

Es gibt keine nicht funktionalen Tests -
Es gibt nur nicht funktionale Anforderungen!

Stephan Jacobs, FH Aachen

Was verstehen andere unter nicht funktionalen Tests?

- *Nichtfunktionale Tests decken unter anderem Fehlverhalten bezüglich Performanz, allgemeiner Benutzbarkeit und Sicherheit auf.*

Quelle: Enzyklopädie der Wirtschaftsinformatik

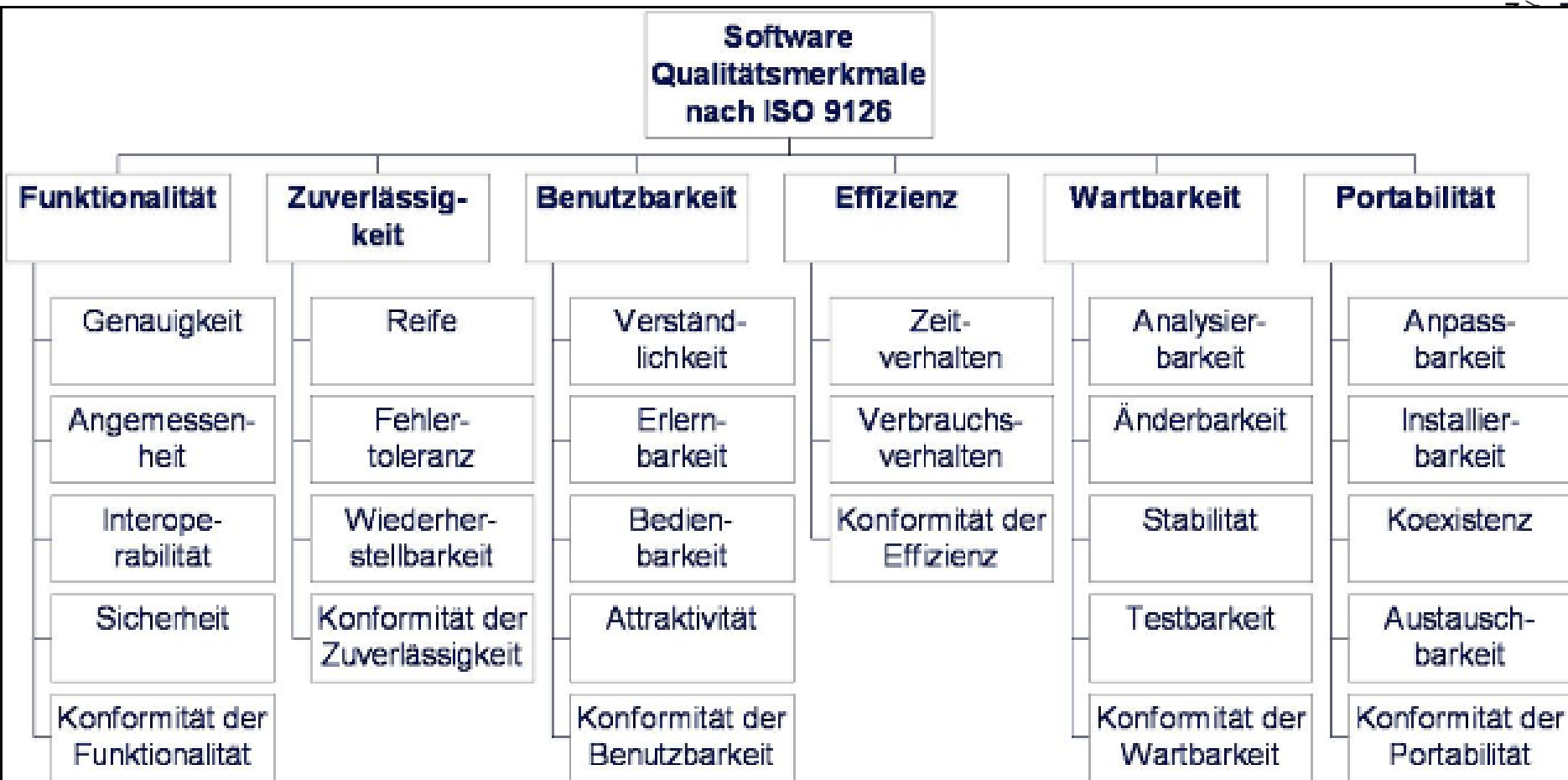
- *Der nicht funktionale, technische Test ist eine wichtige Ergänzung des funktionalen Tests. Spezielle Testarten geben hier Aufschluss über Sicherheit, Gebrauchstauglichkeit und Zuverlässigkeit des Systems und sollten daher nicht vernachlässigt werden.*

Quelle: pro-test.biz

Der nichtfunktionale Test testet nichtfunktionale Eigenschaften bzw. Anforderungen eines System.

