

**Welche Art von**

**Softwaretests**

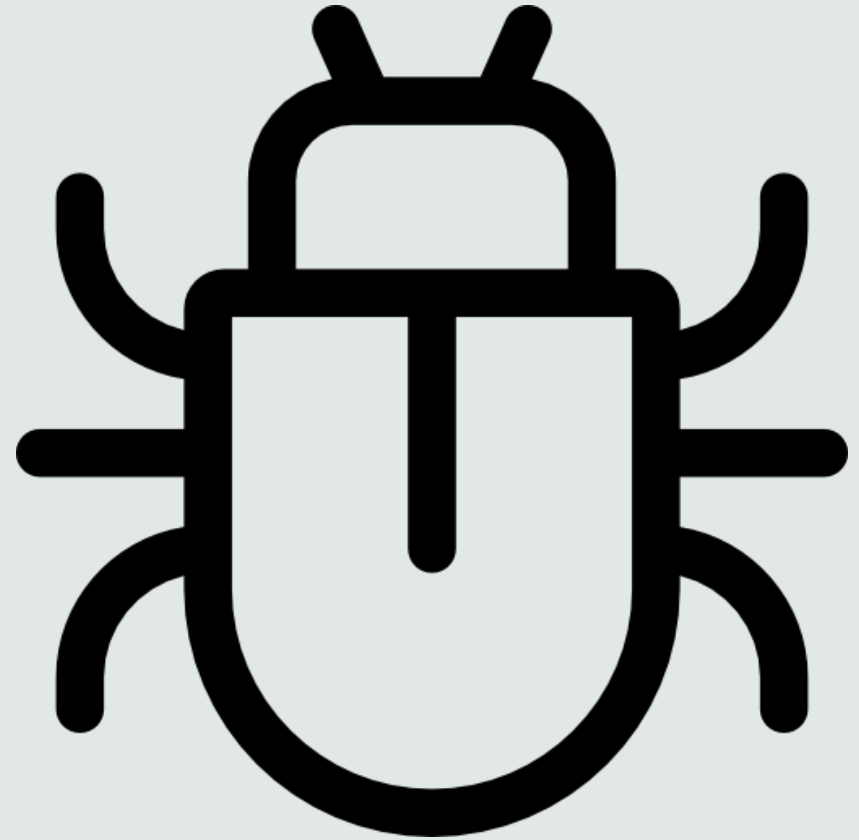
**brauche ich wirklich?**

# Warum testen wir?

**Bugs,**

**Fehler,**

**Defekte?**





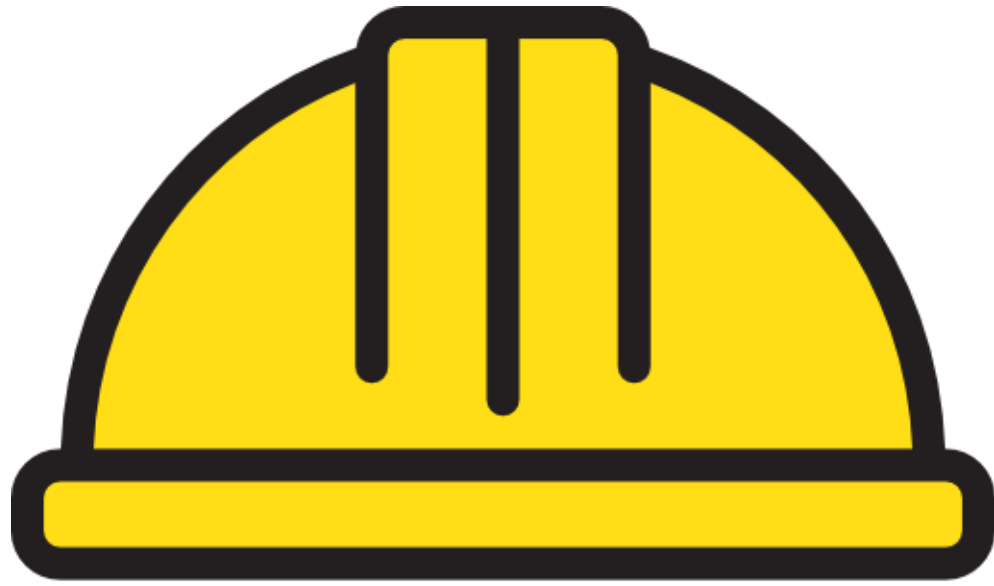
***Kunden-***

***Zufriedenheit?***



***Qualität?***





***Sicherheit?***



# Meistens...

**Meistens...**

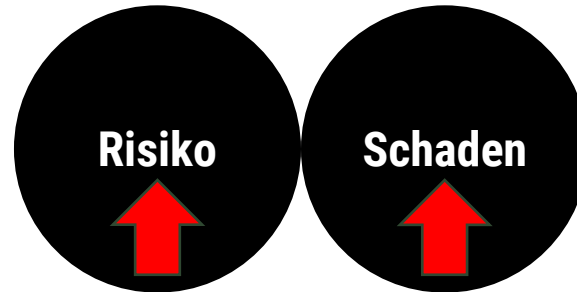
**... um Geld**

**zu sparen**

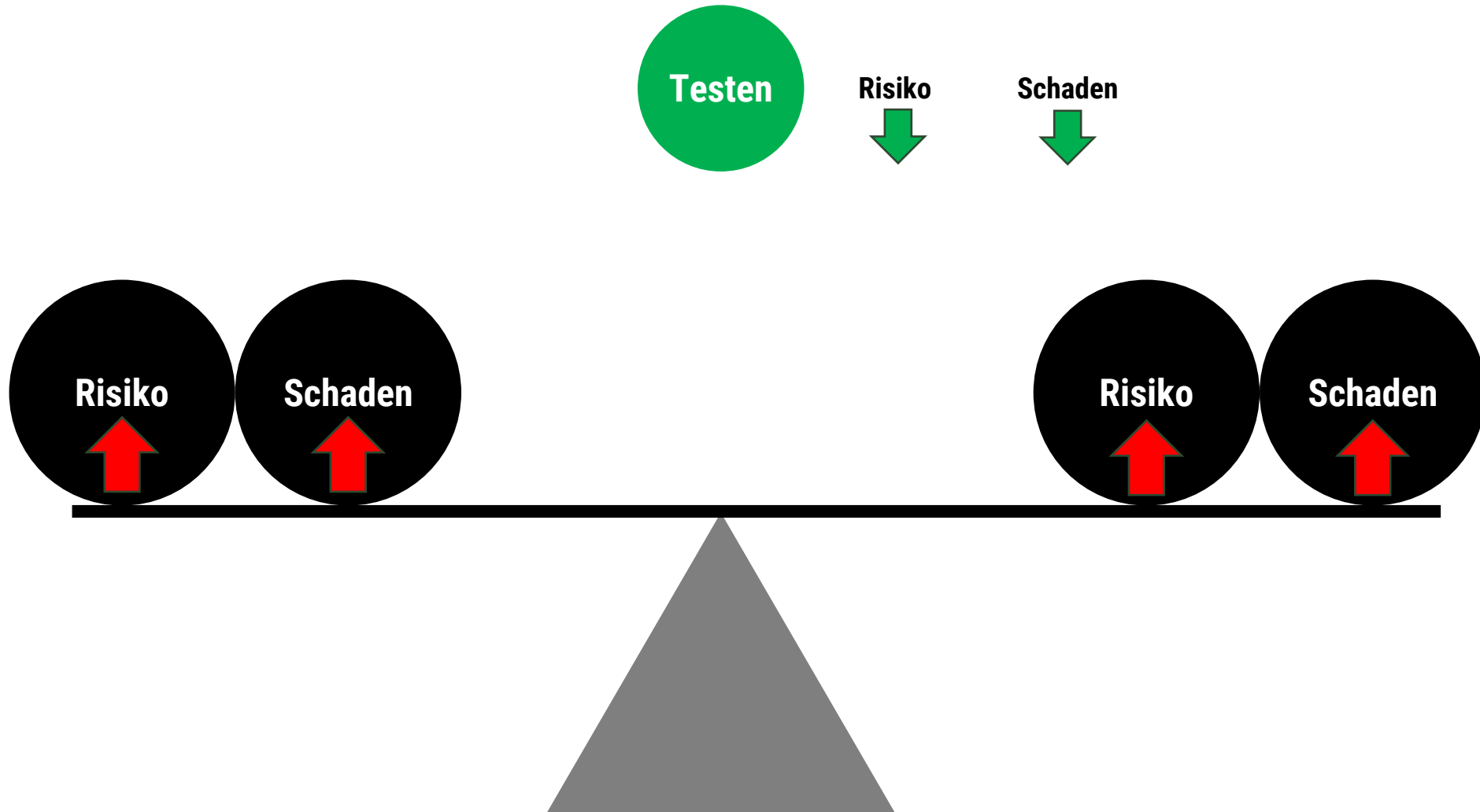


# Wie spart Testen Geld?

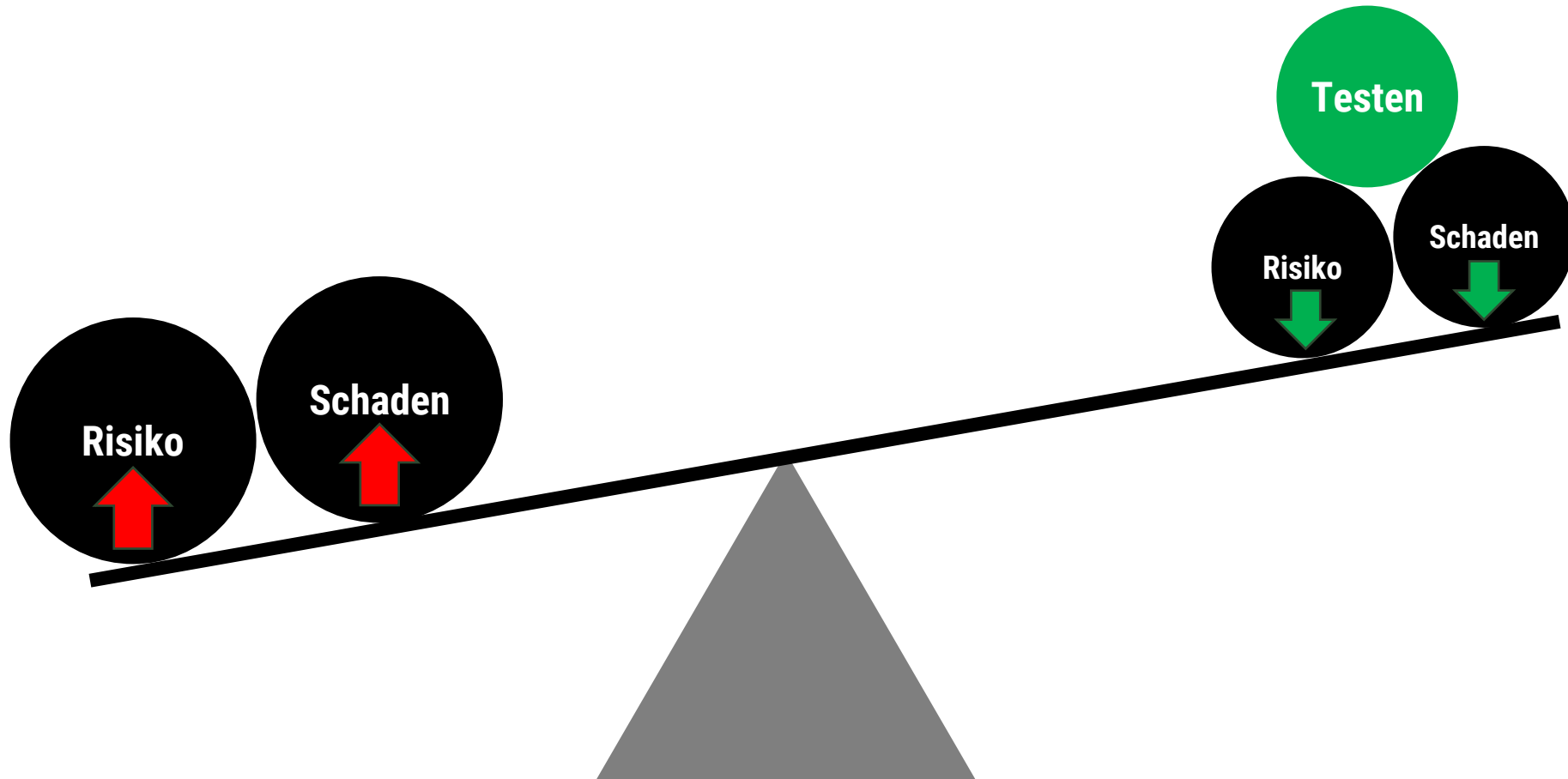
# Wie spart Testen Geld?



