WAS IST EIGENTLICH EINE UNIT ?!?

Saxonia Systems
So geht Software.

Der Sprecher

Hendrik Lösch

Senior Consultant & Coach

Hendrik.Loesch@saxsys.de

@HerrLoesch

Just-About.Net

WPF-Anwendungen mit MVVM und Prism

Modulare Architekturen verstehen und umsetzen

Windows 8 Store Apps mit MVVM und Prism

XAML-Entwurfsmuster, Bootstrapping, Navigation, Messaging

Test Driven Development – Praxisworkshop

Business-Applikationen testgetrieben entwickeln

Inversion of Control und Dependency Injection

Prinzipien der modernen Software-Architektur ...

Test Driven Development mit C#

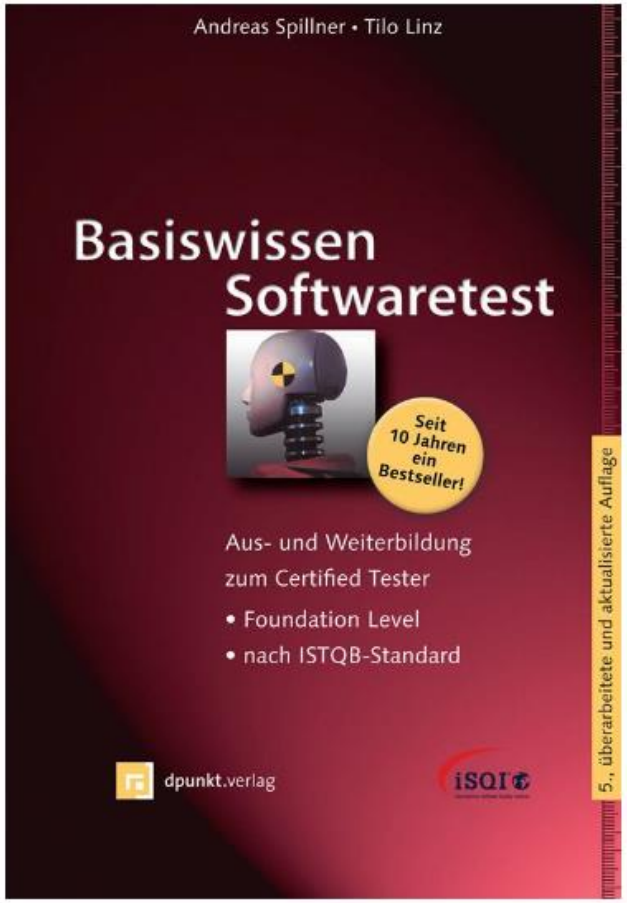
Grundlagen, Frameworks, best Practices

Automatisiertes Testen mit Visual Studio 2012

Grundlagen, Testarten und Strategien

Saxonia Systems

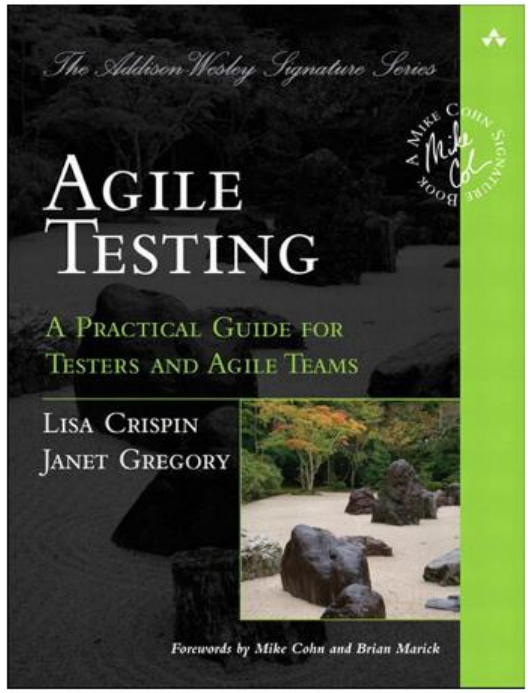
So geht Software.



In der ersten Teststufe, **dem Komponententest**, **werden die** in der nach dem V-Modell unmittelbar vorangehenden Programmierphase erstellten **Softwarebausteine** erstmalig **einem systematischen Test unterzogen**.

Abhängig davon, welche Programmiersprache die Entwickler einsetzen, werden diese kleinsten Softwareeinheiten unterschiedlich bezeichnet, zum Beispiel als **Module**, **Units** oder **Klassen** (im Fall objektorientierter Programmierung). **Die entsprechenden Tests werden Modultest, Unit Test bzw. Klassentest genannt.**

Von der verwendeten Programmiersprache abstrahiert, wird von **Komponente** oder **Softwarebaustein** gesprochen. Der Test eines solchen einzelnen Softwarebausteins wird als Komponententest bezeichnet.

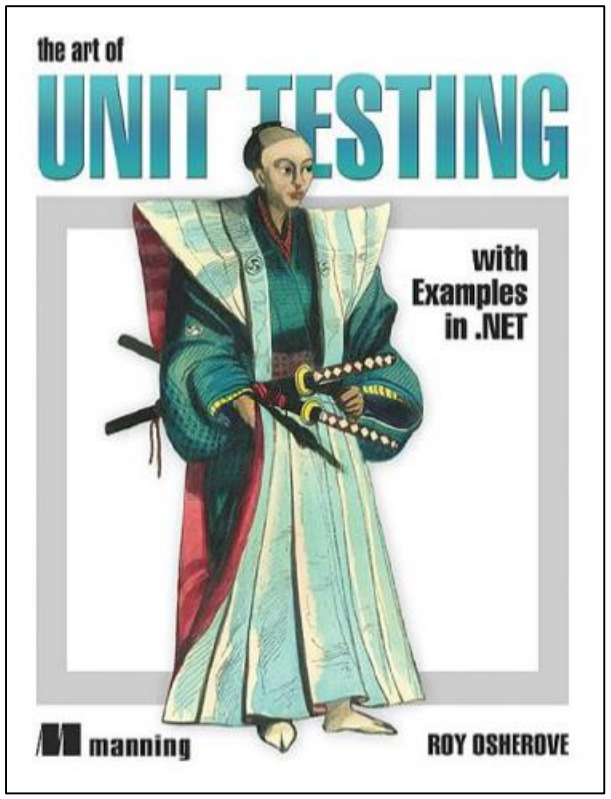


Unit Test

A unit test verifies the behavior of a small part of the overall system. It may be as small as **a single object or method** that is a consequence of one or more design decisions.

Component test

A component test verifies a component's behavior. Component tests help with component design by testing interactions between objects.
A component is a larger part of the overall system that may be separately deployable. For example, on the Windows platform, dynamic linked libraries (DLLs) are used as components, Java Archives (JAR files) are components on the Java platform, and a service-oliented architecture (SOA) uses Web Services as components.



Unit Test

A unit test is an automated Piece of code that **invokes the method or class being tested and then checks some assumptions about the logical behavior of that method or class**. A unit test is almost always writen using a unit-testing framework. It can be written easily and runs quickly. It‘s fully automated, trustworthy, readable and maintainable.

Integration Test

Integration testing means testing two or more dependent sofware modules as a group.

Komponenten?!?

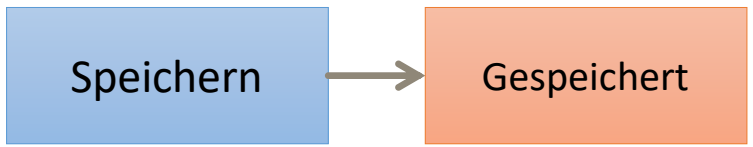
Module?!?

Softwarebausteine?!?

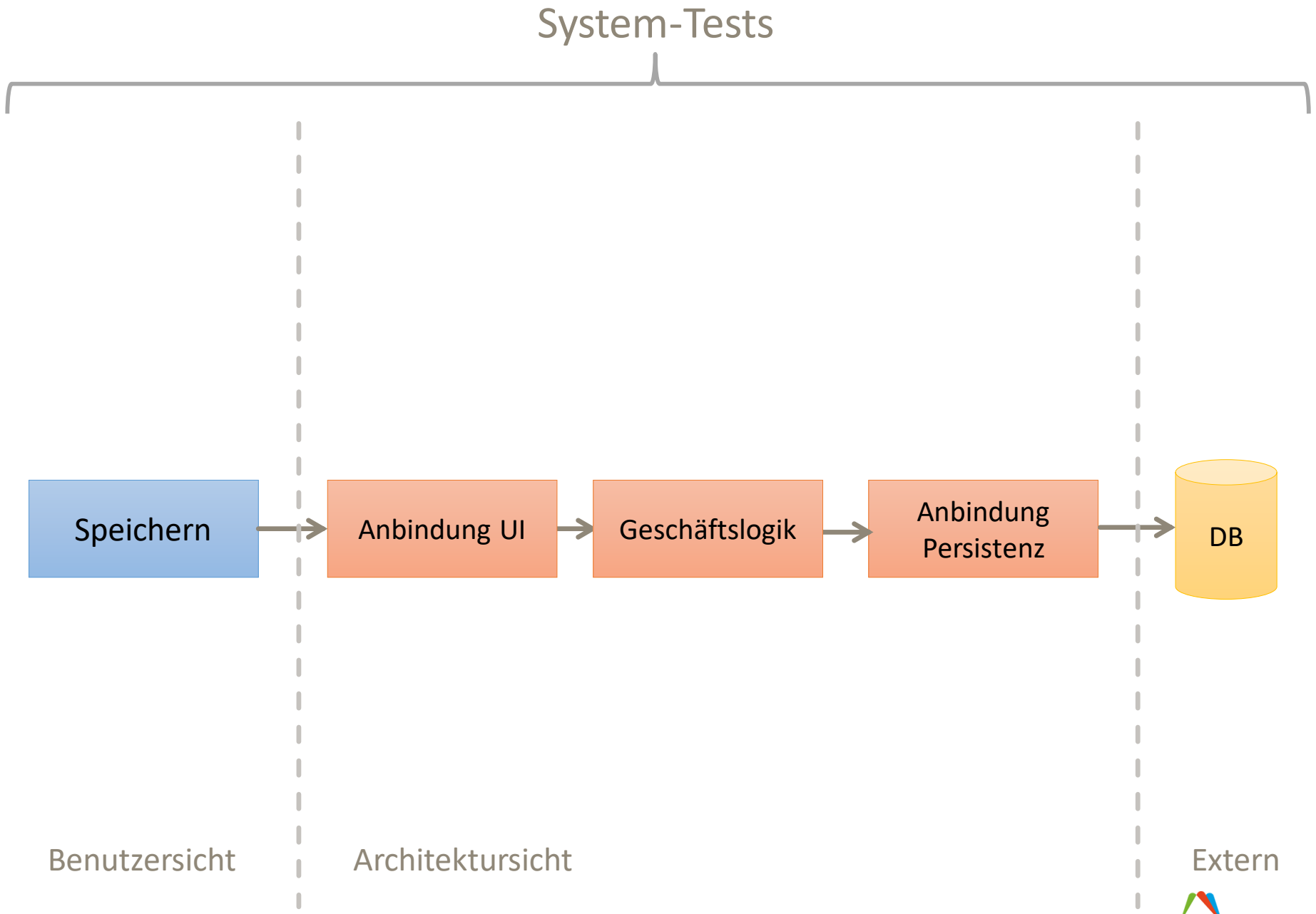
Units?!?

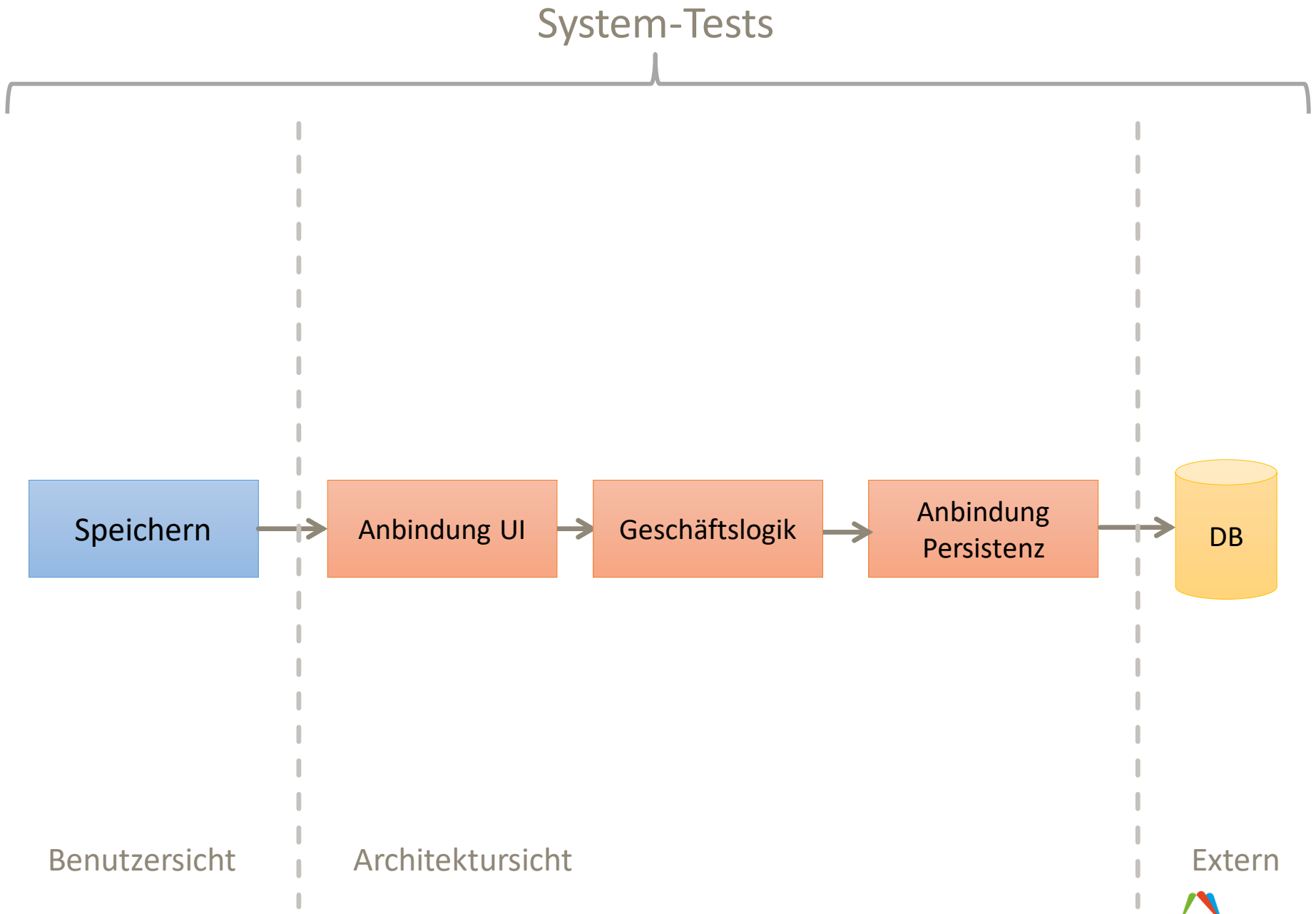
Klassen?!?