Stimmt das?

Häufig wird auf die ISO 9126 (SW Qualität) Bezug genommen



Was sagt der ISTQB zum nicht funktionalen Test?

3 Testen im Softwarelebenszyklus

3.7 Grundsätzliche Testarten

3.7.2 Nicht funktionaler Test

- *Nicht funktionale Anforderungen beschreiben Attribute des funktionalen Verhaltens ...*
- *Ihre Umsetzung beeinflusst stark, wie zufrieden der Kunde ist ...*
- *Merkmale nach ISO 9126 sind:*
 - *Zuverlässigkeit*
 - *Benutzbarkeit*
 - *Effizienz*
 - *...*

Was sagt der ISTQB zum nichtfunktionalen Test? (ctd.)

3 Testen im Softwarelebenszyklus

3.7 Grundsätzliche Testarten

3.7.2 Nicht funktionaler Test

- *Folgende Systemeigenschaften sollten durch entsprechende Tests berücksichtigt werden:*
 - *Lasttest*
 - *Performanztest*
 - *...*
 - *Prüfung der Dokumentation*
 - *Prüfen auf Änderbarkeit/Wartbarkeit*
- *Problem: Lückenhafte und schwammige Formulierung*
- *Empfehlung: Vertreter des Systemtests, sollten beim Review der Anforderungsspezifikation teilnehmen, ...*

Quelle: Basiswissen Softwaretest

- Umgangssprachlich wird der Begriff "nicht funktionaler Test" verwendet, um den Test einer nicht funktionalen Anforderung zu beschreiben.
- Dadurch wird der Test selbst aber nicht "nicht funktional".
 - Im Gegenteil: Die meisten (alle?) "nicht funktionalen Tests" sind funktional. Allerdings liegt eine spezielle Testanordnung vor.
 - Systemumgebung (Kompatibilitätstest, Lasttest, ...)
 - Messungen (Benutzbarkeit, Performanz, ...)
 - Ein "benutzbarer Test" ist etwas anderes als ein Test der Benutzbarkeit

Es gibt keinen nicht funktionalen Test.

Es gibt nur Tests nicht funktionaler Anforderungen.

- Hilft diese Erkenntnis?
 - Nicht wirklich ...

- Wie testen Sie nicht funktionale Anforderungen? Gibt es dabei Probleme?
 - Sind die nicht funktionalen Anforderungen sauber (= testbar) spezifiziert?
 - Benutzerfreundlichkeit? Robustheit? Sicherheit?

- Nehmen Sie am Review von Anforderungsspezifikation teil?
 - Werden die nicht funktionalen Anforderungen dadurch präziser?

- Wer ist für die Präzisierung der (nicht funktionalen) Anforderungen zuständig?
 - Produktmanager?
 - Marketing
 - Kunde?
 - Entwickler / Tester?
- Sind die Verantwortlichen in der Lage, die Anforderungen präzise zu formulieren?
 - Haben die Verantwortlichen die notwendige Kompetenz?
- Was sind überhaupt die notwendigen Kompetenzen? Wie geht das, nicht funktionale Anforderungen sauber (testbar) zu spezifizieren?

- Ende der 90'er Jahre ... eine Abteilung in der SW-Entwicklung bei Ericsson stellt fest: Hauptursache für viele Probleme in der Entwicklung sind unklare Anforderungen
 - Warum sind Projekte verspätet?
 - Wieso werden so viele Fehler gefunden?

- Analyse mit Hilfe von
 - Seven Management Tools → Hier Root Cause Analysis
 - Analyse von Fehlern
 - Analyse von Verspätungen und Change Requests
 - Analyse von Dokumenten, speziell der Anforderungsspezifikation

Fallstudie: Neue Methode zur Spezifikation von Anforderungen

- Strategie:
 - Massiver Ansatz notwendig
 - Anforderungen nicht nur Aufgabe von Produktmanagement. Daher nehmen neben PM auch Entwickler, Tester, ... an einer Schulung teil
- Ein "teurer Experte" wird verpflichtet: Tom Gilb
 - Sein Ansatz "Planguage" wird vereinfacht und angepasst.
- Während eines einwöchigen Trainings ruhen alle Projekte

Fallstudie:

Welche Probleme gab es?