# Wie spart Testen Geld?



# Wie spart Testen Geld?



# Fehler-Beispiele

7:37 Sa., 20. Apr. 🔒 📶 🔊 🔋 11%

< 🔍 Wonach suchst du? 🎤



1/7

**-100%** UVP ~~3.699,38 €~~

**3,38 €**

inkl. MwSt. zzgl. Versandkosten

**Oder 0,31 € mtl. in 12 Raten**

April 2024  
Δ3.376,62 EUR

# Invalide Benutzereingabe

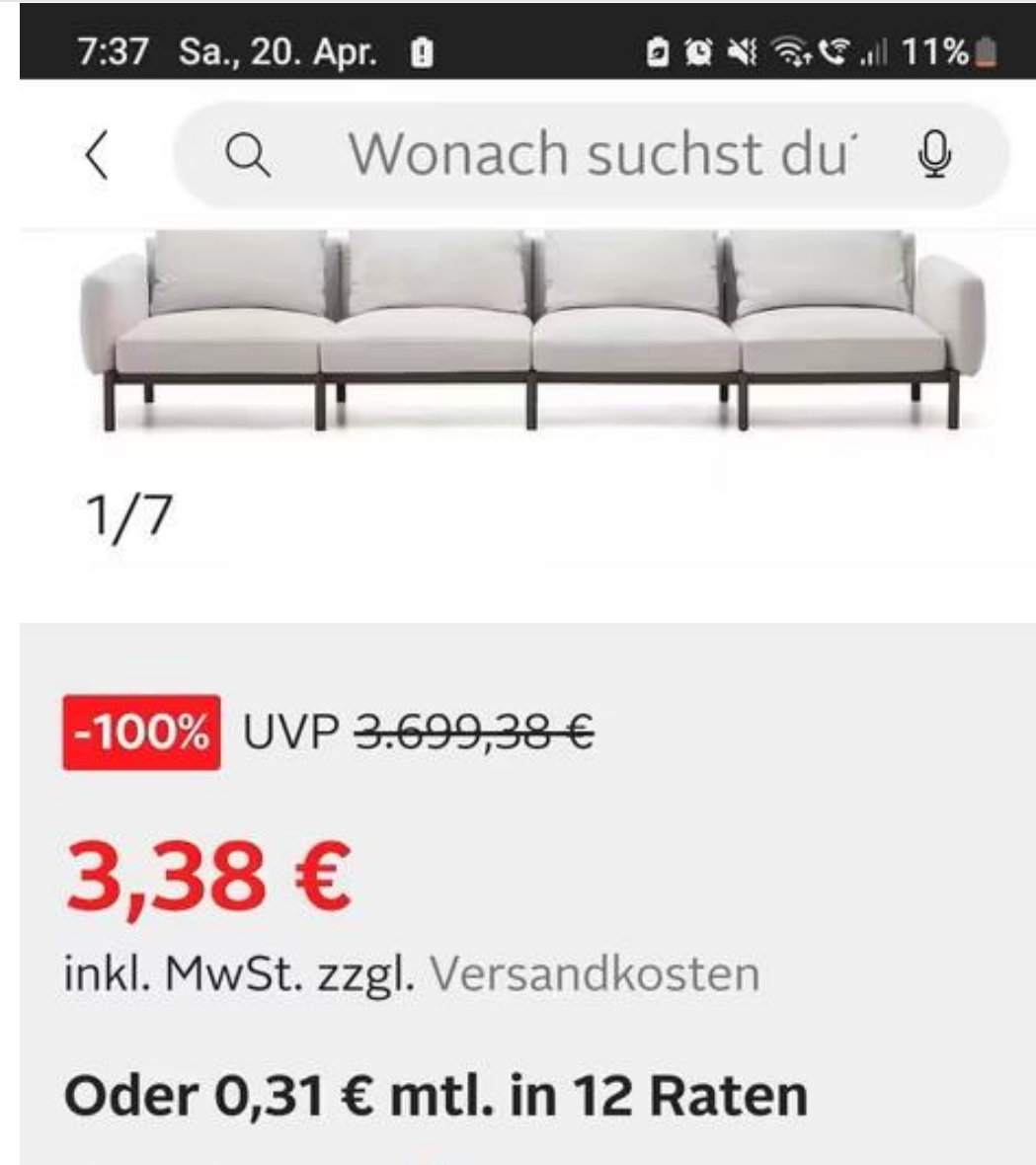
```

1  string input;
2
3  input = "3.380";
4  Console.WriteLine($"{decimal.Parse(input):N2}");
5
6  input = "3,380";
7  Console.WriteLine($"{decimal.Parse(input):N2}");

```

Microsoft Visual Studio-Debu

3.380,00  
3,38



# Datenbank Transaktion fehlerhaft

## Commercial Bank of Ethiopia glitch lets customers withdraw millions

18 March 2024

By Kalkidan Yibeltal, BBC News, Addis Ababa



More than 38 million people hold accounts at Commercial Bank Ethiopia

Ethiopia's biggest commercial bank is scrambling to recoup large sums of money withdrawn by customers after a "systems glitch".

More than \$40m (£31m) was withdrawn or transferred to other banks, local media reported.

März 2024  
ca. 37 mio. EUR

# Prozess hat logische Fehler

UNBERECHTIGTE AUSZAHLUNGEN

## Softwarefehler kostet Casinobetreiber Millionen

Mehrere Personen haben sich aufgrund eines Softwarefehlers zusammen **3,2 Millionen** australische Dollar auszahlen lassen, die ihnen eigentlich nicht zustanden.

18. April 2024, 9:27 Uhr, Marc Stöckel



(Bild: pixabay.com / Bru-n0)

April 2024  
ca. 2 mio. EUR

# Verlust von Daten

## Warning As **26 Billion** Records Leak: Dropbox, LinkedIn, Twitter Named

Davey Winder Senior Contributor



Jan 23, 2024, 08:15am EST

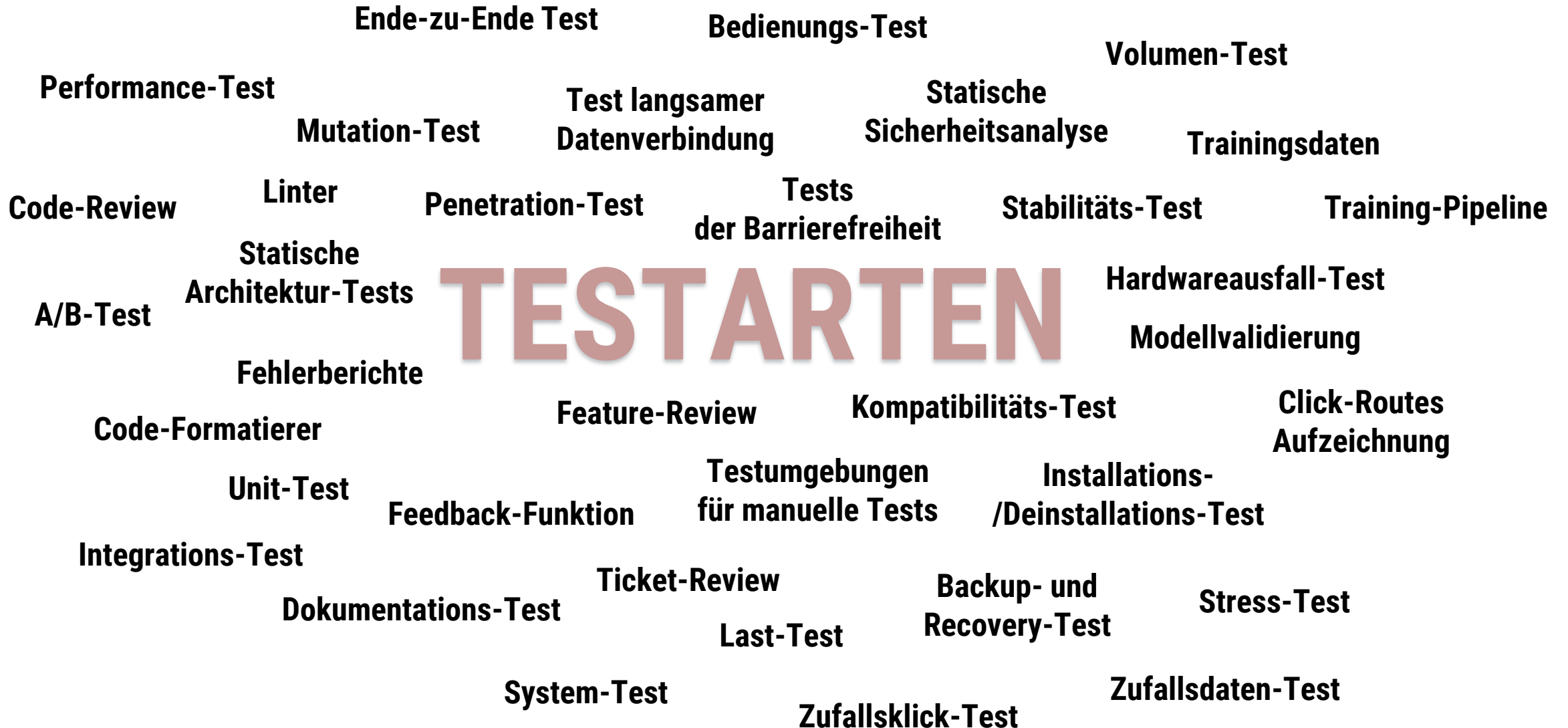
### Here's What You Need To Know

According to researchers from Security Discovery and CyberNews, the newly discovered database of leaked data runs to **12 terabytes** in size and deserves the MOAB title.

Jan. 2024

# Welche Testarten gibt es?

# Arten von Softwaretests



# Arten von Softwaretests

Funktionale Tests	Code-Qualität und Stilprüfungen	Kunden	Entwicklungsprozess	Performance- und Lasttests	Sicherheit	Stabilität	Data Science

# Arten von Softwaretests

Funktionale Tests	Code-Qualität und Stilprüfungen	Kunden	Entwicklungsprozess	Performance- und Lasttests	Sicherheit	Stabilität	Data Science
Unit-Test							

# Arten von Softwaretests

## Unit-Tests

```
5 [TestFixture]
  0 Verweise
6 public class PersonTests
7 {
8     [Test]
9     [TestCase(18, ExpectedResult = true)]
10    [TestCase(19, ExpectedResult = true)]
11    [TestCase(17, ExpectedResult = false)]
12    0 Verweise
13    public bool IstErwachsenTests(int alter)
14    {
15        var person = new Person { Alter = alter };
16        return person.IstErwachsen();
17    }
18 }
```

```
3 public sealed class Person
4 {
5     2 Verweise
6     public int Alter { get; set; }
7     1 Verweis
8     public bool IstErwachsen()
9     {
10        return Alter >= 18;
11    }
12 }
```

Mehr zu Unit-Tests...

## TDD: Von Frust zu Freude

Christoph Zuleger (Cronos), Wilhelm Kuhring (Cronos)

Mittwoch 14:15

# Arten von Softwaretests

Funktionale Tests	Code-Qualität und Stilprüfungen	Kunden	Entwicklungsprozess	Performance- und Lasttests	Sicherheit	Stabilität	Data Science
Unit-Test							
<b>Integrations-Test</b>							

# Arten von Softwaretests

Funktionale Tests	Code-Qualität und Stilprüfungen	Kunden	Entwicklungsprozess	Performance- und Lasttests	Sicherheit	Stabilität	Data Science
Unit-Test							
Integrations-Test							
<b>System-Test</b>							

# Arten von Softwaretests

Funktionale Tests	Code-Qualität und Stilprüfungen	Kunden	Entwicklungsprozess	Performance- und Lasttests	Sicherheit	Stabilität	Data Science
Unit-Test							
Integrations-Test							
System-Test							
<b>Ende-zu-Ende Test</b>							

Mehr zu Ende-zu-Ende-Tests...

## Einsatz von Emulatoren und Simulatoren für realitätsnahe Tests

Roland König

Mittwoch 18:15

# Arten von Softwaretests

Funktionale Tests	Code-Qualität und Stilprüfungen	Kunden	Entwicklungsprozess	P
Unit-Test				
Integrations-Test				
System-Test				
Ende-zu-Ende Test				
<b>Mutation-Test</b>				

## Stryker Mutator

Test your tests with mutation testing.



Mehr zu Mutation-Tests...