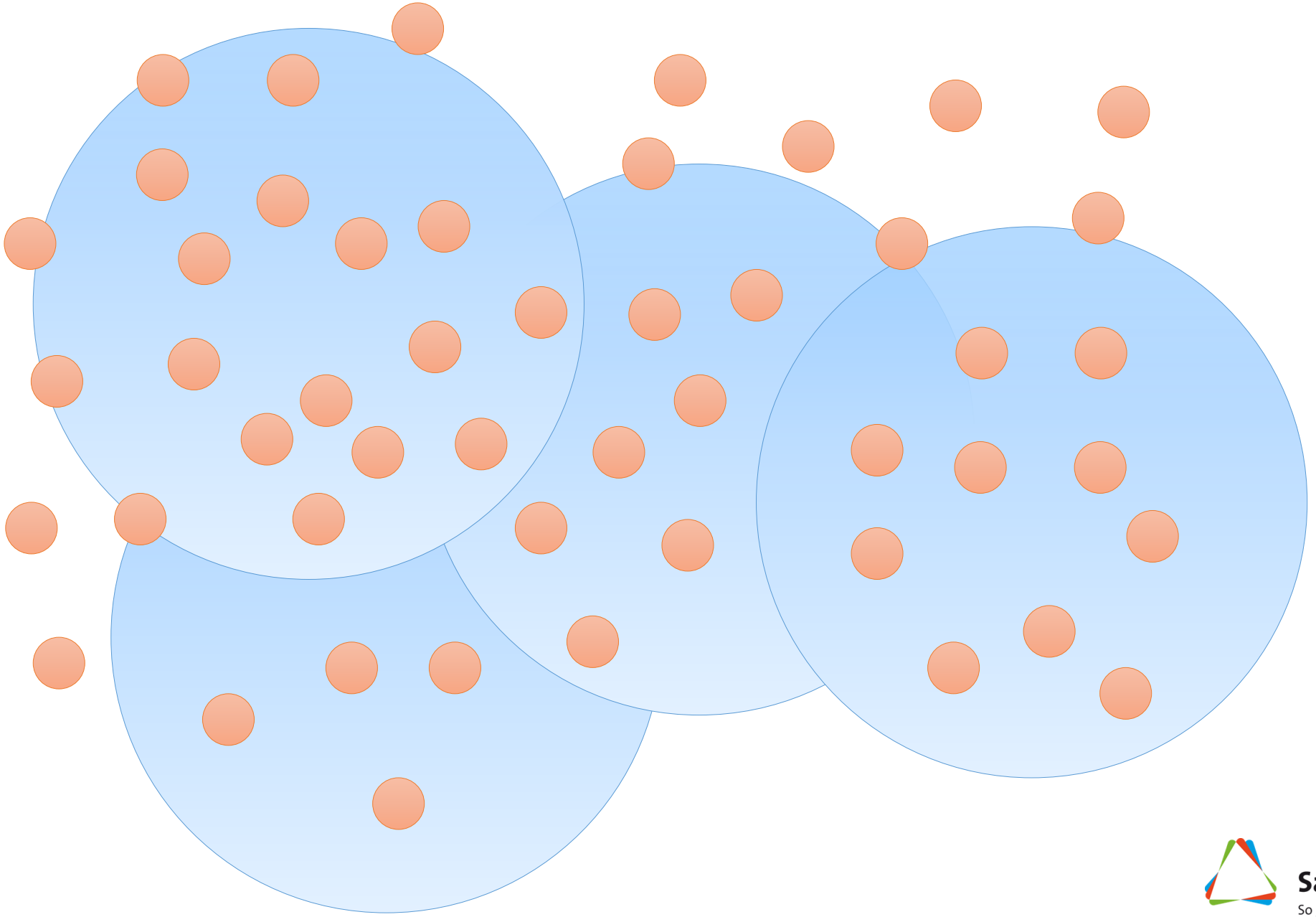
System?!?