- Alter Ansatz ist gekennzeichnet durch
 - Anforderungen sind lange, monolithische Texte
 - Ein Teil der Anforderungen sind tatsächlich Lösungen
 - Nicht funktionale (oder Qualitäts-) Anforderungen sind unpräzise
 - Kosten (Geld, Zeit, Ressourcen) werden nicht spezifiziert
 - Fehlende Quellenangabe
- Beobachtete Effekte des alten Ansatz
 - Anforderungsspezifikationen werden geschrieben aber nicht als Grundlage für die Entwicklung verwendet
 - Spezifikationen sind inkonsistent
 - Nur ein Teil der Spezifikation wird getestet
 - Reviews der Spezifikationen funktionieren nicht.

Fallstudie: Wie funktioniert der neue Ansatz?

- Viele Teile der Methode sind nicht wirklich innovativ, neu oder überraschend.
- Allerdings diszipliniert und motiviert das gemeinsame Erlernen und "Einschwören" auf eine neue Methode bzw. Strategie

Der neue Stil

- Klare Trennung von
 - Anforderungen
 - Funktionale
 - Nicht funktional, Quality-Requirement
 - Randbedingungen (Constraints)
 - Kosten (Geld, Zeit, Ressourcen)
 - Annahmen

Fallstudie: Nicht funktionale Anforderungen messbar machen

- Zur Spezifikation von nicht funktionalen Anforderungen werden diese messbar gemacht.
- Dazu werden einige Konzepte der Sprach Planguage verwendet

Quelle: Tom Gilb, Planguage

- ID: Eindeutiger Identifikator
 - Gist: Kurze Zusammenfassung
 - Scale: Welche Einheit wird zur Messung verwendet
 - Meter: Wie wird gemessen?
 - Past, Record: historische Vergleichswerte
 - Must, Plan, Wish: Zielwerte für das neue System
-
- Die Anforderungen werden in ein Template eingetragen

Beispiel: Nicht funktionale Anforderung

Easy to learn

ID:	NF-100
Gist:	E-Note has to be easy to learn
Scale:	Time which is required to understand the (basic) function "Enter new address"
Meter:	Time which a manager of our company (not familiar with the system) needs to enter his first address
Past:	-
Record:	30s, typical value for iPhone
Must:	< 1 minute, if not understood within a minute, manager will not accept the system ← Marketing
Wish:	25s, beat the iPhone
Plan:	35s ← Marketing

Fallstudie: Lessons Learned

- Verbessertes Verständnis von Anforderungen
 - Unklare Anforderungen sind seltener
 - Anforderungsspezifikationen präziser
- Änderung im Verhalten
 - Auch die Qualität anderer Dokumente wird besser
 - Präzision bekommt einen Eigenwert
- Der "Gilb-Style" wird auch außerhalb von Anforderungsspezifikationen benutzt
 - Ziel eines Workshops
 - Anforderungen an Service-Funktionen innerhalb des Unternehmens
- Entwickler realisieren, dass ihre Position stärker ist, wenn sie präziser sind

Fallstudie: Lessons Learned (ctd.)

- Verbesserung von Reviews und Inspektionen
 - Es werden mehr Fehler gefunden
 - Reviews dienen der Präzisierung
- Kommunikationsprobleme mit Auftraggebern
 - Personen, die nicht an dem Training teilgenommen hatten, waren durch die neue Form der Anforderungen "irritiert".
 - Nicht alle sind in der Lage, präzise zu spezifizieren. Einige wollen/können nicht präzise spezifizieren.
 - Kein Problem unter Ingenieuren
 - Teilweise ein Problem mit Management und Marketing
- Fokus auf nicht funktionale Anforderungen
 - Die größte Qualitätsverbesserung vorher-nachher war im Bereich nicht funktionaler Anforderungen
 - Diese wurden vorher sehr unsauber formuliert
 - Hilflosigkeit: Keiner wusste, wie es geht

- Was hat das jetzt mit Testen nicht funktionaler Anforderungen zu tun?
 1. Auch im Testen ist Präzision notwendig.
 2. Auch im Testen bekommen wir unpräzisen Input. Beteiligung an Reviews hilft da oft nicht weiter.
 3. Auch im Test verbessern wir unsere Position, wenn WIR präzise sind oder Präzision einführen.
 4. Auch im Test haben wir Schnittstellen, die häufig nicht in der Lage oder nicht Willens sind, zu präzisieren.