# Wir testen. Aber testen wir auch gut genug?

Patrick Drechsler (MATHEMA GmbH)

Donnerstag 18:15

# Arten von Softwaretests

Funktionale Tests	Code-Qualität und Stilprüfungen	Kunden	Entwicklungsprozess	Performance- und Lasttests	Sicherheit	Stabilität	Data Science
Unit-Test	Code-Formatierer						
Integrations-Test	Linter						
System-Test	Statische Architektur-Tests						
Ende-zu-Ende Test							
Mutation-Test							

# Arten von Softwaretests

Funktionale Tests	Code-Qualität und Stilprüfungen	Kunden	Entwicklungsprozess	Performance- und Lasttests	Sicherheit	Stabilität	Data Science
Unit-Test	Code-Formatierer	Fehlerberichte					
Integrations-Test	Linter	Feedback-Funktion					
System-Test	Statische Architektur-Tests	A/B-Test					
Ende-zu-Ende Test		Dokumentations-Test					
Mutation-Test		Click-Routes Aufzeichnung					
		Bedienungs-Test					
		Test langsamer Datenverbindung					
		Tests der Barrierefreiheit					

# Der Kunde als Tester...

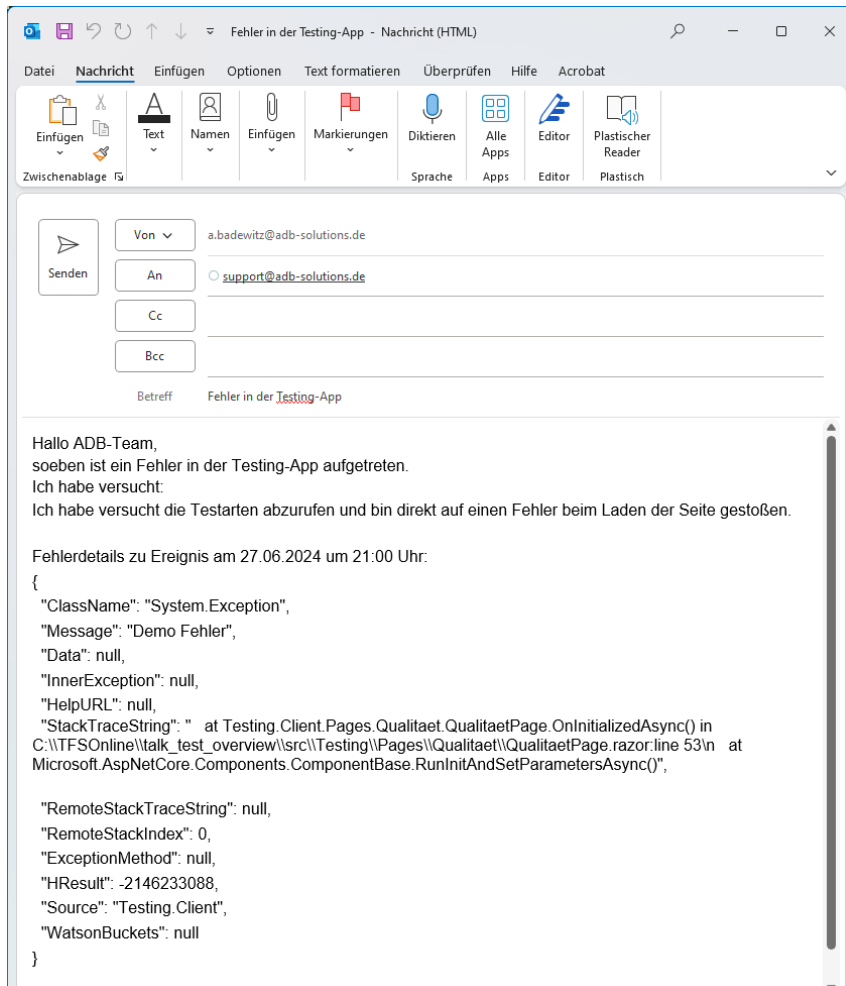


# Arten von Softwaretests

Funktionale Tests	Code-Qualität und Stilprüfungen	Kunden	Entwicklungsprozess	Performance- und Lasttests	Sicherheit	Stabilität	Data Science
Unit-Test	Code-Formatierer	Fehlerberichte					
Integrations-Test	Linter	Feedback-Funktion					
System-Test	Statische Architektur-Tests	A/B-Test					
Ende-zu-Ende Test		Dokumentations-Test					
Mutation-Test		Click-Routes Aufzeichnung					
		Bedienungs-Test					
		Test langsamer Datenverbindung					
		Tests der Barrierefreiheit					

# Arten von Softwaretests

## Fehlerberichte



## Fehler gefunden



Verzeigung, es ist ein Fehler am 25.03.2024 um 18:05 Uhr aufgetreten.



Fehlerbeschreibung: 'This is the message of a Demo-App-Exception'



Microsoft.AspNetCore.Components.EventCallbackWorkItem.InvokeAsync[Object](MulticastDelegate delegate, Object arg) Microsoft.AspNetCore.Components.EventCallbackWorkItem.I...

Bitte helfen Sie uns bei der Lösung dieses Problems und beschreiben kurz was passiert ist.

Fehlerbeschreibung

Ich habe versucht...

SENDEN



Für das Senden von Nachrichten benötigen Sie ein E-Mail-Programm auf Ihrem Gerät oder teilen Sie manuell einen Screenshot dieser Seite per E-Mail an support@email.de

Neuladen

Zur Startseite

# Arten von Softwaretests

Funktionale Tests	Code-Qualität und Stilprüfungen	Kunden	Entwicklungsprozess	Performance- und Lasttests	Sicherheit	Stabilität	Data Science
Unit-Test	Code-Formatierer	Fehlerberichte					
Integrations-Test	Linter	Feedback-Funktion					
System-Test	Statische Architektur-Tests	A/B-Test					
Ende-zu-Ende Test		Dokumentations-Test					
Mutation-Test		Click-Routes Aufzeichnung					
		Bedienungs-Test					
		Test langsamer Datenverbindung					
		Tests der Barrierefreiheit					

# Arten von Softwaretests

## A/B-Tests



Project name Home About Contact Dropdown - Default Static top Fixed top

Welcome to our website

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

[Learn more](#)

Click rate: **52 %**



Project name Home About Contact Dropdown - Default Static top Fixed top

Welcome to our website

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

[→ Learn more](#)

**72 %**

Mehr zu A/B-Tests...

# Test-Driven Leadership: Crafting Testable Strategies

Andrew Murphy

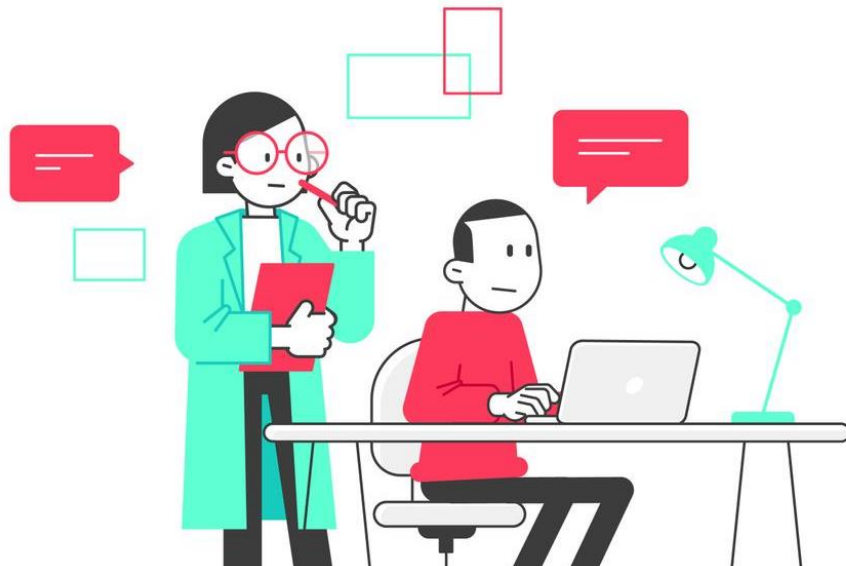
Mittwoch 15:30

# Arten von Softwaretests

Funktionale Tests	Code-Qualität und Stilprüfungen	Kunden	Entwicklungsprozess	Performance- und Lasttests	Sicherheit	Stabilität	Data Science
Unit-Test	Code-Formatierer	Fehlerberichte					
Integrations-Test	Linter	Feedback-Funktion					
System-Test	Statische Architektur-Tests	A/B-Test					
Ende-zu-Ende Test		Dokumentations-Test					
Mutation-Test		Click-Routes Aufzeichnung					
		<b>Bedienungs-Test</b>					
		Test langsamer Datenverbindung					
		Tests der Barrierefreiheit					

# Arten von Softwaretests

## Bedienungs-Tests



Mehr zu Bedienungs-Tests...

# Usability Tests. Kein Budget? Kein Problem!

Simon Eiterig (cronos)

Donnerstag 15:30

<https://www.hotjar.com/de/usability-tests/methoden/>

# Arten von Softwaretests

Funktionale Tests	Code-Qualität und Stilprüfungen	Kunden	Entwicklungsprozess	Performance- und Lasttests	Sicherheit	Stabilität	Data Science
Unit-Test	Code-Formatierer	Fehlerberichte					
Integrations-Test	Linter	Feedback-Funktion					
System-Test	Statische Architektur-Tests	A/B-Test					
Ende-zu-Ende Test		Dokumentations-Test					
Mutation-Test		Click-Routes Aufzeichnung					
		Bedienungs-Test					
		Test langsamer Datenverbindung					
		Tests der Barrierefreiheit					

# Arten von Softwaretests

Funktionale Tests	Code-Qualität und Stilprüfungen	Kunden	Entwicklungsprozess	Performance- und Lasttests	Sicherheit	Stabilität	Data Science
Unit-Test	Code-Formatierer	Fehlerberichte	Code-Review				
Integrations-Test	Linter	Feedback-Funktion	Ticket-Review				
System-Test	Statische Architektur-Tests	A/B-Test	Feature-Review				
Ende-zu-Ende Test		Dokumentations-Test	Testumgebungen für manuelle Tests				
Mutation-Test		Click-Routes Aufzeichnung					
		Bedienungs-Test					
		Test langsamer Datenverbindung					
		Tests der Barrierefreiheit					

# Arten von Softwaretests

Funktionale Tests	Code-Qualität und Stilprüfungen	Kunden	Entwicklungsprozess	Performance- und Lasttests	Sicherheit	Stabilität	Data Science
Unit-Test	Code-Formatierer	Fehlerberichte	Code-Review				
Integrations-Test	Linter	Feedback-Funktion	Ticket-Review				
System-Test	Statische Architektur-Tests	A/B-Test	Feature-Review				
Ende-zu-Ende Test		Dokumentations-Test	Testumgebungen für manuelle Tests				
Mutation-Test		Click-Routes Aufzeichnung					
		Bedienungs-Test					
		Test langsamer Datenverbindung					
		Tests der Barrierefreiheit					

# Arten von Softwaretests

## Testumgebungen für manuelle Tests

**Entwicklungs-Umgebung**  
Manueller Start mit einem  
Kommando

**Staging-Umgebung**  
on commit (release branch)

**Kunden-Test-Umgebung**  
manuell oder on commit

# Arten von Softwaretests

Funktionale Tests	Code-Qualität und Stilprüfungen	Kunden	Entwicklungsprozess	Performance- und Lasttests	Sicherheit	Stabilität	Data Science
Unit-Test	Code-Formatierer	Fehlerberichte	Code-Review	Performance-Test	Penetration-Test	Stabilitäts-Test	Trainingsdaten
Integrations-Test	Linter	Feedback-Funktion	Ticket-Review	Last-Test	Statische Sicherheitsanalyse	Hardwareausfall-Test	Training-Pipeline
System-Test	Statische Architektur-Tests	A/B-Test	Feature-Review	Stress-Test		Kompatibilitäts-Test	Modellvalidierung
Ende-zu-Ende Test		Dokumentations-Test	Testumgebungen für manuelle Tests	Volumen-Test		Installations-/Deinstallations-Test	
Mutation-Test		Click-Routes Aufzeichnung				Backup- und Recovery-Test	
		Bedienungs-Test				Zufallsdaten-Test	
		Test langsamer Datenverbindung				Zufallsklick-Test	
		Tests der Barrierefreiheit					

# Arten von Softwaretests

Funktionale Tests	Code-Qualität und Stilprüfungen	Kunden	Entwicklungsprozess	Performance- und Lasttests	Sicherheit	Stabilität	Data Science
Unit-Test	Code-Formatierer	Fehlerberichte	Code-Review	Performance-Test	Penetration-Test	Stabilitäts-Test	Trainingsdaten
Integrations-Test	Linter	Feedback-Funktion	Ticket-Review	Last-Test	Statische Sicherheitsanalyse	Hardwareausfall-Test	Training-Pipeline
System-Test	Statische Architektur-Tests	A/B-Test	Feature-Review	Stress-Test		Kompatibilitäts-Test	Modellvalidierung
Ende-zu-Ende Test		Dokumentations-Test	Testumgebungen für manuelle Tests	Volumen-Test		Installations-/Deinstallations-Test	
Mutation-Test		Click-Routes Aufzeichnung				Backup- und Recovery-Test	
		Bedienungs-Test				Zufallsdaten-Test	
		Test langsamer Datenverbindung				Zufallsklick-Test	
		Tests der Barrierefreiheit					