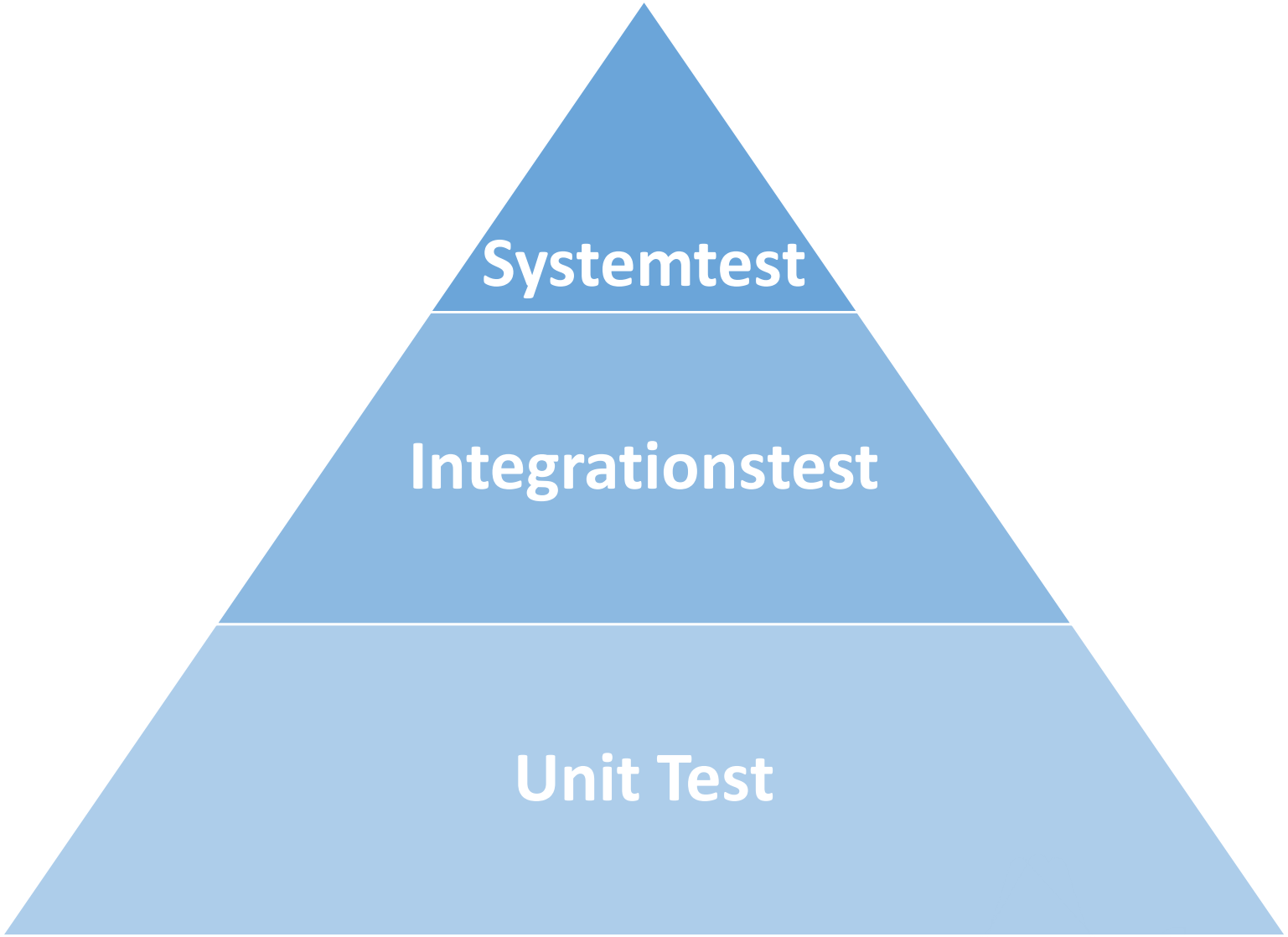


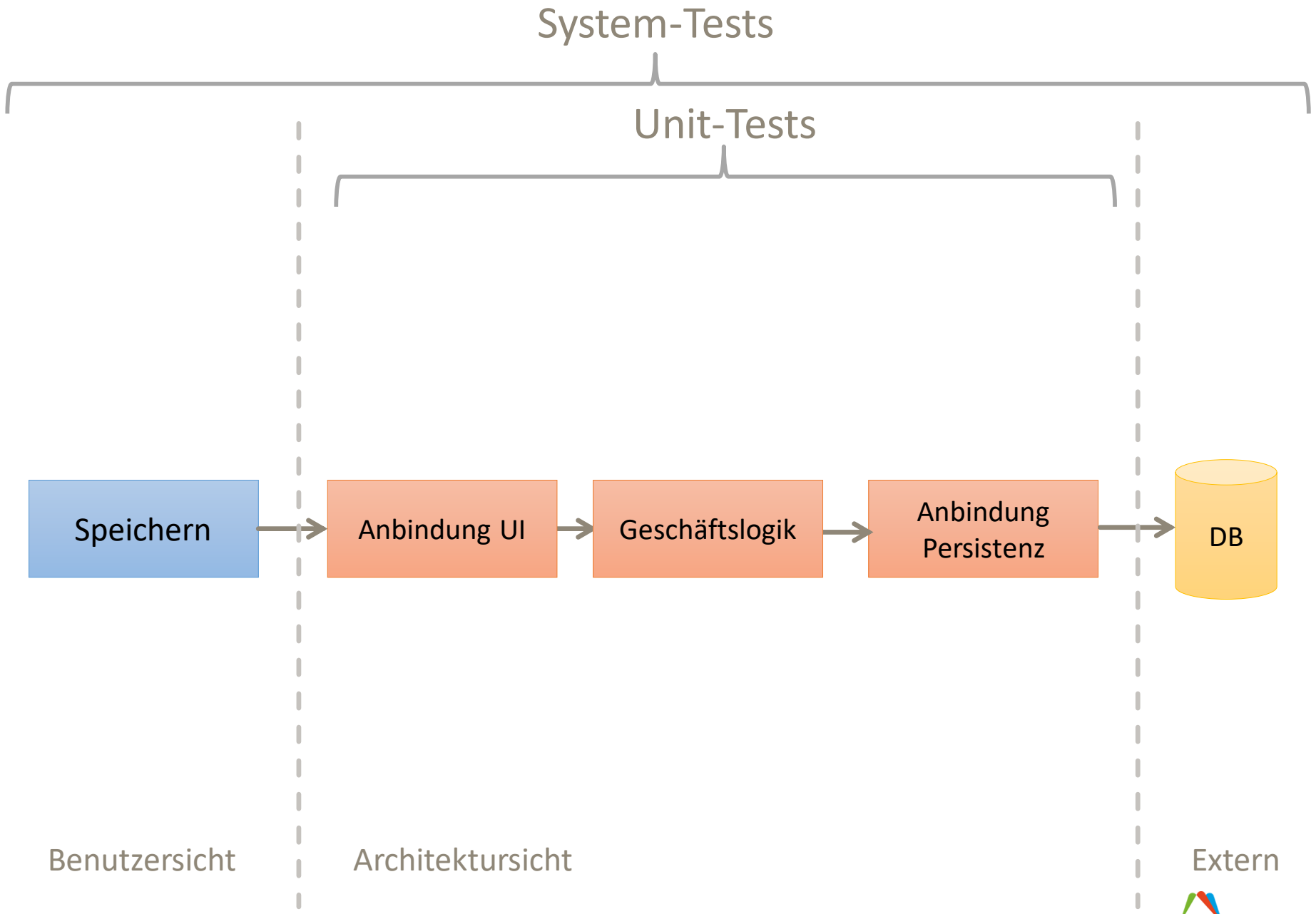
Benutzersicht



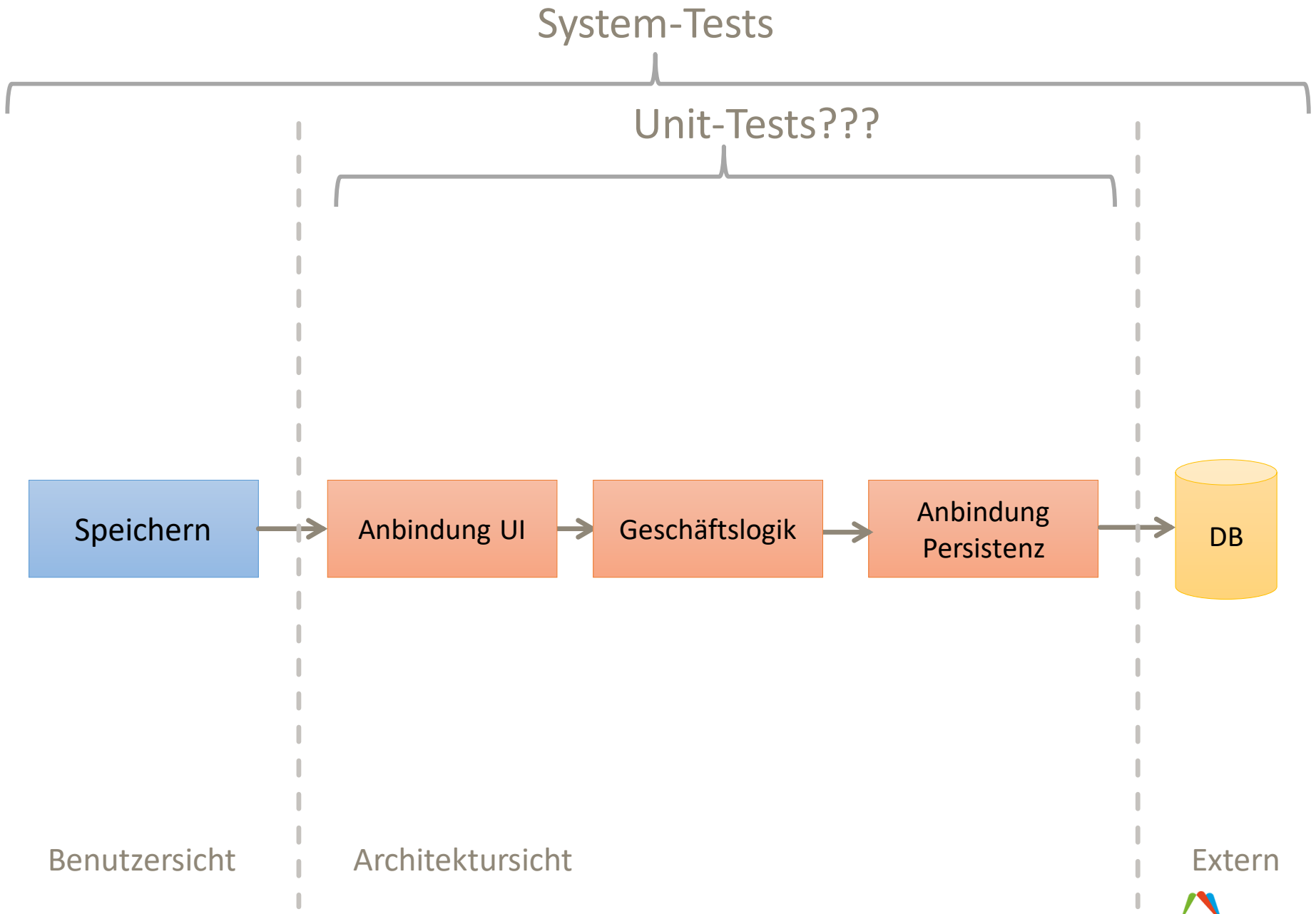




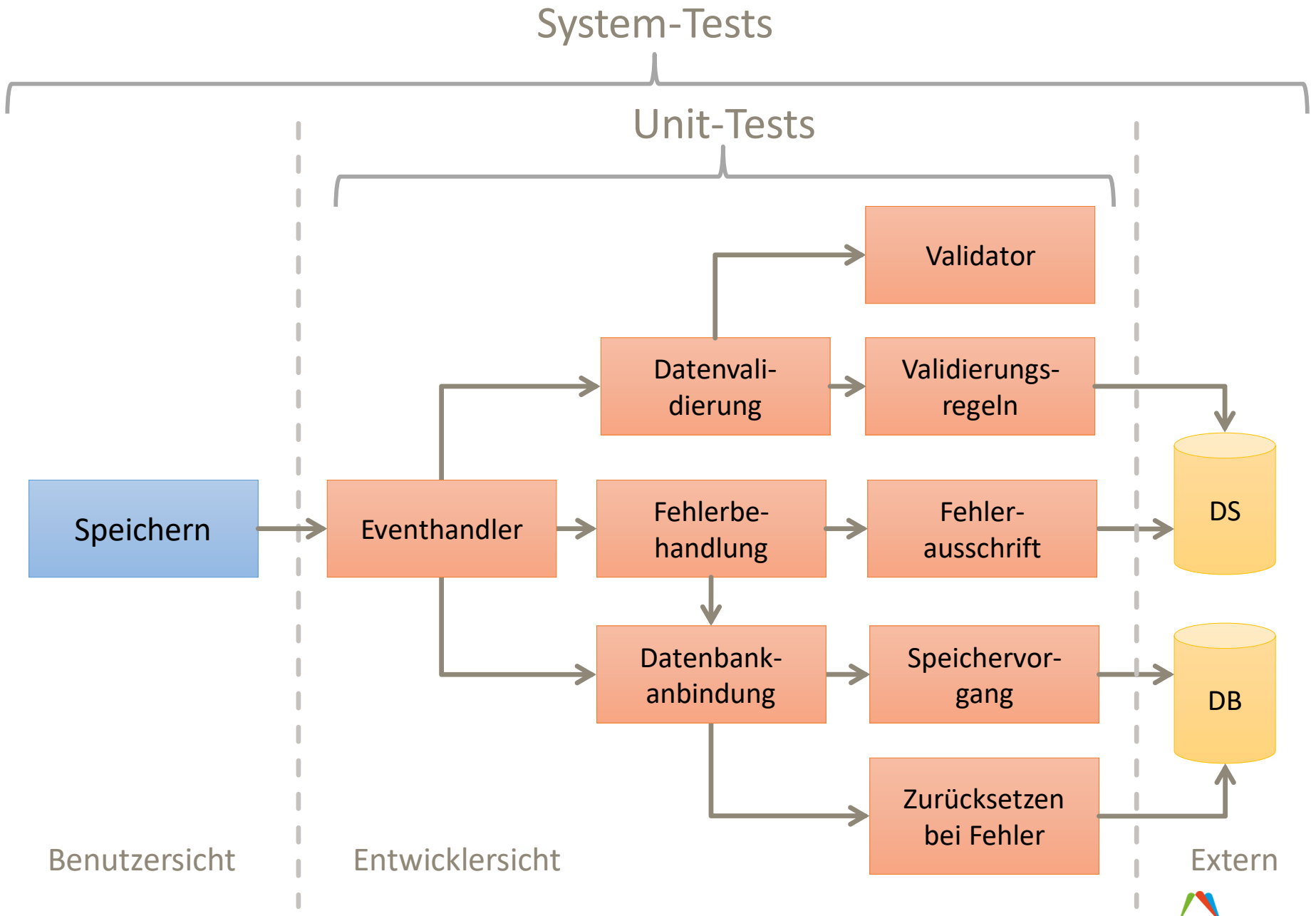

Systemtest
Integrationstest
Unit Test



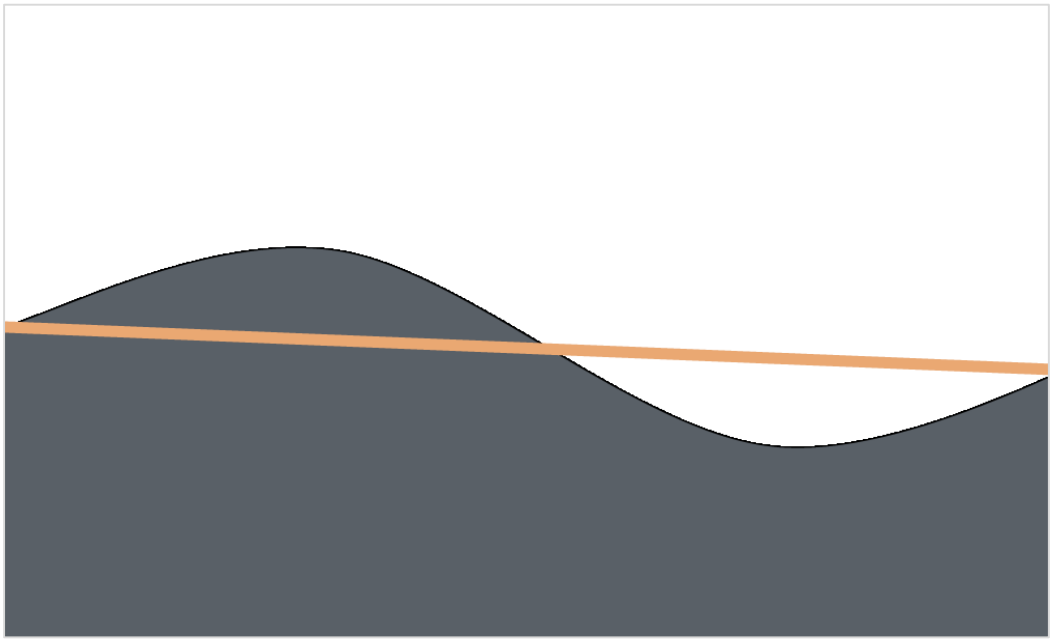
Saxonia Systems
So geht Software.



Saxonia Systems
So geht Software.



Saxonia Systems
So geht Software.



Gerade Messabschnitte von
200 km Länge.
Gesamtlänge ungefähr
2350 km.

WIKIPEDIA

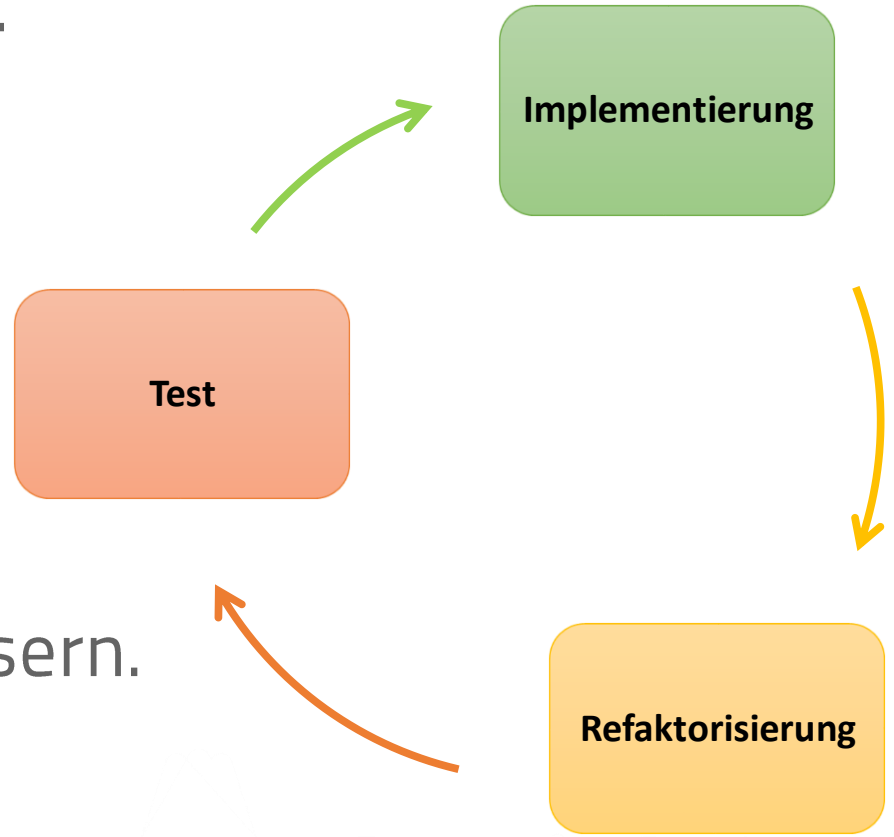
Saxonia Systems
So geht Software.

Schreibe nur Code, der verlangt wird.

Entwickle schrittweise Deinen Code.

Wähle möglichst kleine Schritte.