Im Testen ist Präzision notwendig

- Gerade beim Testen nicht funktionaler Anforderungen ist eine sauber Spezifikation notwendig
 - Was soll getestet werden?
 - Was soll nicht getestet werden?
 - Wie sieht überhaupt die Anforderung aus?
- Präzision erfordert eine Sprache, mit der man einfach präzise kommunizieren kann.
 - Formale Sprachen sind zu schwierig (eignen sich nicht für nicht funktionale Anforderungen?)
 - Natürliche Sprache ist zu ungenau
 - Eine semi-formale Sprache wie Planguage kann hilfreich sein

Im Test bekommen wir unpräzisen Input

- Der Systemtest baut auf den gleichen Dokumenten auf, wie die Entwicklung
 - Anforderungsspezifikation
- Organisatorische Schnittstellen zum Systemtest sind u.a. Marketing, Produkt Management
 - Hier ist (s.o.) häufig nicht die notwendigen Kompetenzen bzw. der Wille vorhanden, präzise zu sein.

Präzise Spezifikationen verbessern die Position des Tests

- Sind die Spezifikationen eindeutig und präzise, dann ...
 - ... ist die Dokumentation von Testfällen einfacher
 - ... ist die Zuordnung von Testfällen zu Anforderungen einfacher
 - ... ist die Abschätzung von Risiken – wird ausreichend getestet – einfacher
 - ...



Ein paar Projekte später ...

- Wir sind für den Systemtest zuständig und haben Schwierigkeiten mit der Genauigkeit der Anforderungen
 - Einige Anforderungen sind ungenau
 - Das System soll möglichst robust sein ...
 - Andere Anforderungen sind genau aber nicht direkt testbar
 - Die Signallaufzeit soll maximal 0,3s betragen
Frage: In welchem Set-Up soll das gemessen werden? Mit Backbone? Ohne Backbone? Mit Hintergrundlast? Ohne? ...
 - Das System soll eine maximale "Unplanned Downtime" von 5 Minuten pro Jahr haben
Frage: Wie soll das innerhalb eines Systemtests von drei Monaten gemessen werden?

- Was tun?

Einführung einer Verification Requirements Specification

- Versuch 1: Wir bitten die Auftraggeber (Marketing, Produktmanagement, ...) um Präzisierung.
 - Ergebnis: Die anderen Rollen können sich häufig nicht in die Problematik des Tests hineindenken.
 - Teilweise fehlt die Fähigkeit zur Präzisierung
 - Wir kommen nicht weiter.
- Versuch 2: Wir präzisieren (high level), wie wir die Tests durchführen wollen.
 - Keine Testfallbeschreibung!
 - Präzisierung der Anforderungen hinsichtlich des Tests
 - Diese Präzisierung wird von den Auftraggebern bestätigt. Ggfs. werden Anforderungen angepasst.
 - M.E. kann dieser Schritt nur von Testern durchgeführt werden. Diese sind Kunden des eigenen Dokuments und schreiben dies zielgerichtet. Jemand ohne Test-Hintergrund, ist hierzu nicht in der Lage.