# Arten von Softwaretests

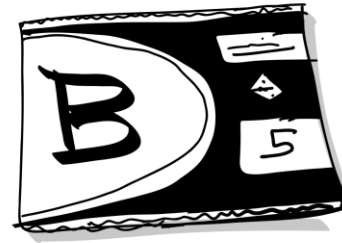
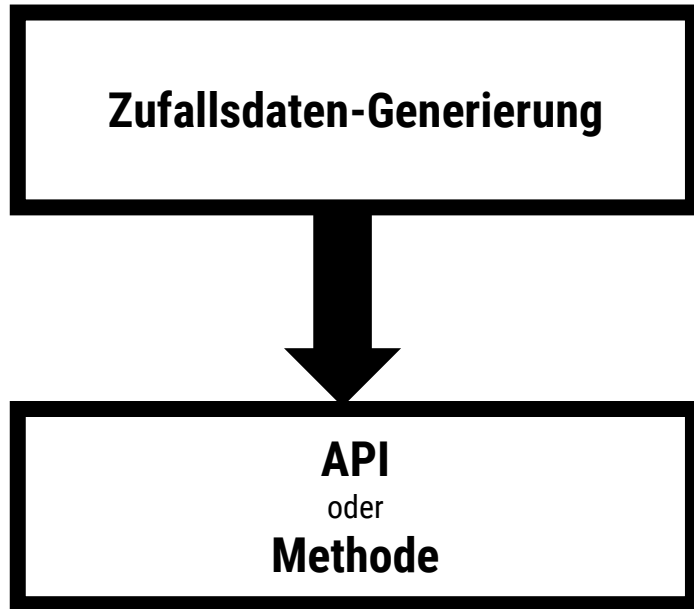
Funktionale Tests	Code-Qualität und Stilprüfungen	Kunden	Entwicklungsprozess	Performance- und Lasttests	Sicherheit	Stabilität	Data Science
Unit-Test	Code-Formatierer	Fehlerberichte	Code-Review	Performance-Test	Penetration-Test	Stabilitäts-Test	Trainingsdaten
Integrations-Test	Linter	Feedback-Funktion	Ticket-Review	Last-Test	Statische Sicherheitsanalyse	Hardwareausfall-Test	Training-Pipeline
System-Test	Statische Architektur-Tests	A/B-Test	Feature-Review	Stress-Test		Kompatibilitäts-Test	Modellvalidierung
Ende-zu-Ende Test		Dokumentations-Test	Testumgebungen für manuelle Tests	Volumen-Test		Installations-/Deinstallations-Test	
Mutation-Test		Click-Routes Aufzeichnung				Backup- und Recovery-Test	
		Bedienungs-Test				Zufallsdaten-Test	
		Test langsamer Datenverbindung				Zufallsklick-Test	
		Tests der Barrierefreiheit					

# Arten von Softwaretests

Funktionale Tests	Code-Qualität und Stilprüfungen	Kunden	Entwicklungsprozess	Performance- und Lasttests	Sicherheit	Stabilität	Data Science
Unit-Test	Code-Formatierer	Fehlerberichte	Code-Review	Performance-Test	Penetration-Test	Stabilitäts-Test	Trainingsdaten
Integrations-Test	Linter	Feedback-Funktion	Ticket-Review	Last-Test	Statische Sicherheitsanalyse	Hardwareausfall-Test	Training-Pipeline
System-Test	Statische Architektur-Tests	A/B-Test	Feature-Review	Stress-Test		Kompatibilitäts-Test	Modellvalidierung
Ende-zu-Ende Test		Dokumentations-Test	Testumgebungen für manuelle Tests	Volumen-Test		Installations-/Deinstallations-Test	
Mutation-Test		Click-Routes Aufzeichnung				Backup- und Recovery-Test	
		Bedienungs-Test				Zufallsdaten-Test	
		Test langsamer Datenverbindung				Zufallsklick-Test	
		Tests der Barrierefreiheit					

# Arten von Softwaretests

## Zufallsdaten-Test



*Bogus*

CSANDUN BLOGS

#	Entity	Properties
1	Address	ZipCode, City, Country, Latitude, etc
2	Commerce	Department name, ProductName, ProductAdjective, Price, etc
3	Company	CompanyName, CatchPhrase, Bs, etc
4	Database	Column, Type,Collation etc
5	Date	Past, Soon, Between, etc
6	Finance	Account number, TransactionType, Currency, CreditCardNumber, etc
7	Hacker	Abbreviation ,Adjective etc
8	Images	Random image, Animals image, Nature image, etc.
9	Internet	Email, DomainName, Ipv6, Password, et
10	Lorem	single word, Words, Sentence, Paragraphs, etc.
11	Name	FirstName, LastName, etc
12	Phone	PhoneNumber , PhoneNumberFormat
13	Rant	Random user review, etc
14	System	FileName, MimeType, FileExt, etc.
15	Vehicle	Manufacturer , Model , etc
16	Random/Randomizer	Number, Digits, Hash, String, etc

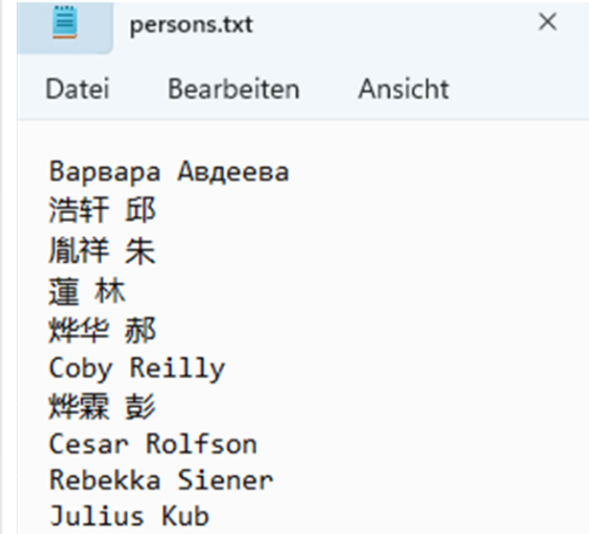
<https://github.com/bchavez/Bogus>

<https://csandunblogs.com/sane-fake-data-generator-for-net-bogus-open-source-libraries-2>

# Arten von Softwaretests

## Zufallsdaten-Test

```
1 using Bogus;
2
3 for (int i = 0; i < 10; i++)
4 {
5     var faker = FakerFactory.CreateRandomFaker("de", "en", "es", "zh_CN", "ru", "ja");
6     File.AppendAllLines("persons.txt", new List<string>() { faker.Name.FullName() });
7 }
8
9 public static class FakerFactory
10 {
11     public static Faker CreateRandomFaker(params string[] locales)
12     {
13         var random = new Random();
14         var randomLocale = locales[random.Next(locales.Length)];
15         return new Faker(randomLocale);
16     }
17 }
18
```



persons.txt

Datei Bearbeiten Ansicht

Barbara Авдеева  
浩轩 邱  
胤祥 朱  
蓮 林  
烨华 郝  
Coby Reilly  
烨霖 彭  
Cesar Rolfson  
Rebekka Siener  
Julius Kub

# Arten von Softwaretests

## Zufallsdaten-Test

```
var userIds = 0;  
var testUsers = new Faker<User>()  
    .RuleFor(u => u.Id, f => userIds++)  
    .RuleFor(u => u.Gender, f => f.PickRandom<Bogus.DataSets.Name.Gender>())  
    .RuleFor(u => u.FirstName, (f, u) => f.Name.FirstName(u.Gender))  
    .RuleFor(u => u.LastName, (f, u) => f.Name.LastName(u.Gender))  
    .RuleFor(u => u.UserName, (f, u) => f.Internet.UserName(u.FirstName, u.LastName))  
    .RuleFor(u => u.Email, (f, u) => f.Internet.Email(u.FirstName, u.LastName))  
    .RuleFor(u => u.CartId, f => f.PickRandom(guids))  
    .RuleFor(u => u.Orders, f => [..testOrders.Generate(3)])  
    .UseSeed(123);
```



**Generate any amount  
of fake data**

```
var users = testUsers.Generate(1000);
```

# Arten von Softwaretests

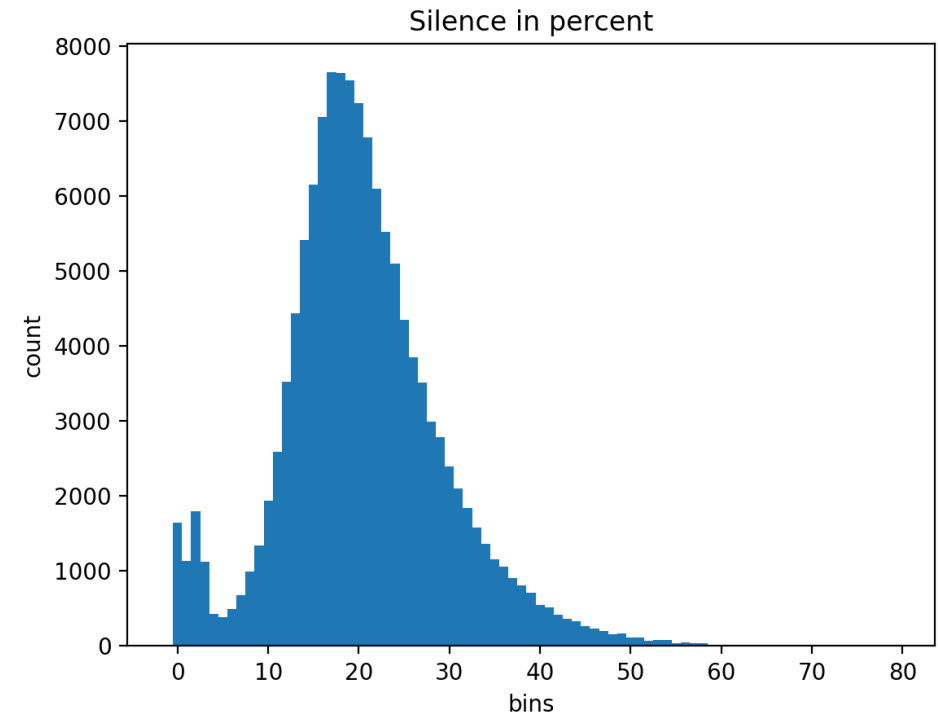
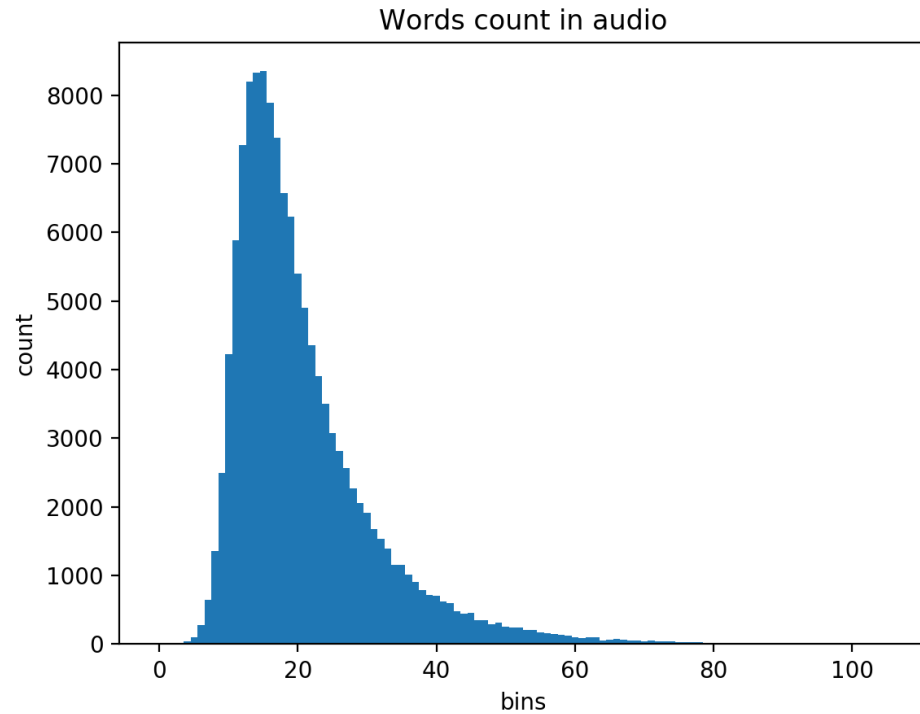
Funktionale Tests	Code-Qualität und Stilprüfungen	Kunden	Entwicklungsprozess	Performance- und Lasttests	Sicherheit	Stabilität	Data Science
Unit-Test	Code-Formatierer	Fehlerberichte	Code-Review	Performance-Test	Penetration-Test	Stabilitäts-Test	Trainingsdaten
Integrations-Test	Linter	Feedback-Funktion	Ticket-Review	Last-Test	Statische Sicherheitsanalyse	Hardwareausfall-Test	Training-Pipeline
System-Test	Statische Architektur-Tests	A/B-Test	Feature-Review	Stress-Test		Kompatibilitäts-Test	Modellvalidierung
Ende-zu-Ende Test		Dokumentations-Test	Testumgebungen für manuelle Tests	Volumen-Test		Installations-/Deinstallations-Test	
Mutation-Test		Click-Routes Aufzeichnung				Backup- und Recovery-Test	
		Bedienungs-Test				Zufallsdaten-Test	
		Test langsamer Datenverbindung				Zufallsklick-Test	
		Tests der Barrierefreiheit					