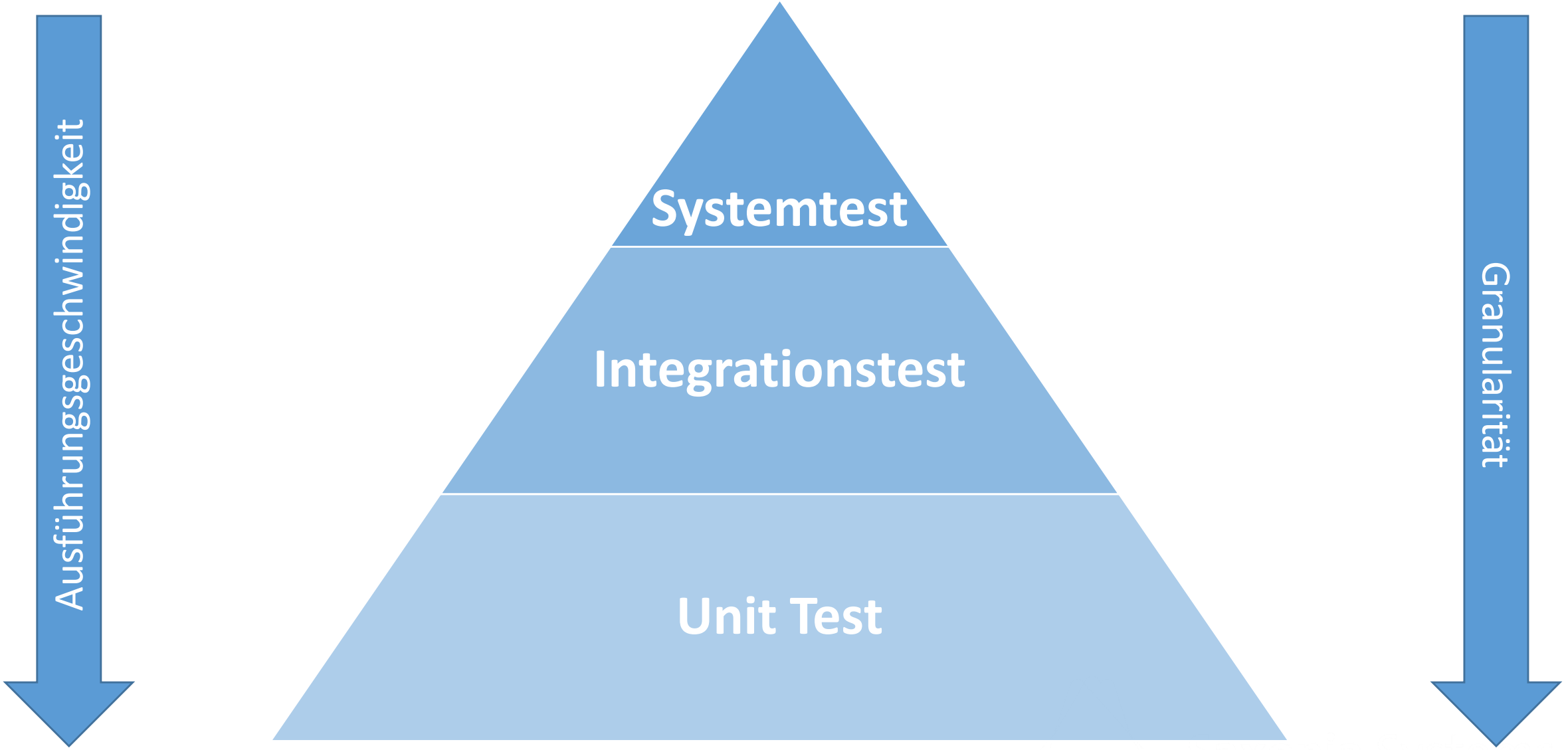
Jeder Schritt muss den Code verbessern.

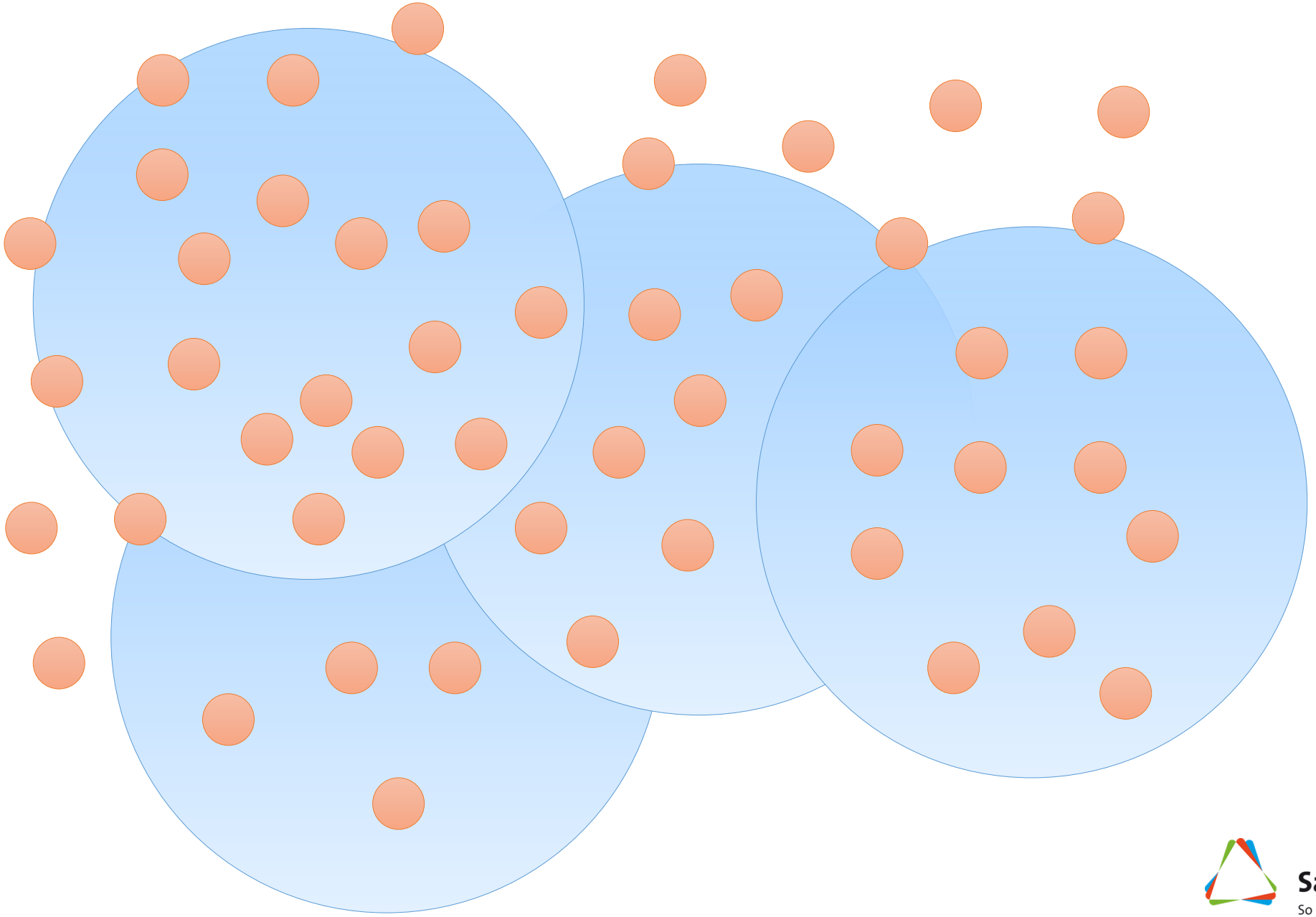


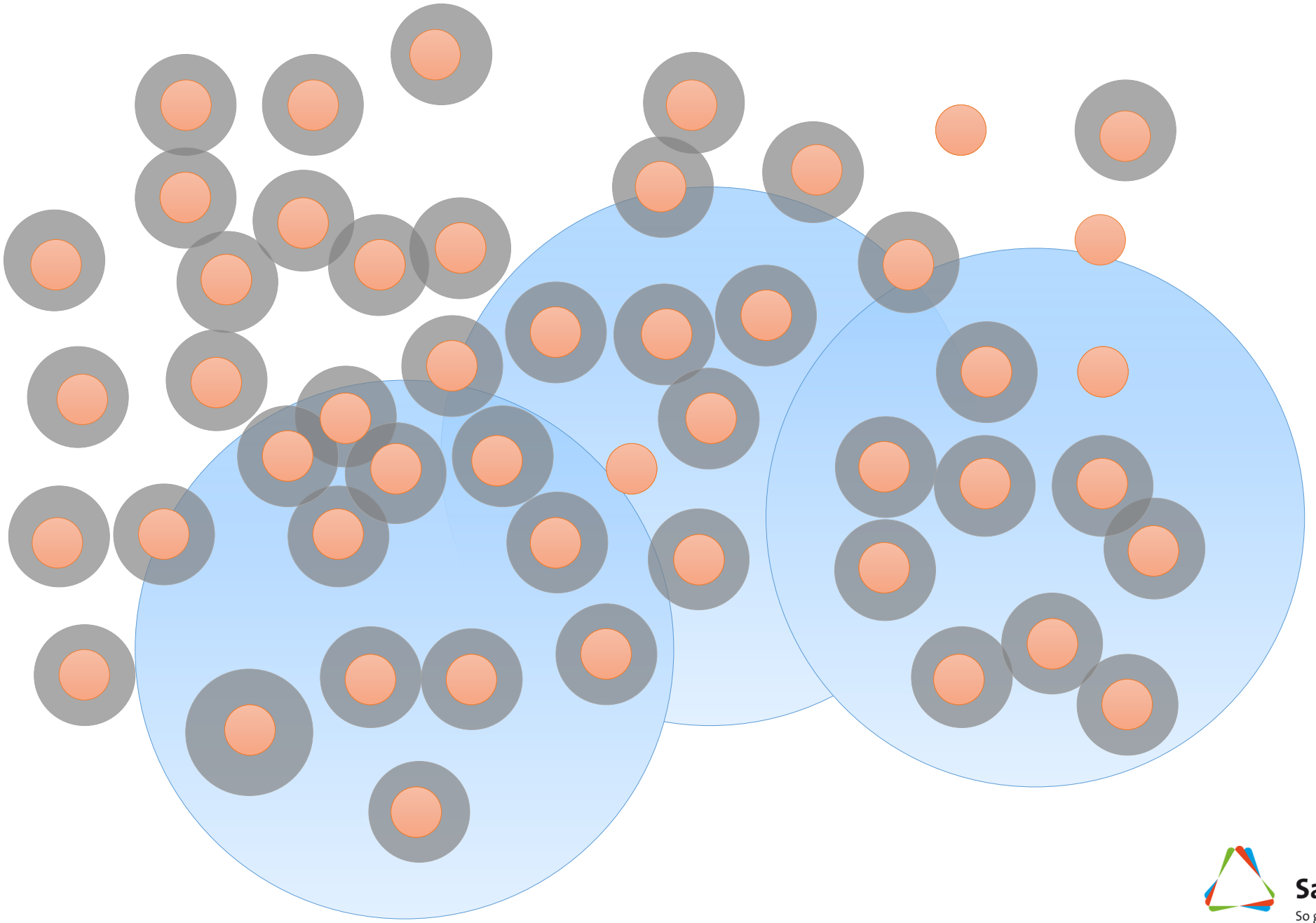
DEMO

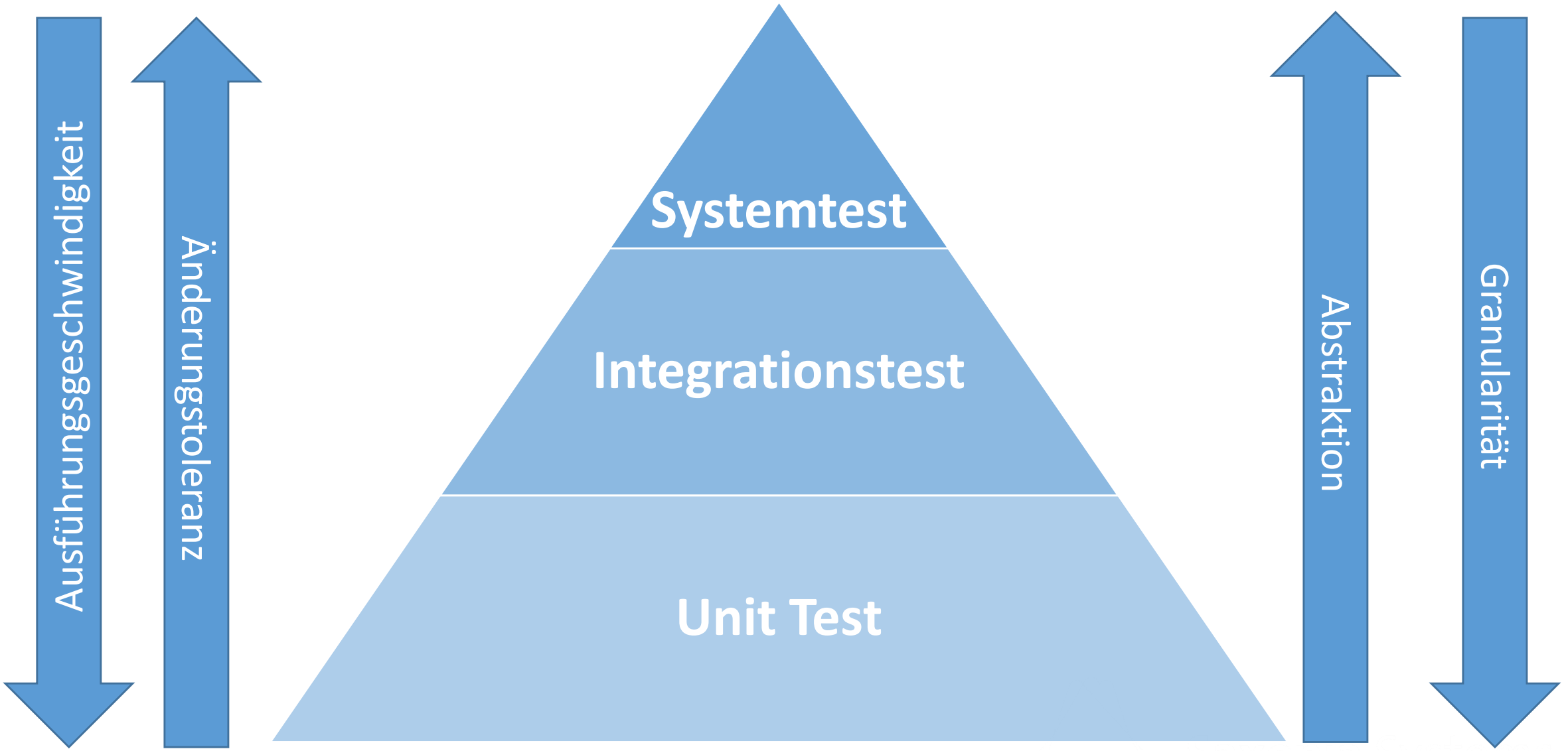
„Klassen Tests“

Testen nur eine Klasse unter Berücksichtigung deren direkter Einflussfaktoren.









