für Metriken
 5. Erprobung der Metriken und Identifikation sinnvoller Grenzwerte
 6. Festlegung auf initialen Satz an Metriken
- Kontinuierliche Evaluierung, Monitoring, Anpassung

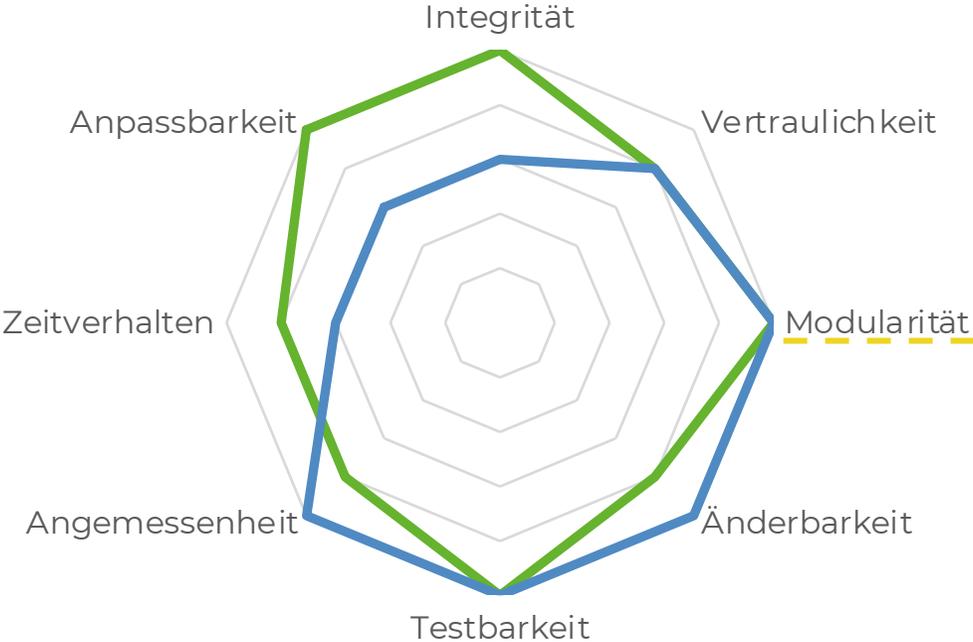
- ▲ Qualitätsmerkmale können mit vielen Metriken angenähert werden
- ▲ Annäherung ist in Ordnung
 - ▲ solange Einschränkungen bekannt sind
 - ▲ solange keine falschen Schlüsse daraus gezogen werden
- ▲ Am Besten
 - ▲ Mehrere Metriken verwenden und diese in der Gesamtheit betrachten
 - ▲ Keine „Harten Grenzwerte“ sondern Staffelung

QUALITÄTBEWERTUNG VON SOFTWARESYSTEMEN

...in der Praxis

Soll- und Risiko-Einschätzung

— Priorität — Risiko



▲ Beispiel „Modularität“ (Wartbarkeit)

- ▲ **Priorität: 5** - Das zu entwickelnde System besteht aus mehreren fachlichen Bausteinen, die unabhängig voneinander mit Anforderungen versorgt werden und deren Größe die Entwicklung durch unterschiedliche Teams erfordert.
- ▲ **Risiko: 5** - Bei Nichterrechung können neue Anforderungen nicht mehr zeitnah umgesetzt werden, zu ändernder Code ist nicht in Modulgrenzen identifizierbar und bei der Umsetzungen mehrere Anforderungen kann es zu Konflikten am gemeinsam geänderten Code kommen.

▲ Beispiel „Modularität“ (Wartbarkeit)

▲ Metrik 1: Anzahl der Module

- ▲ Anzahl der Top-Level-Maven-Module im Projekt

▲ Metrik 2: Kopplungsgrad zwischen Modulen

- ▲ Anzahl der Abhängigkeiten zwischen Modulen

▲ Metrik 3: Gleichmäßigkeit der Module

- ▲ Größenverteilung über die Module

➤ Was ist Modularität?

^ Beispiel „Modularität“ (Wartbarkeit)

^ Metrik 1: Anzahl der Module

- ^ Objektivität ✓
- ^ Zuverlässigkeit ✓
- ^ Validität ✗
- ^ Normierung ✓
- ^ Vergleichbarkeit ✓
- ^ Ökonomie ✓
- ^ Nützlichkeit ✗

^ Metrik 2: Kopplungsgrad zwischen Modulen

- ^ Objektivität (✓)
- ^ Zuverlässigkeit ✓
- ^ Validität ✓
- ^ Normierung ✓
- ^ Vergleichbarkeit ✓
- ^ Ökonomie (✓)
- ^ Nützlichkeit ✓

^ Metrik 3: Gleichmäßigkeit der Module

- ^ Objektivität (✓)
- ^ Zuverlässigkeit ✓
- ^ Validität (✓)
- ^ Normierung (✓)
- ^ Vergleichbarkeit (✓)
- ^ Ökonomie ✓
- ^ Nützlichkeit (✓)

▲ Beispiel „Modularität“ (Wartbarkeit)

- ▲ Erfassung der Metriken muss automatisiert laufen, jedoch transparent
- ▲ Problem: Zur Berechnung der Metrik wird Wissen über die Architektur benötigt
 - Was sind Module?
 - Wie manifestieren sich Abhängigkeiten zwischen Modulen?
- Je nach Metrik werden verschiedene Datenquellen benötigt, z.B.
 - ▲ Source Code
 - ▲ Git-Historie
 - ▲ Laufzeitinformationen
 - ▲ Architekturdokumentation
 - ▲ ...

^ Beispiel „Modularität“ (Wartbarkeit)



DEMO

Hilfe: Es werden schon zahlreiche Metriken erhoben 😞

- ▲ Qualitätsziele definieren (wenn noch nicht vorhanden)
- ▲ Übersicht über genutzte Metriken aufbereiten
- ▲ Zuordnung der Metriken zu den Qualitätszielen
 - ▲ Mehrfachzuordnung möglich!
 - Wenn keine Zuordnung: Einzelfallbetrachtung
- ▲ Bewertung der Metriken nach Qualitätskriterien
- ▲ Festlegen (und dokumentieren ;)) genutzter Metriken
 - ▲ Einschließlich was, warum, wie, für wen, ...
 - ▲ **Commitment aller relevanter Stakeholder!**

VIELEN DANK

Noch Fragen?

www.buschmais.de

BUSCHMAIS

KONTAKT

Stephan Pirnbaum

stephan.pirnbaum@buschmais.com

+49 351 320923-22

Social Media



@spirnbaum



stephan-pirnbaum

BUSCHMAIS GbR

Leipziger Straße 93

01127 Dresden

Tel. +49 351 3209230

info@buschmais.com

www.buschmais.de