# Arten von Softwaretests

Funktionale Tests	Code-Qualität und Stilprüfungen	Kunden	Entwicklungsprozess	Performance- und Lasttests	Sicherheit	Stabilität	Data Science
Unit-Test	Code-Formatierer	Fehlerberichte	Code-Review	Performance-Test	Penetration-Test	Stabilitäts-Test	Trainingsdaten
Integrations-Test	Linter	Feedback-Funktion	Ticket-Review	Last-Test	Statische Sicherheitsanalyse	Hardwareausfall-Test	Training-Pipeline
System-Test	Statische Architektur-Tests	A/B-Test	Feature-Review	Stress-Test		Kompatibilitäts-Test	Modellvalidierung
Ende-zu-Ende Test		Dokumentations-Test	Testumgebungen für manuelle Tests	Volumen-Test		Installations-/Deinstallations-Test	
Mutation-Test		Click-Routes Aufzeichnung				Backup- und Recovery-Test	
		Bedienungs-Test				Zufallsdaten-Test	
		Test langsamer Datenverbindung				Zufallsklick-Test	
		Tests der Barrierefreiheit					

# Arten von Softwaretests

## Trainingsdaten



# Arten von Softwaretests

Funktionale Tests	Code-Qualität und Stilprüfungen	Kunden	Entwicklungsprozess	Performance- und Lasttests	Sicherheit	Stabilität	Data Science
Unit-Test	Code-Formatierer	Fehlerberichte	Code-Review	Performance-Test	Penetration-Test	Stabilitäts-Test	Trainingsdaten
Integrations-Test	Linter	Feedback-Funktion	Ticket-Review	Last-Test	Statische Sicherheitsanalyse	Hardwareausfall-Test	Training-Pipeline
System-Test	Statische Architektur-Tests	A/B-Test	Feature-Review	Stress-Test		Kompatibilitäts-Test	Modellvalidierung
Ende-zu-Ende Test		Dokumentations-Test	Testumgebungen für manuelle Tests	Volumen-Test		Installations-/Deinstallations-Test	
Mutation-Test		Click-Routes Aufzeichnung				Backup- und Recovery-Test	
		Bedienungs-Test				Zufallsdaten-Test	
		Test langsamer Datenverbindung				Zufallsklick-Test	
		Tests der Barrierefreiheit					

**Welche Art von**

**Softwaretests**

**brauche ich wirklich?**

# Warum testen wir?

**Meistens...**

**... um Geld**

**zu sparen**



# Die Lösung?

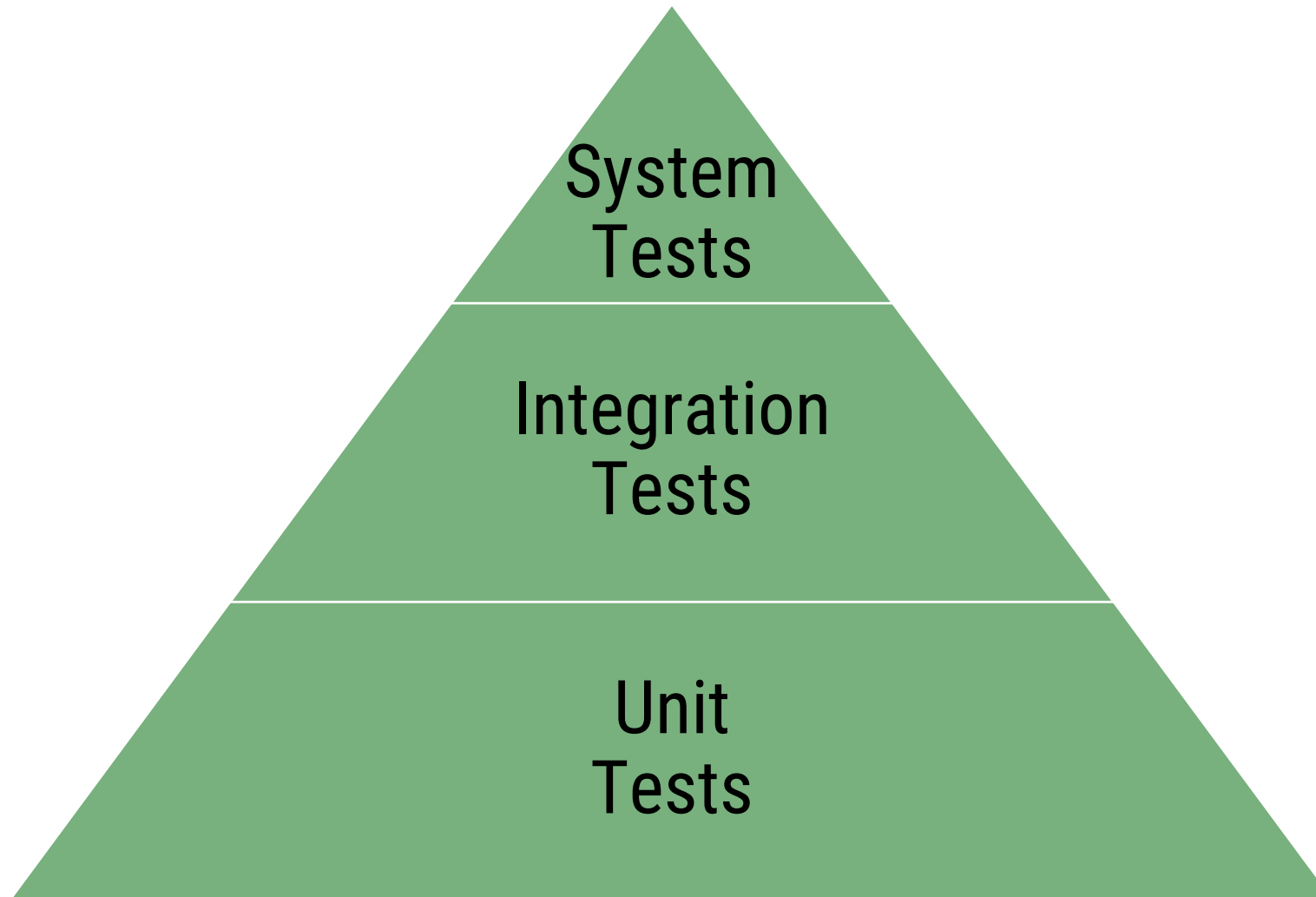
**Welche Art von**

**Softwaretests**

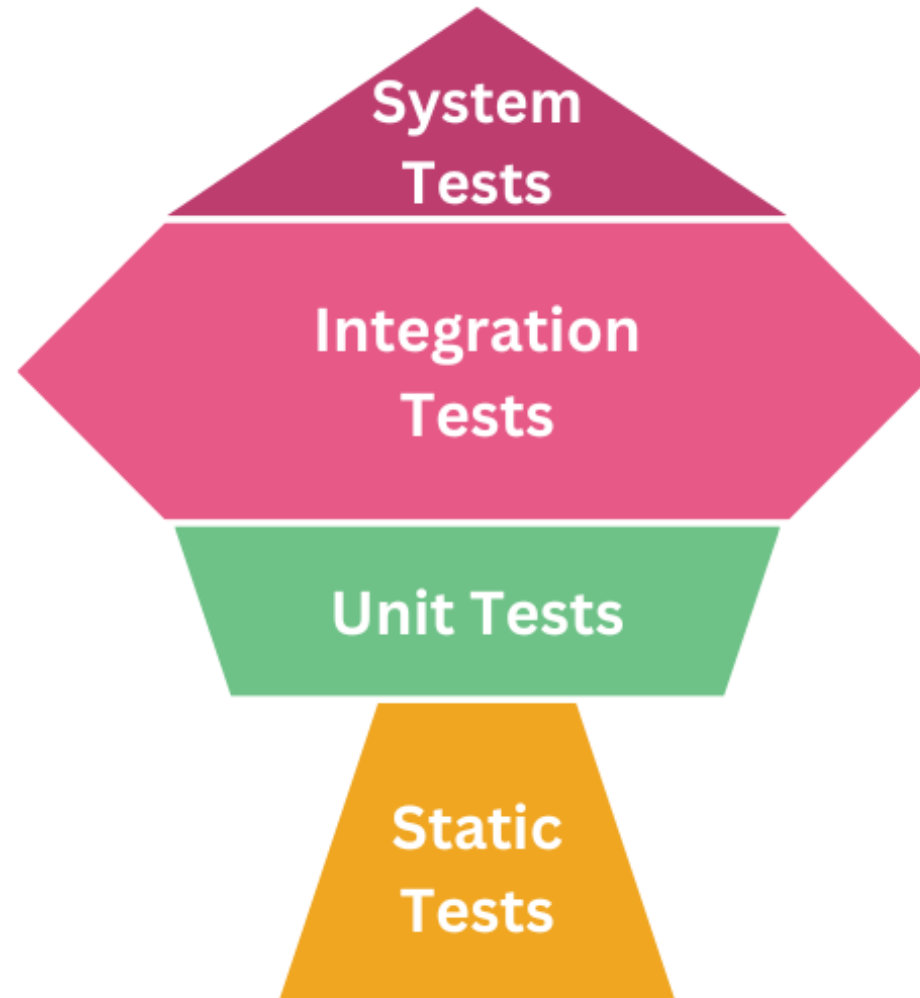
**brauche ich wirklich?**



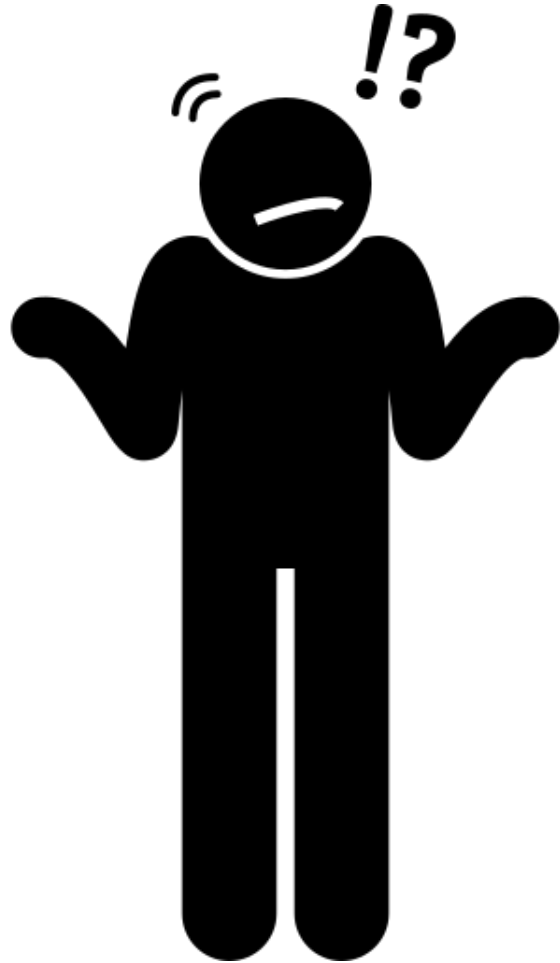
# Lösung = Testpyramide?



# Lösung = Testtrophäe?



# Es gibt keine einfache Lösung...



## Warum?



**... leider.**

# Unterschiedlichkeiten

🏠 Gebäudenr. 4214 / 🏢 EG 📄 ← Z.