Saxonia Systems
So geht Software.

DEMO

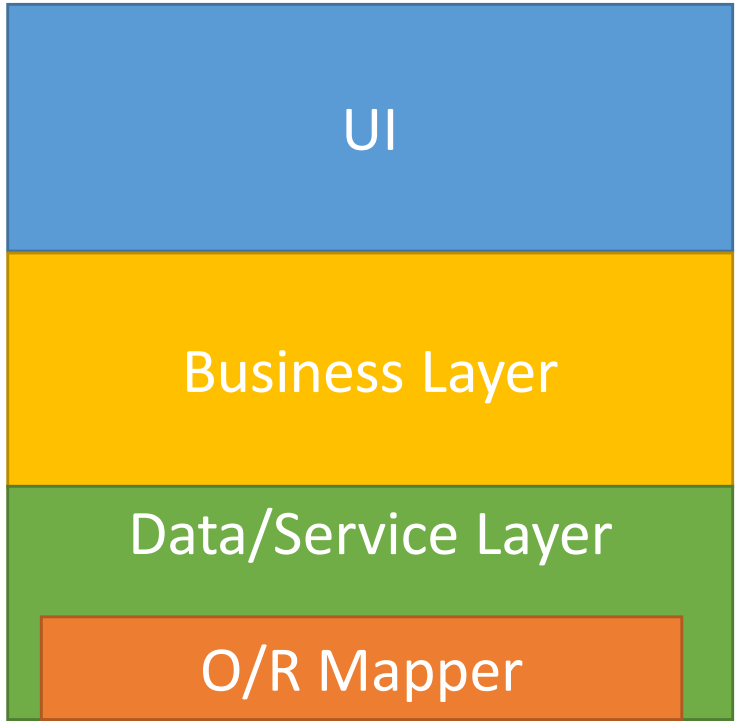
„Klassen Tests“

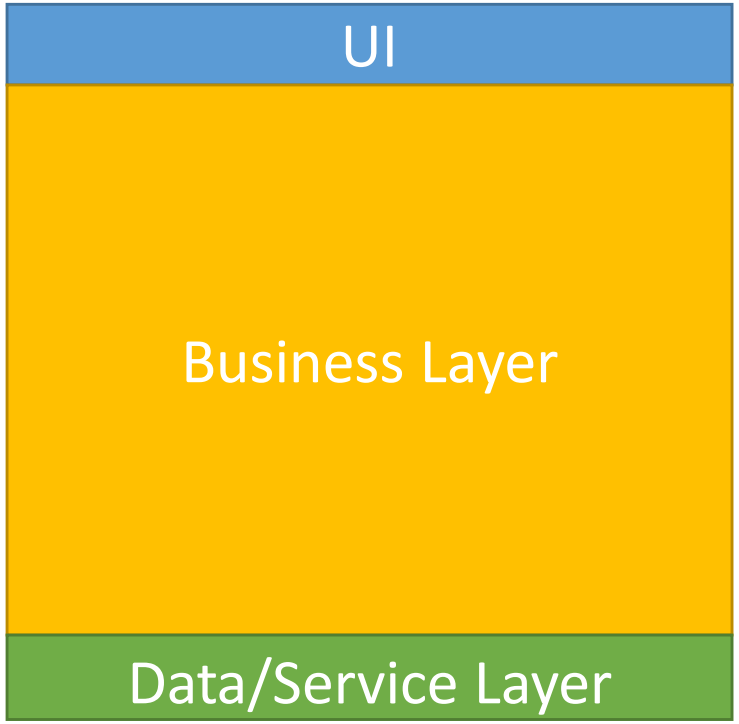
Vorteil

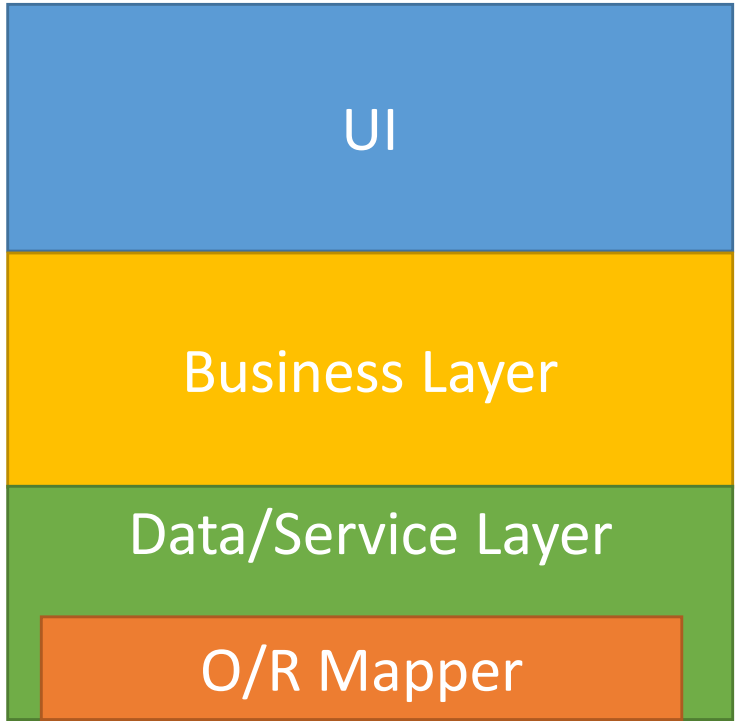
- Sehr genaue Fehlerbeschreibung
- Schnell ausführbar

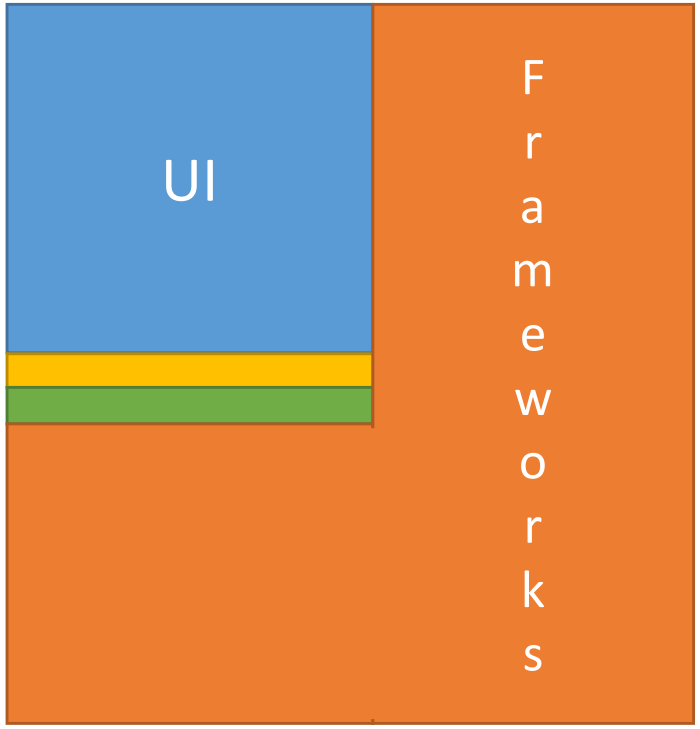
Nachteile

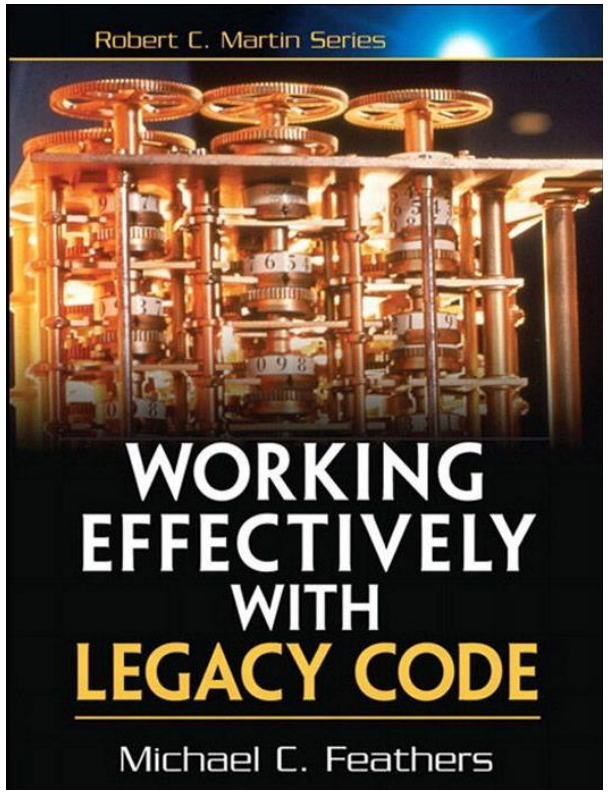
- hoher Gebrauch von Testdoubeln bei interaktionsbasierten Tests
- Durch Refactoring vergleichsweise häufige Anpassungen notwendig
- Keine Aussage zur Qualität der tatsächlichen Interaktion






UI
Business Layer
Data/Service Layer
O/R Mapper


UI
Frameworks



Unit tests run fast. If they don't run fast, they aren't unit tests. Other kinds of tests offen masquerade as unit tests. A test is **not** a unit test if:

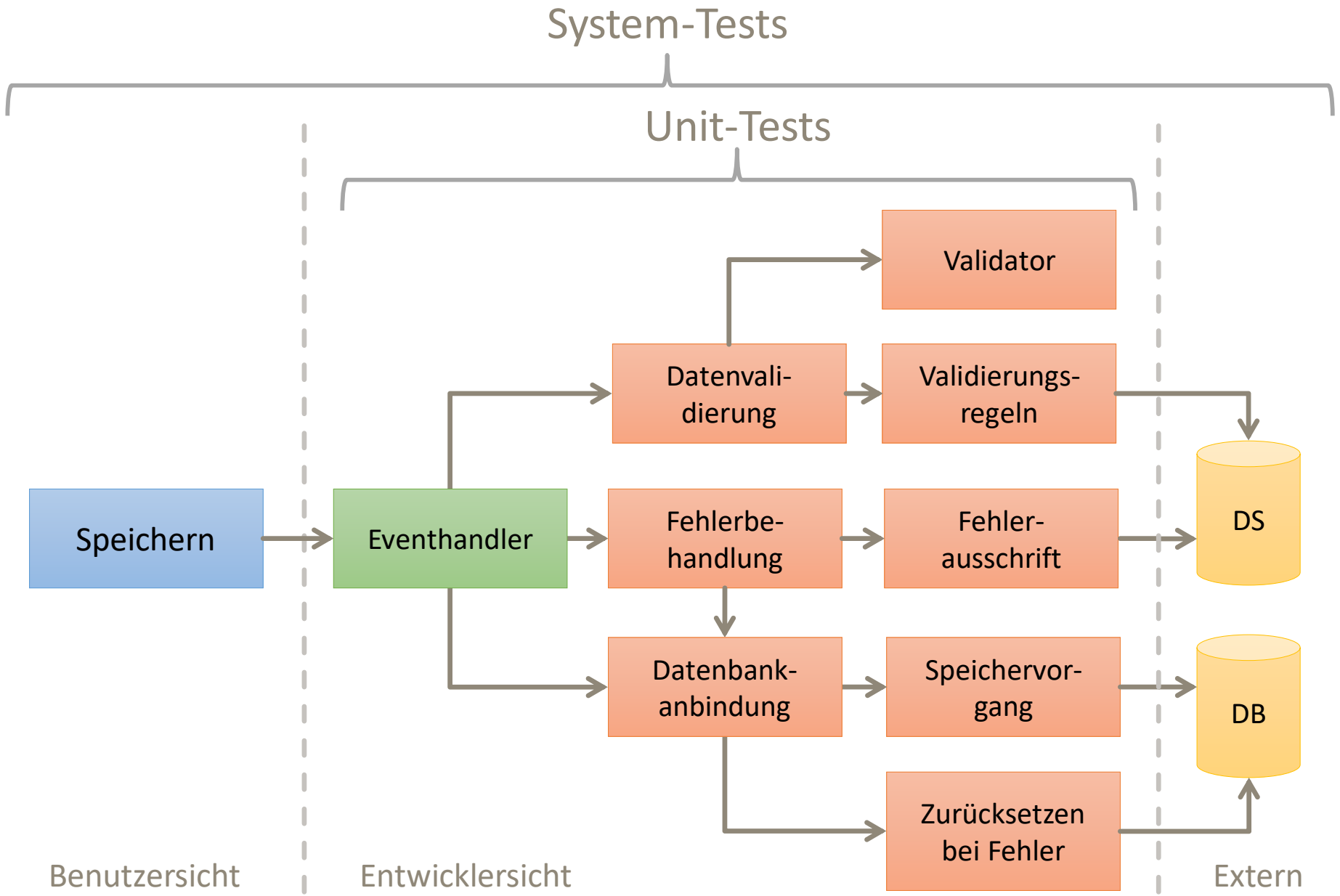
1. It talks to a database.
2. It communicates across a network.
3. It touches the file system.
4. You have to do special things to your environment (such as editing configuration files) to run it.