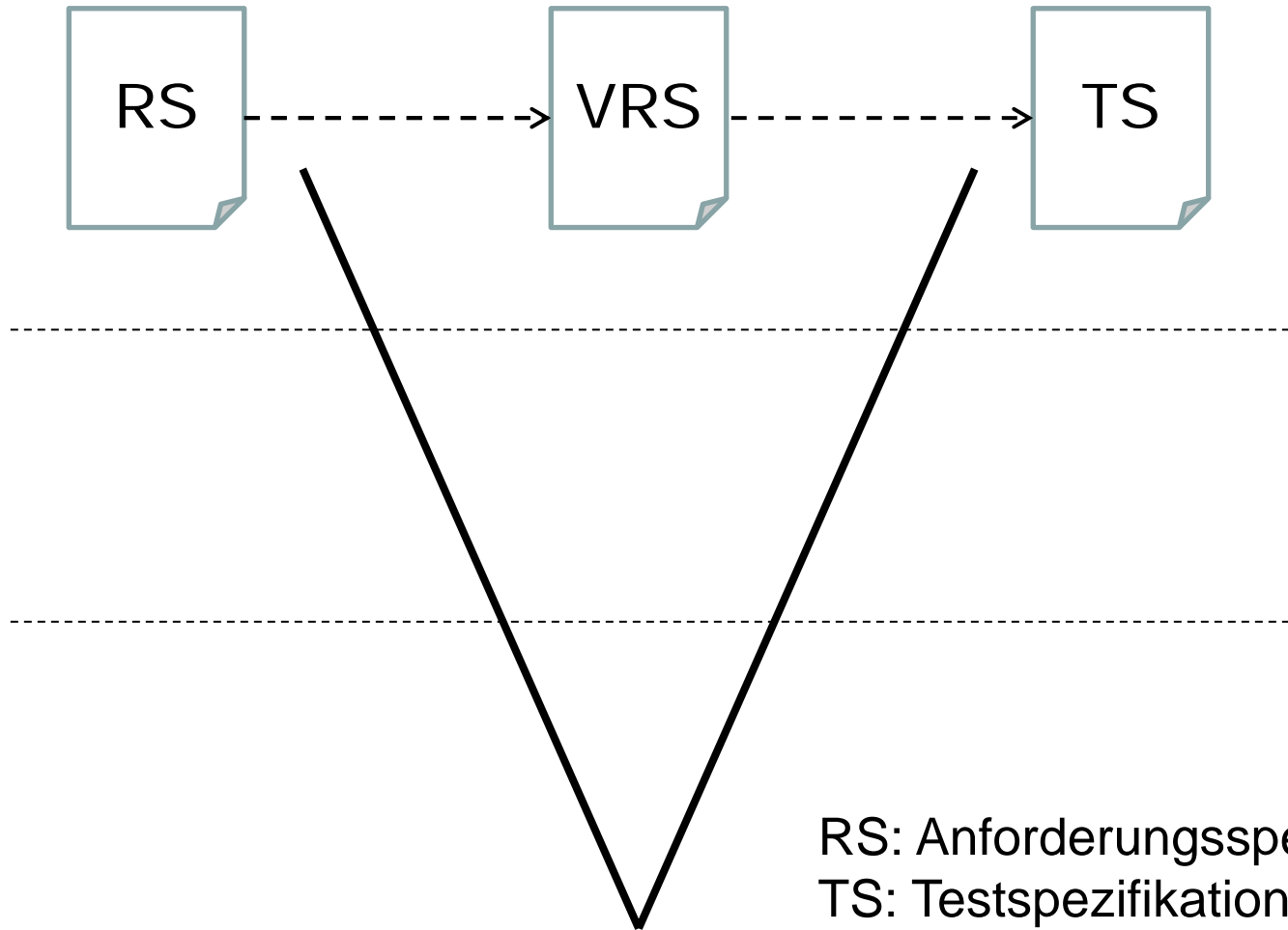
Beispielanforderungen in VRS

- Signallaufzeit maximal 0,3s
 - Testsystem: Split Architecture, ATM Transmission
 - Hintergrundlast: keine
 - Anzahl Messungen: 100
 - Erfolgreich wenn: 95% Messungen < 0,3s
 - Quelle: Anforderung Char_123

- Unplanned Downtime < 5 Minuten pro Jahr
 - Testsystem: Split Architecture, IP Transmission
 - Hintergrundlast: Lastprofil nach XY
 - Testdauer: 169h
 - Erfolgreich wenn:
 - Anforderung: Stab_345



Die VRS im V-Modell



RS: Anforderungsspezifikation
TS: Testspezifikation
VRS: Verification Req. Spec.

- Die VRS sind dann hilfreich, wenn der "Abstand" zwischen den Anforderungen und dem Test groß ist
- Beispiel:
 - Nicht funktionale Anforderungen
 - Kunden mit unklaren Vorstellungen
- In diesem Fall hilft die VRS, diesen Abstand zu überbrücken
- Wieso hilfreich:
 - VRS einfacher zu schreiben und zu verstehen als Testspezifikation
 - VRS kann (muss) von Testern geschrieben werden. Kein Warten auf andere.
 - Wie Anforderungen getestet werden, wird von Testern gesteuert
 - Präziserer Input für Testspezifikation

Thesen (zu diskutieren)

- Präzision ist beim Testen und bei der Spezifikation von Anforderungen notwendig
 - Und wenn Präzisierung nur zu Widerspruch führt.
- Nicht funktionale Anforderungen müssen präzise (= messbar?) spezifiziert werden.
- Präzisierung ist nicht allein eine Frage einer Methode sondern eine Frage des Verhaltens oder der Kultur.
- Die Präzisierung der Anforderungen ist nicht alleine Aufgabe von Produktmanagement (o.ä.). Insbesondere wenn PM nicht die notwendige Kompetenz hat – nicht zynisch gemeint (!) – ist es auch die Aufgabe der Tester, die Präzisierung von Anforderungen zu unterstützen.

Andreas Spillner, Tilo Linz: "Basiswissen Softwaretest", 5.Auflage, dpunkt.verlag, 2012

Guido Wirtz: "Nichtfunktionaler Test", Enzyklopädie der Wirtschaftsinformatik, online, 2012
<http://www.enzyklopaedie-der-wirtschaftsinformatik.de>

Stephan Jacobs, "Introducing measurable quality requirements: a case study", RE'99, pp. 172-179, 1999

Tom Gilb: "Planguage – A Handbook for Advanced practical Management", Manuscript, 1996