IDs 03-42-4214 | ---

Liegenschaft auswählen

03-42

Gebäude auswählen

4214 Zentrales Hörsaalgebäude

Geschoss auswählen

EG

Raum auswählen ↻

+ Neuer Raum

🔍 Suche

🗨️ 0 🗑️

Türschild-Nr. HO01 + HINWEIS 🗨️

Nutzungsart: **Flure, Hallen**

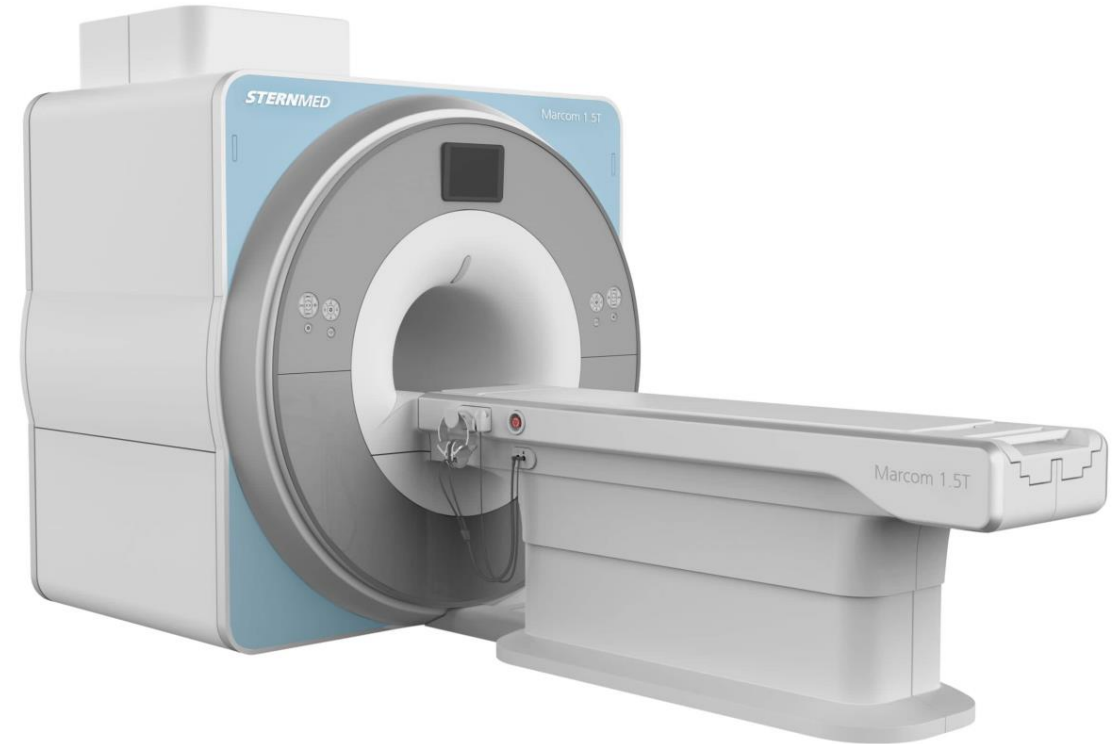
Bezeichnung: **Foyer (Steinboden)**

Boden: **Fliesen**

Aufzug bearbeiten

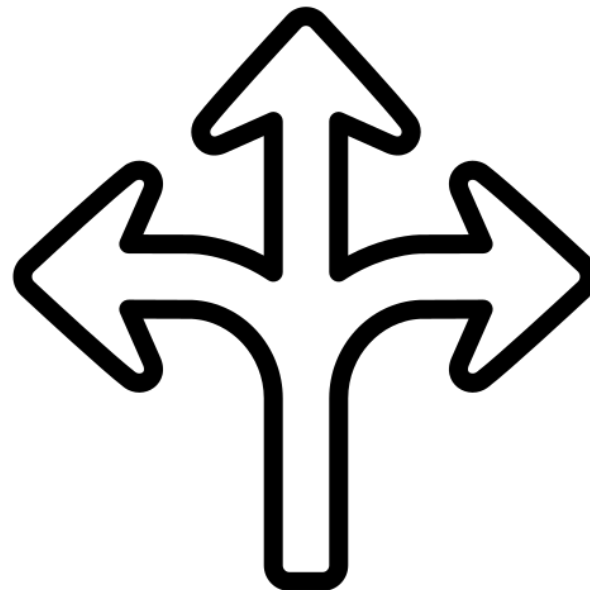
📘 Weder Türen, Fenster noch Krananlagen erfasst

vs.



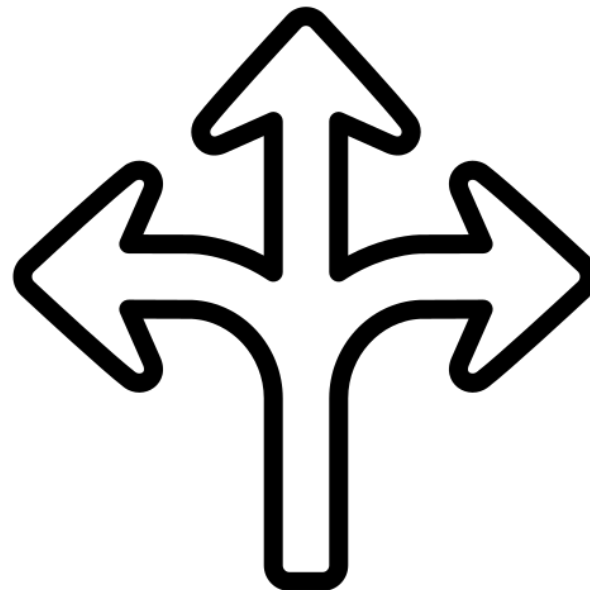
<https://sternmed.de/medizingeraete/mrt-system-marcom-1-5t/>

# Drei Möglichkeiten, die richtige Art von Softwaretests zu finden



# Drei Möglichkeiten, die richtige Art von Softwaretests zu finden

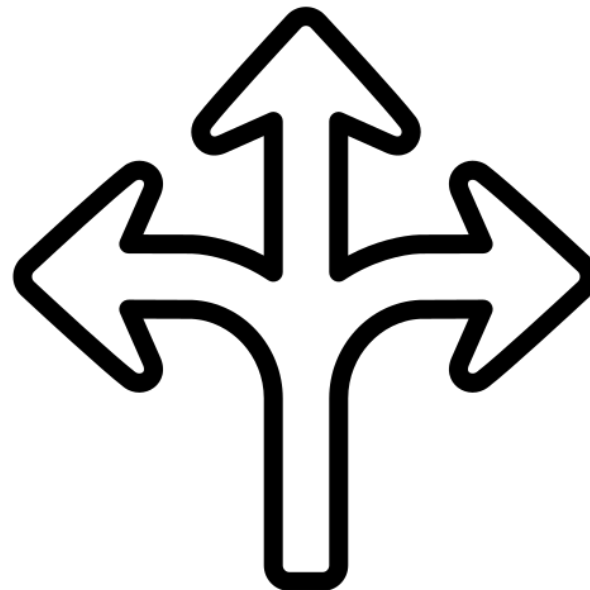
Erfahrung/Intuition



# Drei Möglichkeiten, die richtige Art von Softwaretests zu finden

Erfahrung/Intuition

Risikoanalyse



# Risiko

		Wahrscheinlich				
Schaden						

# Risiko = Wahrscheinlichkeit \* Schaden

		Wahrscheinlich				
		1 Selten	2 Unwahrscheinlich	3 Möglich	4 Wahrscheinlich	5 Fast sicher
Schaden	5 Katastrophal					
	4 Groß					
	3 Moderat					
	2 Geringfügig					
	1 Vernachlässigbar					

# Risiko = Wahrscheinlichkeit \* Schaden

		Wahrscheinlich				
		1 Selten	2 Unwahrscheinlich	3 Möglich	4 Wahrscheinlich	5 Fast sicher
Schaden	5 Katastrophal	5	10	15	20	25
	4 Groß	4	8	12	16	20
	3 Moderat	3	6	9	12	15
	2 Geringfügig	2	2	6	8	10
	1 Vernachlässigbar	1	2	3	4	5

Risiko =  **Niedrig**  **Mittel**  **Hoch**  **Sehr hoch**

# Beispiel

# Prozess hat logische Fehler

UNBERECHTIGTE AUSZAHLUNGEN

## Softwarefehler kostet Casinobetreiber Millionen

Mehrere Personen haben sich aufgrund eines Softwarefehlers zusammen **3,2 Millionen** australische Dollar auszahlen lassen, die ihnen eigentlich nicht zustanden.

18. April 2024, 9:27 Uhr, Marc Stöckel



(Bild: pixabay.com / Bru-n0)

Spielautomaten in einem Casino

<https://www.golem.de/news/unberechtigte-auszahlungen-softwarefehler-kostet-casinobetreiber-millionen-2404-184286.html>

# Risiko = Wahrscheinlichkeit \* Schaden

		Wahrscheinlich				
		1 Selten	2 Unwahrscheinlich	3 Möglich	4 Wahrscheinlich	5 Fast sicher
Schaden	5 Katastrophal					
	4 Groß					
	3 Moderat					
	2 Geringfügig					
	1 Vernachlässigbar					

Risiko =  **Niedrig**  **Mittel**  **Hoch**  **Sehr hoch**

# Risiko = Wahrscheinlichkeit \* Schaden

		Wahrscheinlich				
		1 Selten	2 Unwahrscheinlich	3 Möglich	4 Wahrscheinlich	5 Fast sicher
Schaden	5 Katastrophal					
	4 Groß					
	3 Moderat					
	2 Geringfügig					
	1 Vernachlässigbar					

Risiko =  **Niedrig**  **Mittel**  **Hoch**  **Sehr hoch**

# Risiko = Wahrscheinlichkeit \* Schaden

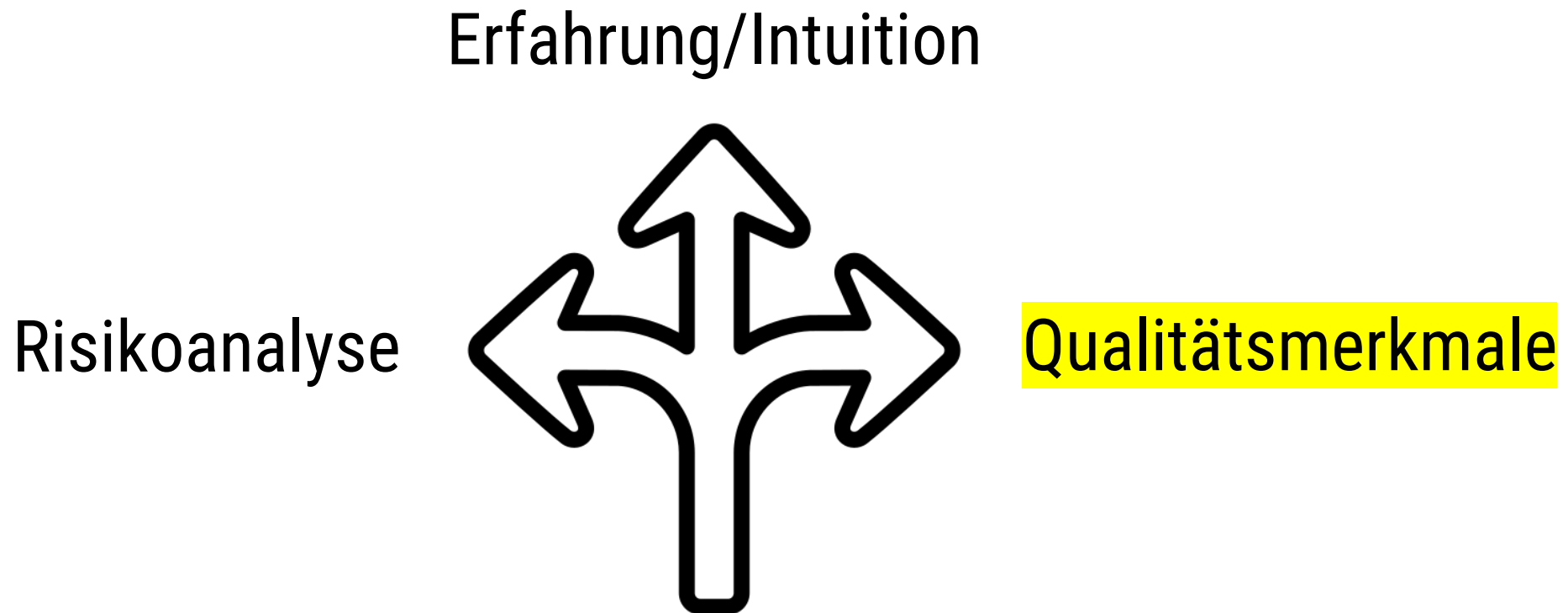
		Wahrscheinlich				
		1 Selten	2 Unwahrscheinlich	3 Möglich	4 Wahrscheinlich	5 Fast sicher
Schaden	5 Katastrophal			15		
	4 Groß					
	3 Moderat					
	2 Geringfügig					
	1 Vernachlässigbar					

Risiko =  **Niedrig**  **Mittel**  **Hoch**  **Sehr hoch**

# Risikoanalyse

Name	Wahrscheinlich	*	Schaden	=	Risiko
Invalide Benutzereingabe	2 Unwahrscheinlich	*	5 Katastrophal	=	10
Prozess hat logische Fehler	3 Möglich	*	5 Katastrophal	=	15
Datenbank Transaktion fehlerhaft	3 Möglich	*	5 Katastrophal	=	15
Verlust von Daten	3 Möglich	*	5 Katastrophal	=	15