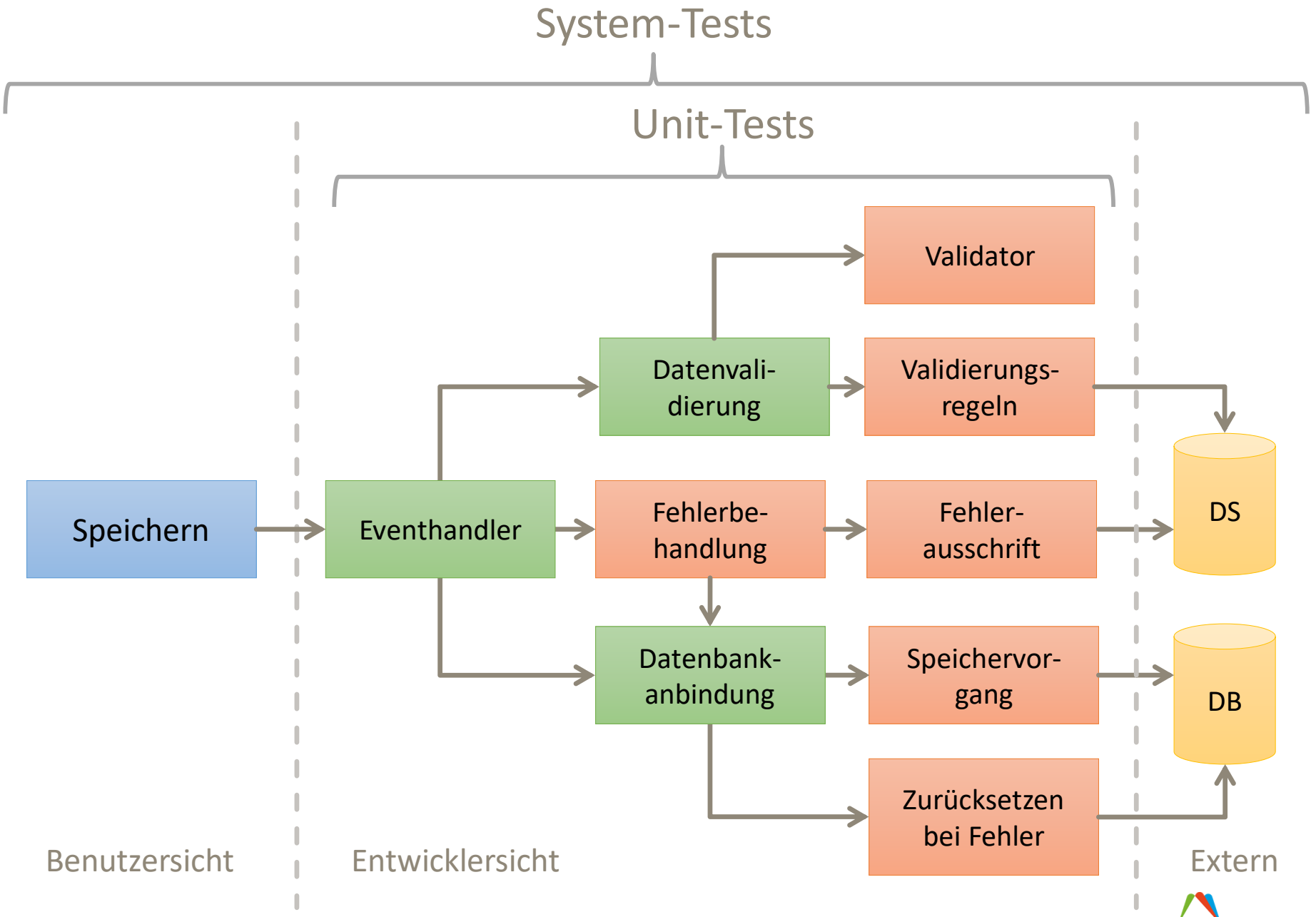
Tests that do these things aren't bad. Often they are worth writing, and you generally will write them in unit test harnesses. However, it is important to be able to separate them from true unit tests so that you can keep a set of tests that you can runfast whenever you make changes.

„Komponenten Tests“

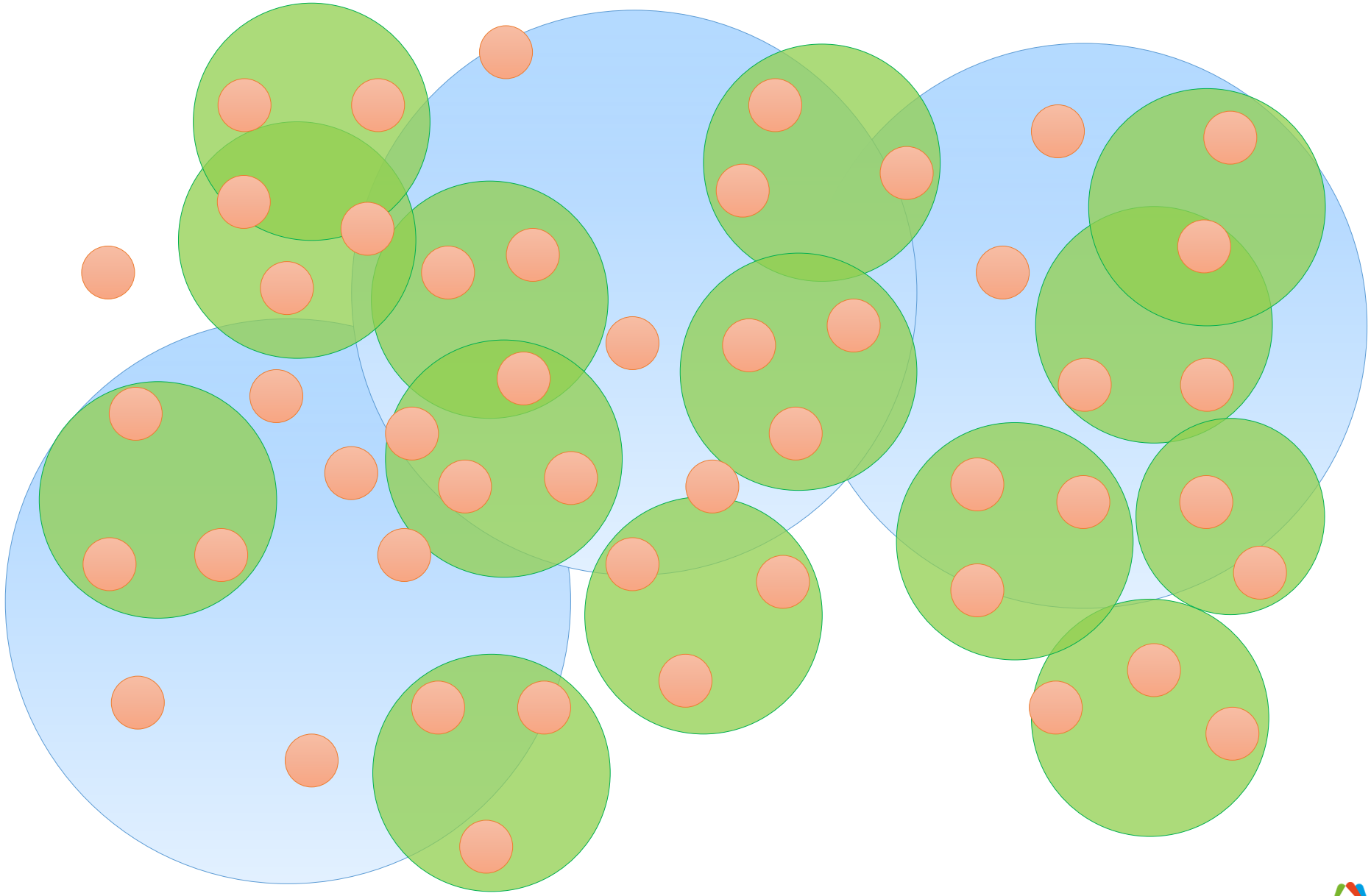
Testen einen Verbund von Klassen und Methoden, basierend auf ihrem einzigen öffentlichen Einstiegspunkt.

DEMO





Saxonia Systems
So geht Software.



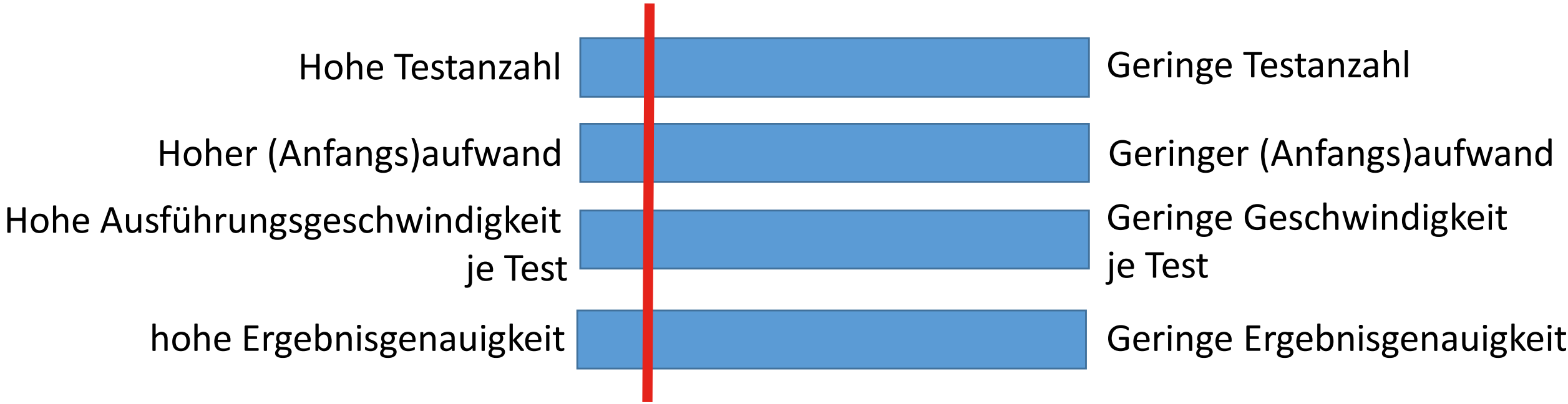
„Komponenten Tests“

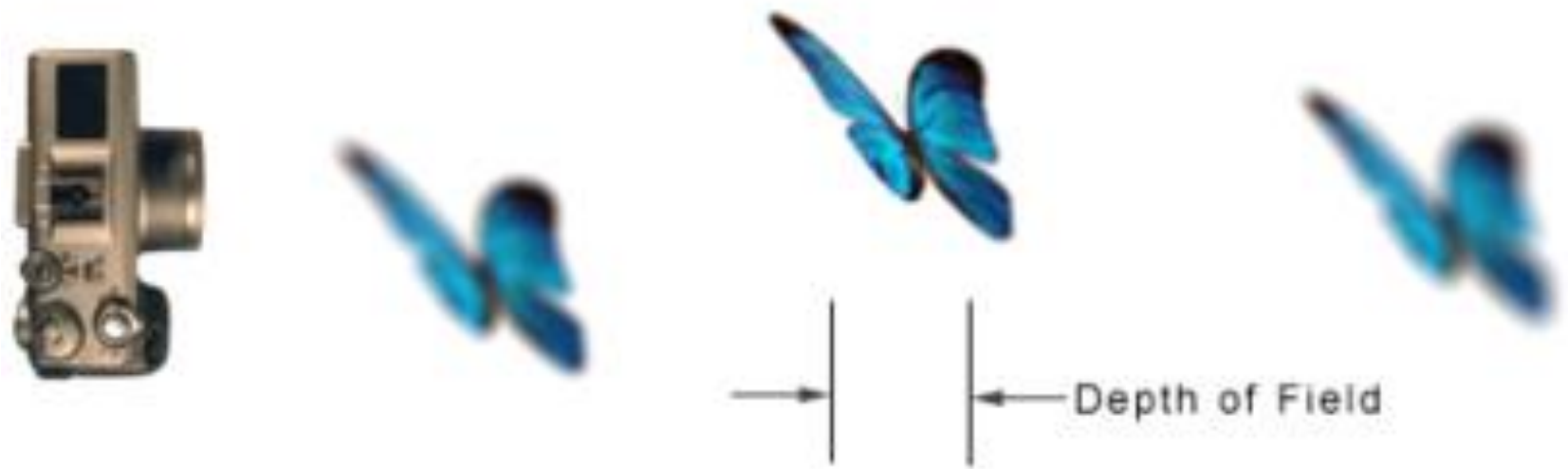
Vorteile

- Geringe Anzahl von Testfällen im Verhältnis zu Klassentests
- Robuster gegenüber Änderungen am SUT
- Fehler werden gefunden, die sich aus der Interaktion von Bestandteilen ergeben