# Drei Möglichkeiten, die richtige Art von Softwaretests zu finden



# Qualitätsmerkmale und -kriterien nach ISO 25010

Funktionale Eignung	Zuverlässigkeit	Benutzbarkeit/Usability	Leistungseffizienz	Kompatibilität	Wartbarkeit	Informationssicherheit	Übertragbarkeit	Betriebssicherheit
Vollständigkeit	Ausgereiftheit	Erkennbarkeit	Zeitverhalten	Koexistenz	Modularität	Vertraulichkeit	Anpassungsfähigkeit	Betriebseinschränkung
Korrektheit	Verfügbarkeit	Erlernbarkeit	Ressourcennutzung	Interoperabilität	Wiederverwendbarkeit	Integrität	Skalierbarkeit	Risikoerkennung
Angemessenheit	Fehlertoleranz	Bedienbarkeit	Kapazität		Analysierbarkeit	Nachweisbarkeit	Installierbarkeit	Ausfallsicherheit
	Wiederherstellbarkeit	Toleranz gegenüber Fehlbedienung			Modifizierbarkeit	Nachvollziehbarkeit	Austauschbarkeit	Gefahrenwarnung
		Ästhetik			Testbarkeit	Authentizität		Sichere Integration
		Barrierefreiheit				Widerstandsfähigkeit		
		Benutzerunterstützend						
		Selbsterklärend						

# Beispiel

# Merkmale und Kriterien Einfache Dateneingabe

- **Funktionale Eignung**
  - Korrektheit (Ziel)
- **Zuverlässigkeit**
  - Ausgereiftheit (Fehlerfrei u. Normalbedingung)
- **Benutzbarkeit/Usability**
  - Bedienbarkeit
  - Toleranz gegenüber Fehlbedienung

🏠 Gebäudenr. 4214 / 📄 EG 📄

IDs 03-42-4214 | ---

Liegenschaft auswählen

03-42

Gebäude auswählen

4214 Zentrales Hörsaalgebäude

Geschoss auswählen

EG

Raum auswählen

+ Neuer Raum

🔍 Suche

Türschild-Nr. HO01

+ HINWEIS

Nutzungsart: **Flure, Hallen**

Bezeichnung: **Foyer (Steinboden)**

Boden: **Fliesen**

Aufzug bearbeiten

📄 Weder Türen, Fenster noch Krananlagen erfasst

# Merkmale und Kriterien

## MRT

- **Funktionale Eignung**

- Korrektheit

- **Zuverlässigkeit**

- Ausgereiftheit
- Verfügbarkeit
- Fehlertoleranz

- **Benutzbarkeit/Usability**

- Erlernbarkeit
- Bedienbarkeit
- Toleranz gegenüber Fehlbedienung

- **Leistungseffizienz**

- Zeitverhalten

- **Kompatibilität**

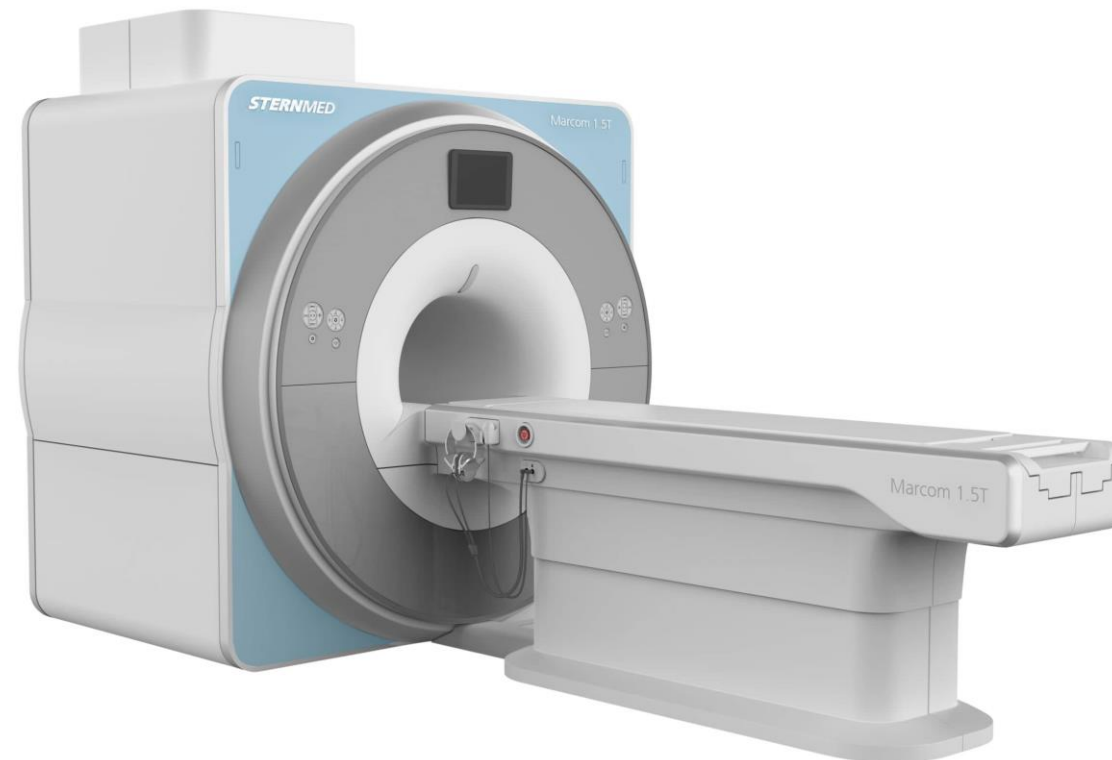
- Interoperabilität

- **Wartbarkeit**

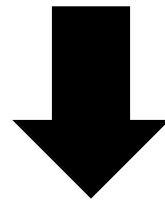
- Modularität
- Analysierbarkeit
- Testbarkeit

- **Betriebssicherheit**

- Betriebseinschränkung
- Risikoerkennung
- Ausfallsicherheit
- Gefahrenwarnung
- Sichere Integration



# Qualitätskriterien



# geeignete Testarten

# Welche Art von Softwaretests brauche ich wirklich?

## Qualitätsmerkmale und -kriterien nach ISO 25010

Bitte gewünschte Kriterien selektieren:

<p><b>Funktionale Eignung</b> 0/3 Kriterien</p> <p>Vollständigkeit ?</p> <p>Korrektheit ?</p> <p>Angemessenheit ?</p>	<p><b>Zuverlässigkeit</b> 0/4 Kriterien</p> <p>Ausgereiftheit ?</p> <p>Verfügbarkeit ?</p> <p>Fehlertoleranz ?</p> <p>Wiederherstellbarkeit ?</p>	<p><b>Benutzbarkeit/Usability</b> 0/8 Kriterien</p> <p>Erkennbarkeit ?</p> <p>Erlernbarkeit ?</p> <p>Bedienbarkeit ?</p> <p>Toleranz gegenüber Fehlbedienung ?</p> <p>Ästhetik ?</p> <p>Barrierefreiheit ?</p> <p>Benutzerunterstützend ?</p> <p>Selbsterklärend ?</p>	<p><b>Leistungseffizienz</b> 0/3 Kriterien</p> <p>Zeitverhalten ?</p> <p>Ressourcennutzung ?</p> <p>Kapazität ?</p>	<p><b>Kompatibilität</b> 0/2 Kriterien</p> <p>Koexistenz ?</p> <p>Interoperabilität ?</p>	<p><b>Wartbarkeit</b> 0/5 Kriterien</p> <p>Modularität ?</p> <p>Wiederverwendbarkeit ?</p> <p>Analysierbarkeit ?</p> <p>Modifizierbarkeit ?</p> <p>Testbarkeit ?</p>
<p><b>Informationssicherheit</b> 0/6 Kriterien</p> <p>Vertraulichkeit ?</p> <p>Integrität ?</p> <p>Nachweisbarkeit ?</p> <p>Nachvollziehbarkeit ?</p> <p>Authentizität ?</p> <p>Widerstandsfähigkeit ?</p>	<p><b>Übertragbarkeit</b> 0/4 Kriterien</p> <p>Anpassungsfähigkeit ?</p> <p>Skalierbarkeit ?</p> <p>Installierbarkeit ?</p> <p>Austauschbarkeit ?</p>	<p><b>Betriebssicherheit</b> 0/5 Kriterien</p> <p>Betriebseinschränkung ?</p> <p>Risikoerkennung ?</p> <p>Ausfallsicherheit ?</p> <p>Gefahrenwarnung ?</p> <p>Sichere Integration ?</p>			

# Merkmale und Kriterien Einfache Dateneingabe

- Funktionale Eignung
  - Korrektheit
- Zuverlässigkeit
  - Ausgereiftheit
- Benutzbarkeit/Usability
  - Bedienbarkeit
  - Toleranz gegenüber Fehlbedienung

🏠 Gebäudenr. 4214 / 📄 EG 📄 ← Z.

IDs 03-42-4214 | ---

Liegenschaft auswählen

03-42

Gebäude auswählen

4214 Zentrales Hörsaalgebäude

Geschoss auswählen

EG

Raum auswählen ↻

+ Neuer Raum

🔍 Suche

🗑️

🗨️ 0 Türschild-Nr. HO01 🗑️

+ HINWEIS 🗨️

Nutzungsart: **Flure, Hallen**

Bezeichnung: **Foyer (Steinboden)**

Boden: **Fliesen**

Aufzug bearbeiten

i Weder Türen, Fenster noch Krananlagen erfasst

# Qualitätsmerkmale und -kriterien nach ISO 25010

Bitte gewünschte Kriterien selektieren

## Funktionale Eignung

1/3 Kriterien

Vollständigkeit ?

**Korrektheit** ?

Angemessenheit ?

## Zuverlässigkeit

1/4 Kriterien

**Ausgereiftheit** ?

Verfügbarkeit ?

Fehlertoleranz ?

Wiederherstellbarkeit ?

## Benutzbarkeit/Usability

2/8 Kriterien

Erkennbarkeit ?

Erlernbarkeit ?

**Bedienbarkeit** ?

**Toleranz gegenüber  
Fehlbedienung** ?

Ästhetik ?

Barrierefreiheit ?

Benutzerunterstützend ?

## Leistungseffizienz

0/3 Kriterien

Zeitverhalten ?

Ressourcennutzung ?

Kapazität ?

## Kompatibilität

0/2 Kriterien

Koexistenz ?

Interoperabilität ?

## Geeignete Testarten

### Code-Qualität und Stilprüfungen

Linters

### Funktionale Tests

Unit-Test

Integrations-Test

System-Test

Mutation-Test

### Kunden

A/B-Test

Fehlerberichte

Click-Routes Aufzeichnung

Feedback-Funktion

Test langsamer Datenverbindung

Bedienungs-Test

### Entwicklungsprozess

Code-Review (Pull-Request)