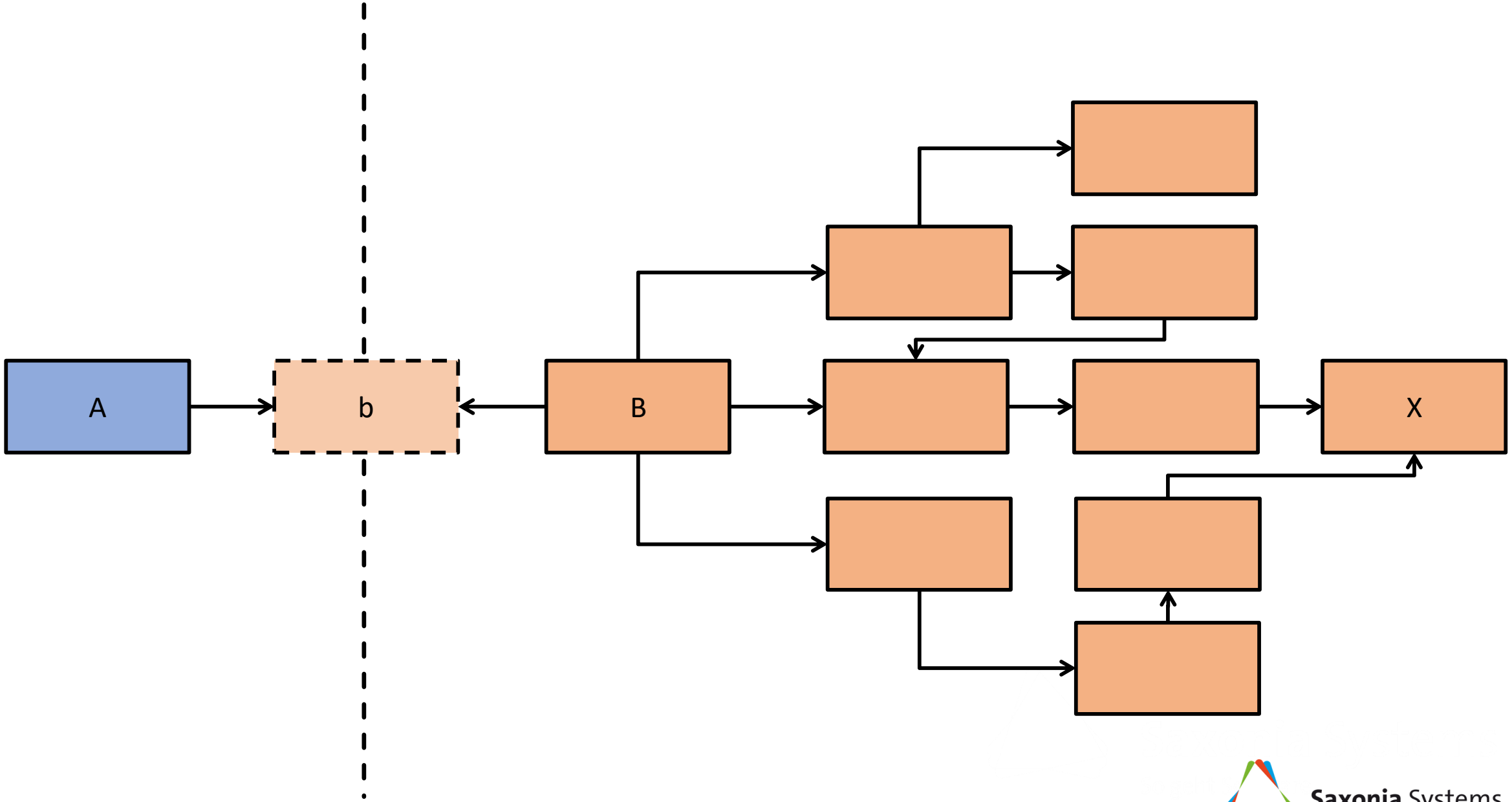
Nachteile

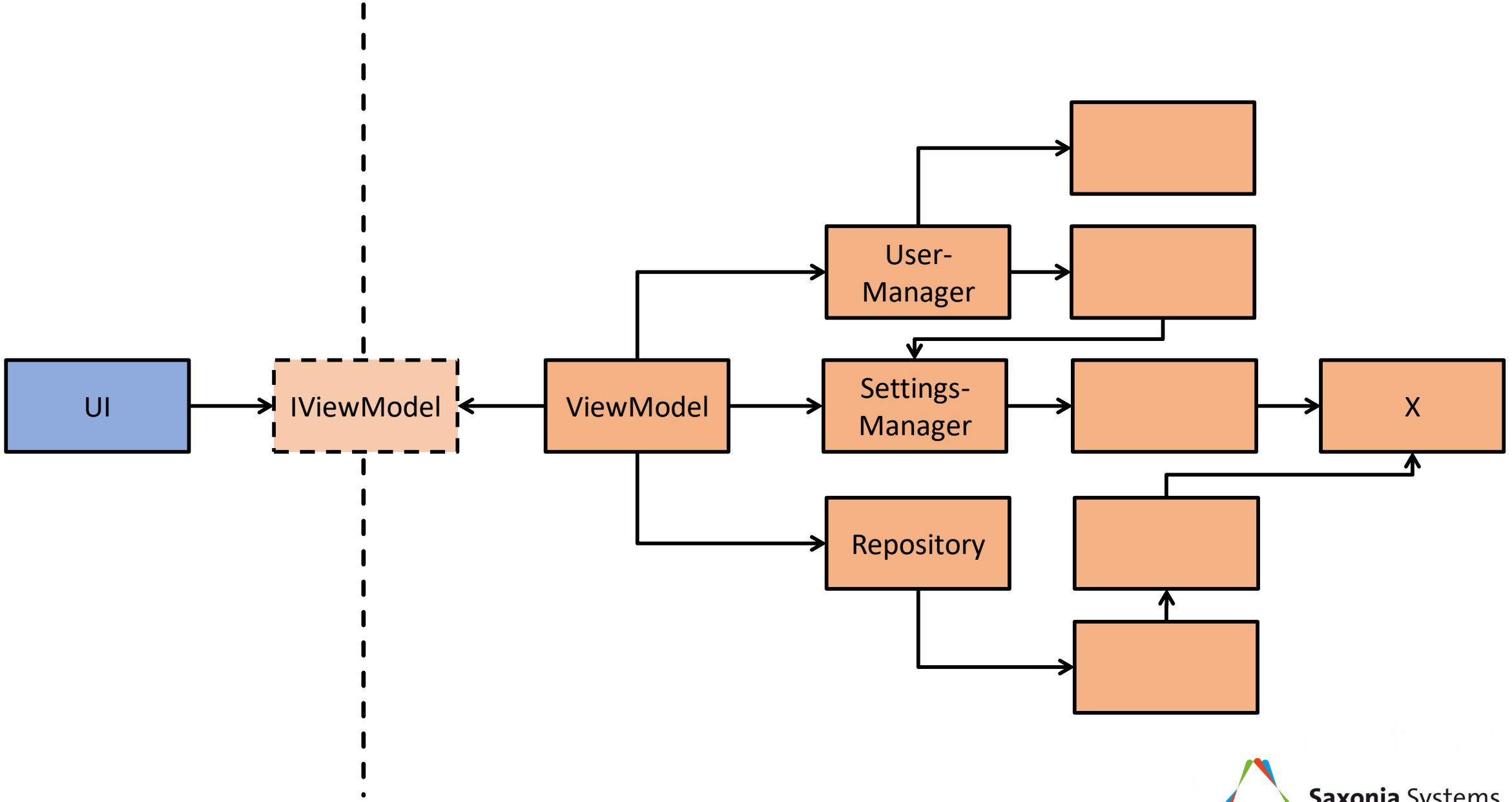
- Längere Ausführungszeiten
- Schlechtere Aussagekraft
- Fehleranfällig durch nicht betrachtete Nebeneffekte
- Teils komplexe Vorbedingungen