Ticket-Review

Feature-Review

Testumgebungen für manuelle Tests

### Stabilität

Stabilitäts-Test

Zufallsdaten-Test

Zufallsklick-Test

# Merkmale und Kriterien

## MRT

- **Funktionale Eignung**

- Korrektheit

- **Zuverlässigkeit**

- Ausgereiftheit
- Verfügbarkeit
- Fehlertoleranz

- **Benutzbarkeit/Usability**

- Erlernbarkeit
- Bedienbarkeit
- Toleranz gegenüber Fehlbedienung

- **Leistungseffizienz**

- Zeitverhalten

- **Kompatibilität**

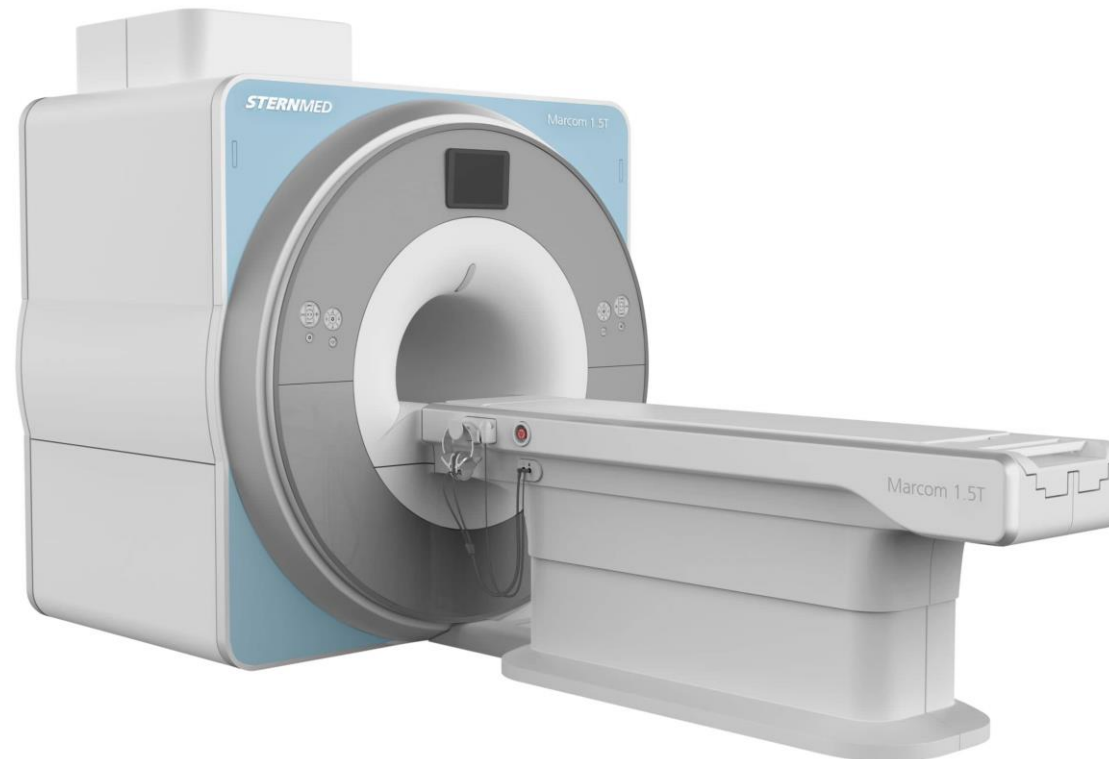
- Interoperabilität

- **Wartbarkeit**

- Modularität
- Analysierbarkeit
- Testbarkeit

- **Betriebssicherheit**

- Betriebseinschränkung
- Risikoerkennung
- Ausfallsicherheit
- Gefahrenwarnung
- Sichere Integration



## Qualitätsmerkmale und -kriterien nach ISO 25010

Bitte gewünschte Kriterien selektieren

Funktionale Eignung	Zuverlässigkeit	Benutzbarkeit/Usability	Leistungseffizienz	Kompatibilität	Wartbarkeit	Betriebssicherheit
1/3 Kriterien	3/4 Kriterien	3/8 Kriterien	1/3 Kriterien	1/2 Kriterien	3/5 Kriterien	5/5 Kriterien
<input type="checkbox"/> Vollständigkeit	<input checked="" type="checkbox"/> Ausgereiftheit	<input type="checkbox"/> Erkennbarkeit	<input checked="" type="checkbox"/> Zeitverhalten	<input type="checkbox"/> Koexistenz	<input checked="" type="checkbox"/> Modularität	<input checked="" type="checkbox"/> Betriebseinschränkung
<input checked="" type="checkbox"/> Korrektheit	<input checked="" type="checkbox"/> Verfügbarkeit	<input checked="" type="checkbox"/> Erlernbarkeit	<input type="checkbox"/> Ressourcennutzung	<input checked="" type="checkbox"/> Interoperabilität	<input type="checkbox"/> Wiederverwendbarkeit	<input checked="" type="checkbox"/> Risikoerkennung
<input type="checkbox"/> Angemessenheit	<input checked="" type="checkbox"/> Fehlertoleranz	<input checked="" type="checkbox"/> Bedienbarkeit	<input type="checkbox"/> Kapazität		<input type="checkbox"/> Analysierbarkeit	<input checked="" type="checkbox"/> Ausfallsicherheit
	<input type="checkbox"/> Wiederherstellbarkeit	<input checked="" type="checkbox"/> Toleranz gegenüber Fehlbedienung			<input type="checkbox"/> Modifizierbarkeit	<input checked="" type="checkbox"/> Gefahrenwarnung
		<input type="checkbox"/> Ästhetik			<input checked="" type="checkbox"/> Testbarkeit	<input checked="" type="checkbox"/> Sichere Integration
		<input type="checkbox"/> Barrierefreiheit				
		<input type="checkbox"/> Benutzerunterstützend				
		<input type="checkbox"/> Selbsterklärend				

## Geeignete Testarten

### Code-Qualität und Stilprüfungen

- Code-Formatierer
- Linters
- Statische Architektur-Tests

### Funktionale Tests

- Unit-Test
- Integrations-Test
- System-Test
- Ende-zu-Ende Test
- Mutation-Test

### Kunden

- Dokumentations-Test
- A/B-Test
- Fehlerberichte
- Click-Routes Aufzeichnung
- Feedback-Funktion
- Test langsamer Datenverbindung
- Bedienungs-Test

### Entwicklungsprozess

- Code-Review (Pull-Request)
- Ticket-Review
- Feature-Review
- Testumgebungen für manuelle Tests

### Performance- und Lasttests

- Performance-Test
- Stress-Test
- Volumen-Test

### Stabilität

- Stabilitäts-Test
- Hardwareausfall-Test
- Kompatibilitäts-Test
- Backup- und Recovery-Test
- Zufallsdaten-Test
- Zufallsklick-Test

# Zusammenfassung

**... um Geld  
zu sparen**



# Unterschiedlichkeiten

🏠 Gebäudenr. 4214 / 🏢 EG 📄 ← Z.

IDs 03-42-4214 | ---

Liegenschaft auswählen  
03-42

Gebäude auswählen  
4214 Zentrales Hörsaalgebäude

Geschoss auswählen  
EG

Raum auswählen ↻

+ Neuer Raum

🔍 Suche

🗨️ 0 Türschild-Nr. HO01 🗑️

+ HINWEIS 🗨️

Nutzungsart: **Flure, Hallen**

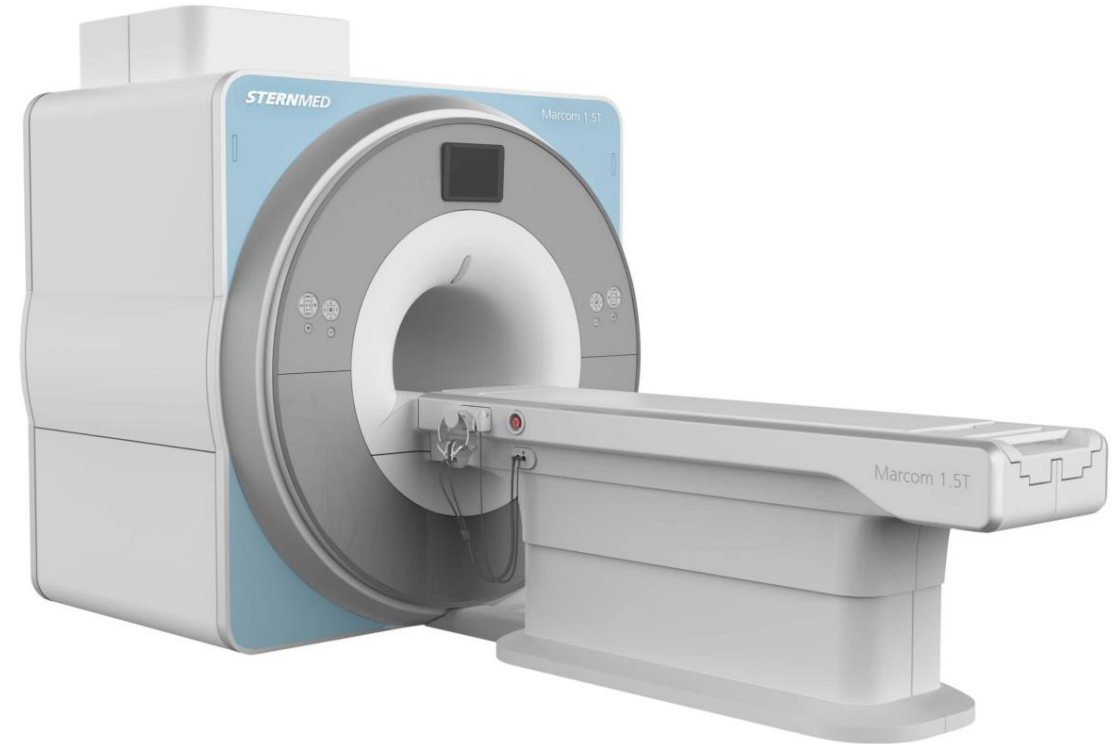
Bezeichnung: **Foyer (Steinboden)**

Boden: **Fliesen**

Aufzug bearbeiten

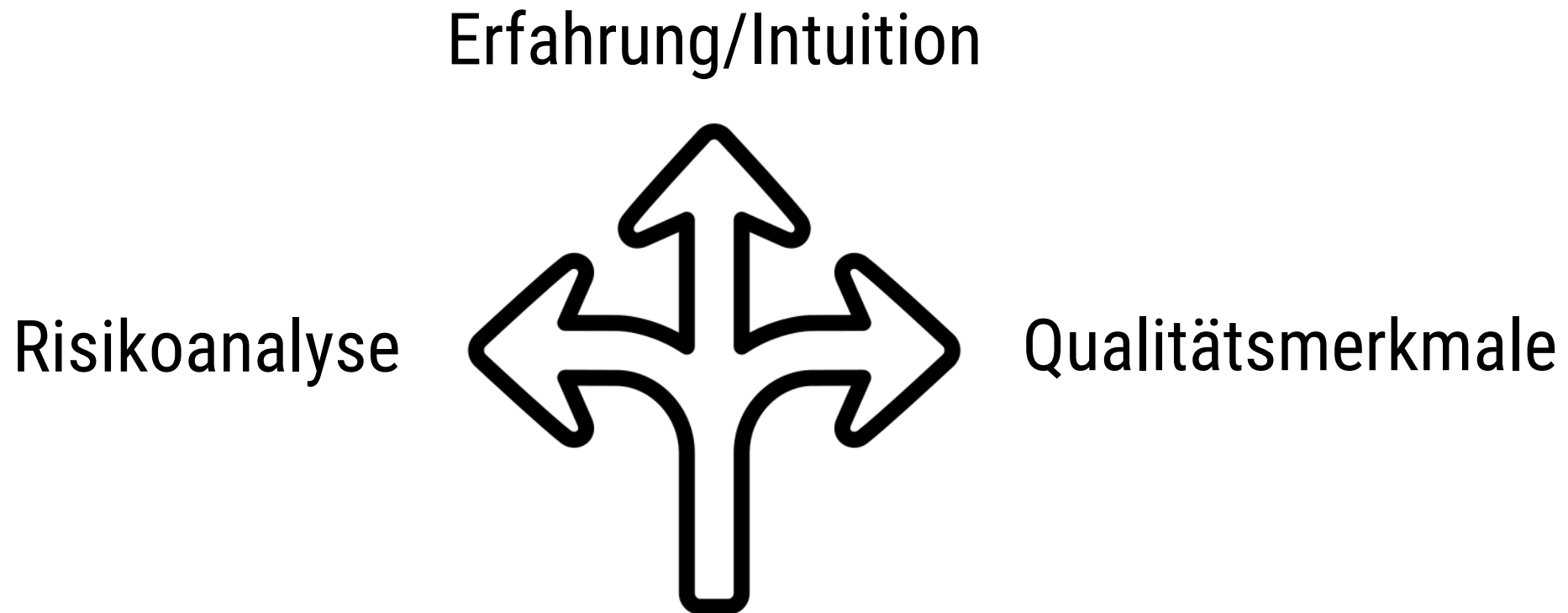
📘 Weder Türen, Fenster noch Krananlagen erfasst

vs.



<https://sternmed.de/medizingeraete/mrt-system-marcom-1-5t/>

# Drei Möglichkeiten, die richtige Art von Softwaretests zu finden



# Zusammenfassung

- Wir testen, weil wir Geld sparen wollen
- Jedes Projekt benötigt unterschiedliche Arten von Tests
- 3 Wege



Prototyp

<https://testarten.adbsw.app>

**Teste nicht einfach irgendetwas, sondern die richtigen Dinge.**

# Alexander Badewitz

- Leidenschaftlicher Software-Entwickler
- Erste Programmiererfahrung vor 20 Jahren
- Schwerpunkt: Unternehmensanwendungen mit C# .NET
- Fokus auf Webanwendungen
- Schnittstellen, Software Architektur, Testen, Clean Code, ...
- C#, Razor, Blazor, ASP.NET MVC/Core

<https://adb-solutions.de>

a.badewitz@adb-solutions.de

# Pascal Puchtlér

- Freelancer seit 2023
- Erste Programmiererfahrung vor 18 Jahren
- Schwerpunkt: Testen mit Python
- 3 wissenschaftliche Veröffentlichungen
- Software Architektur, Testen, Datenbanken, Clean Code, AI, ..
- Python, JS/TS, C#, C++, ...

<https://www.linkedin.com/in/pascal-puchtlér/>

pascal@puchtlér.software