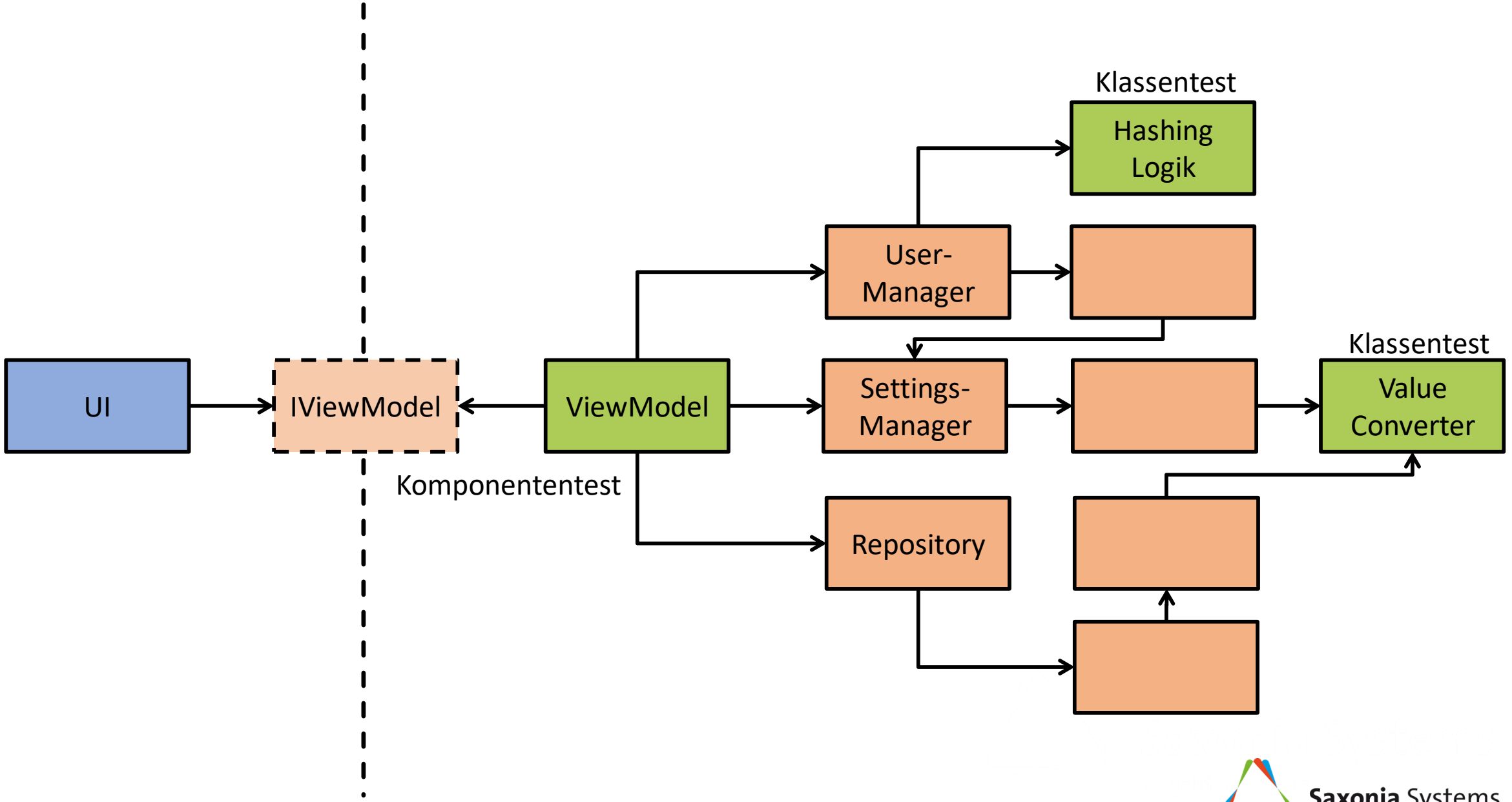


Quelle: http://fabiopereira.me/blog/2012/03/18/introducing-depth-of-test-dot/

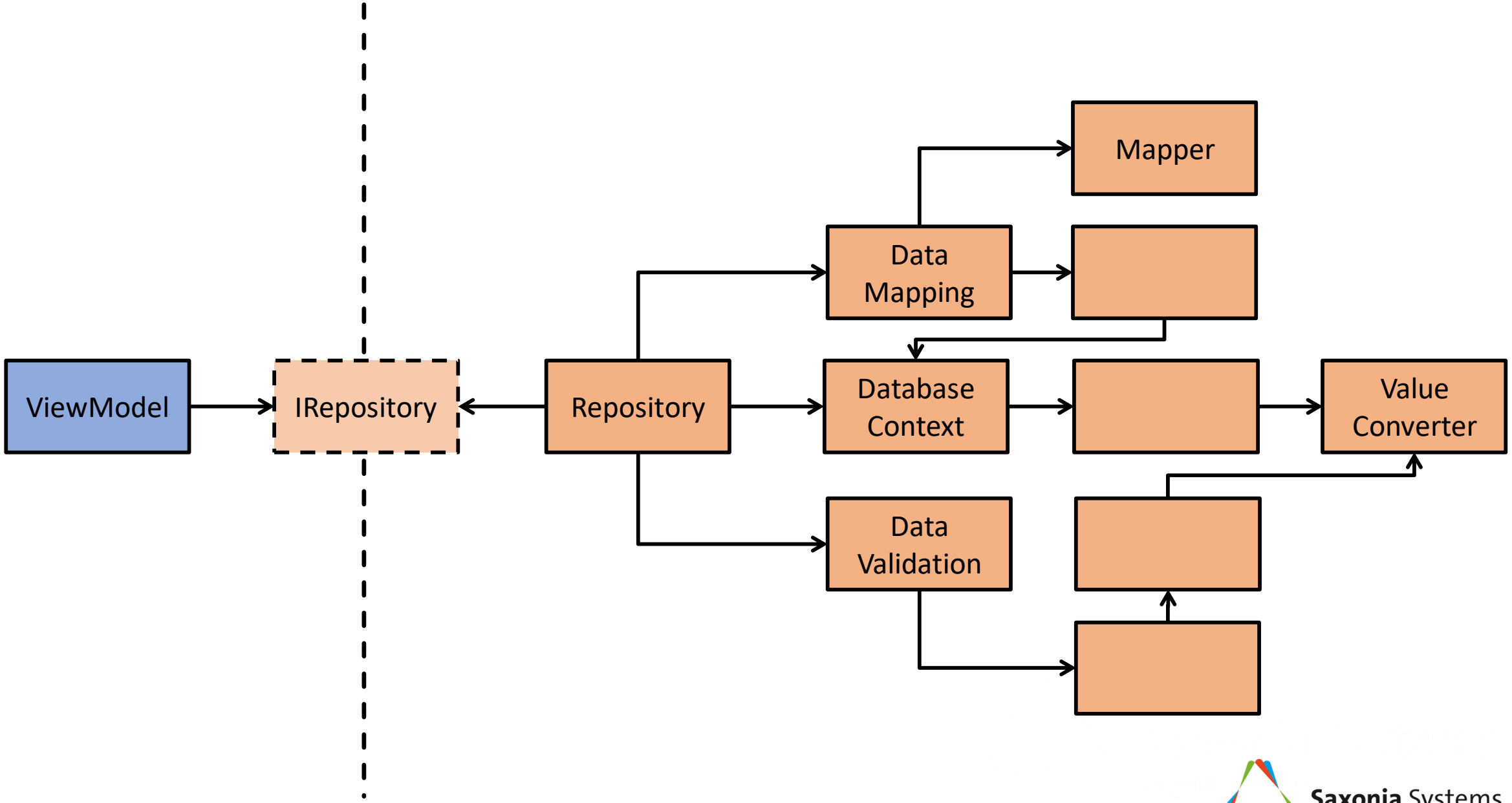




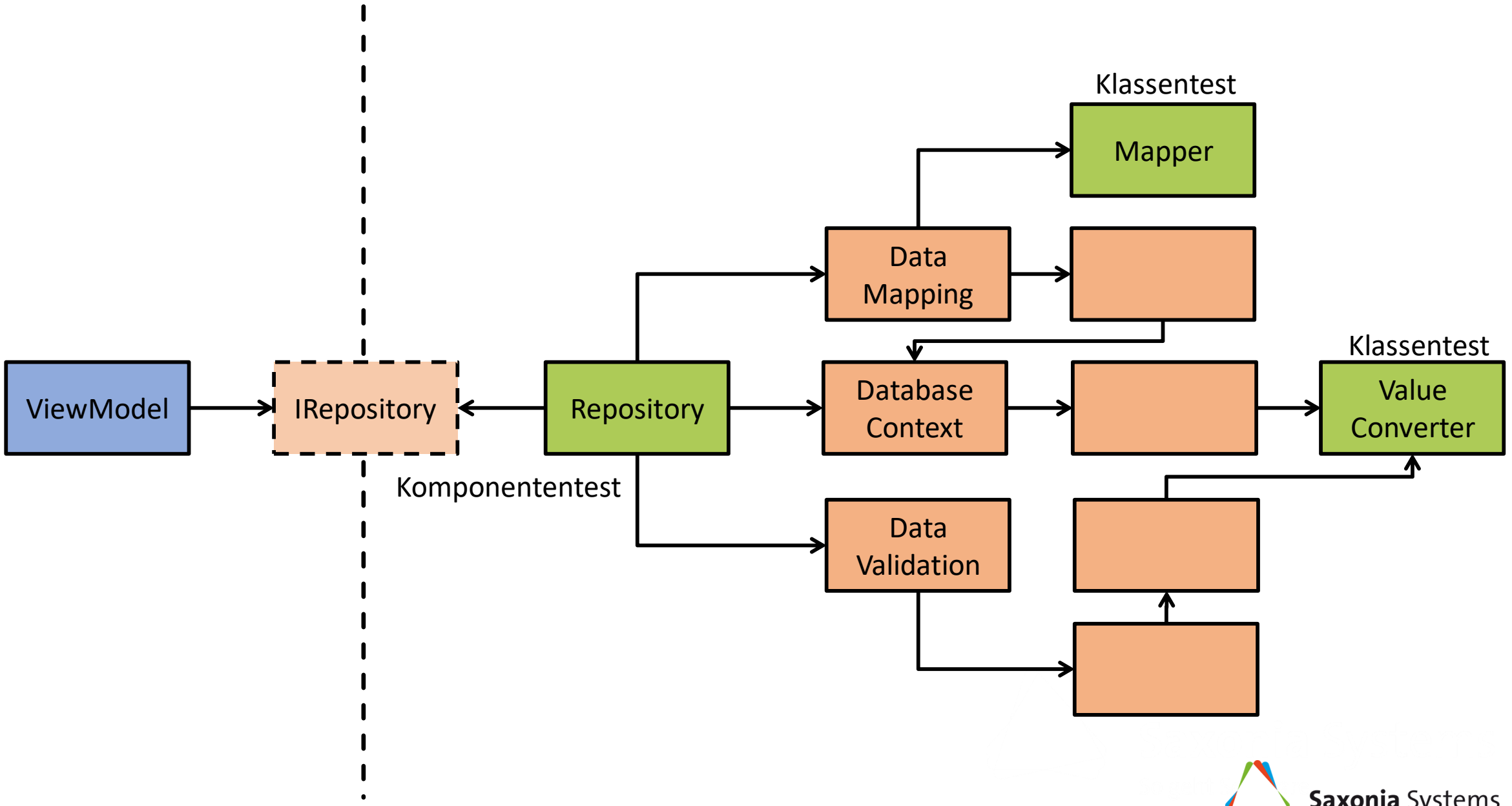
Saxonia Systems
So geht Software.



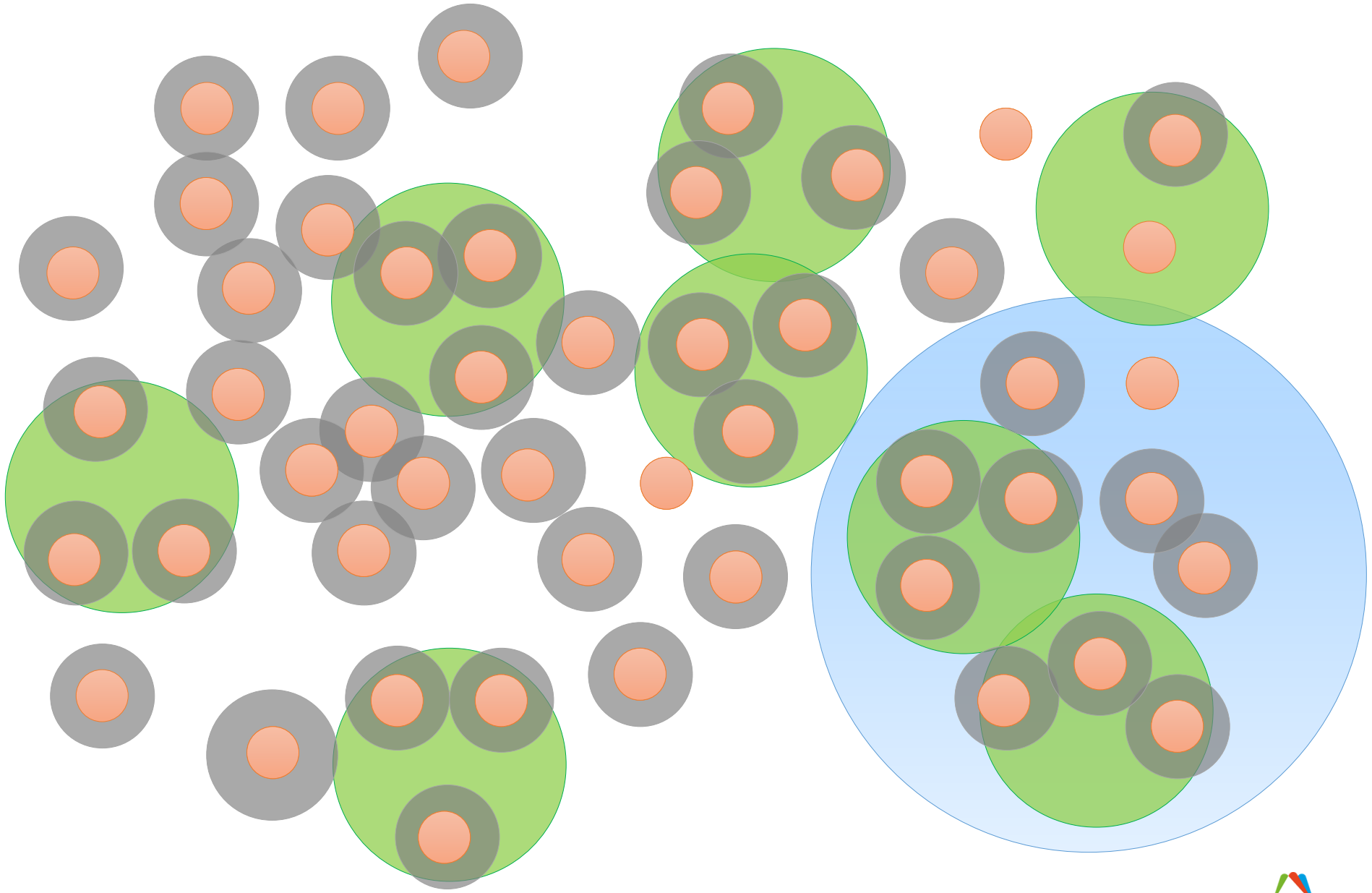
Saxonia Systems
So geht Software.

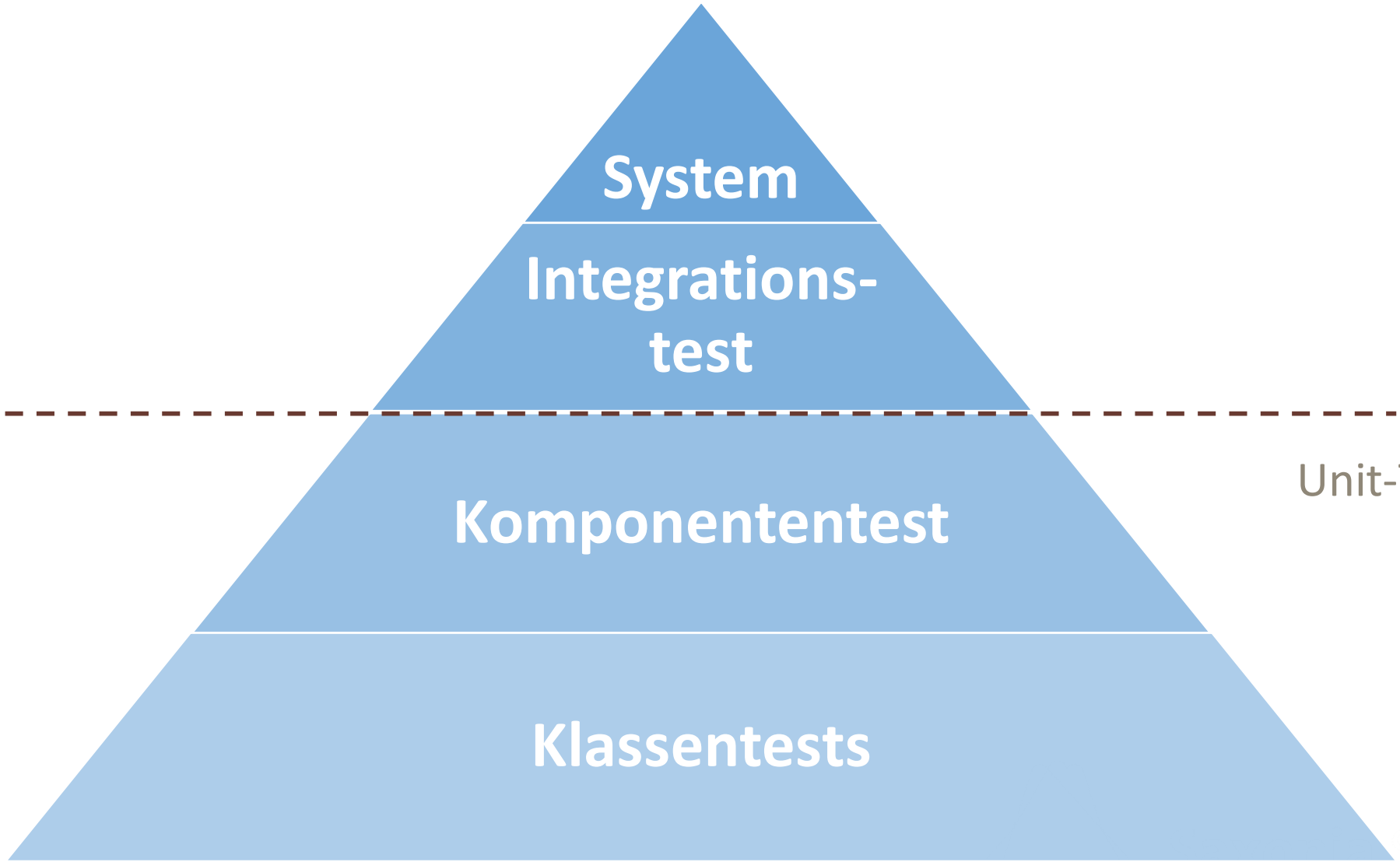


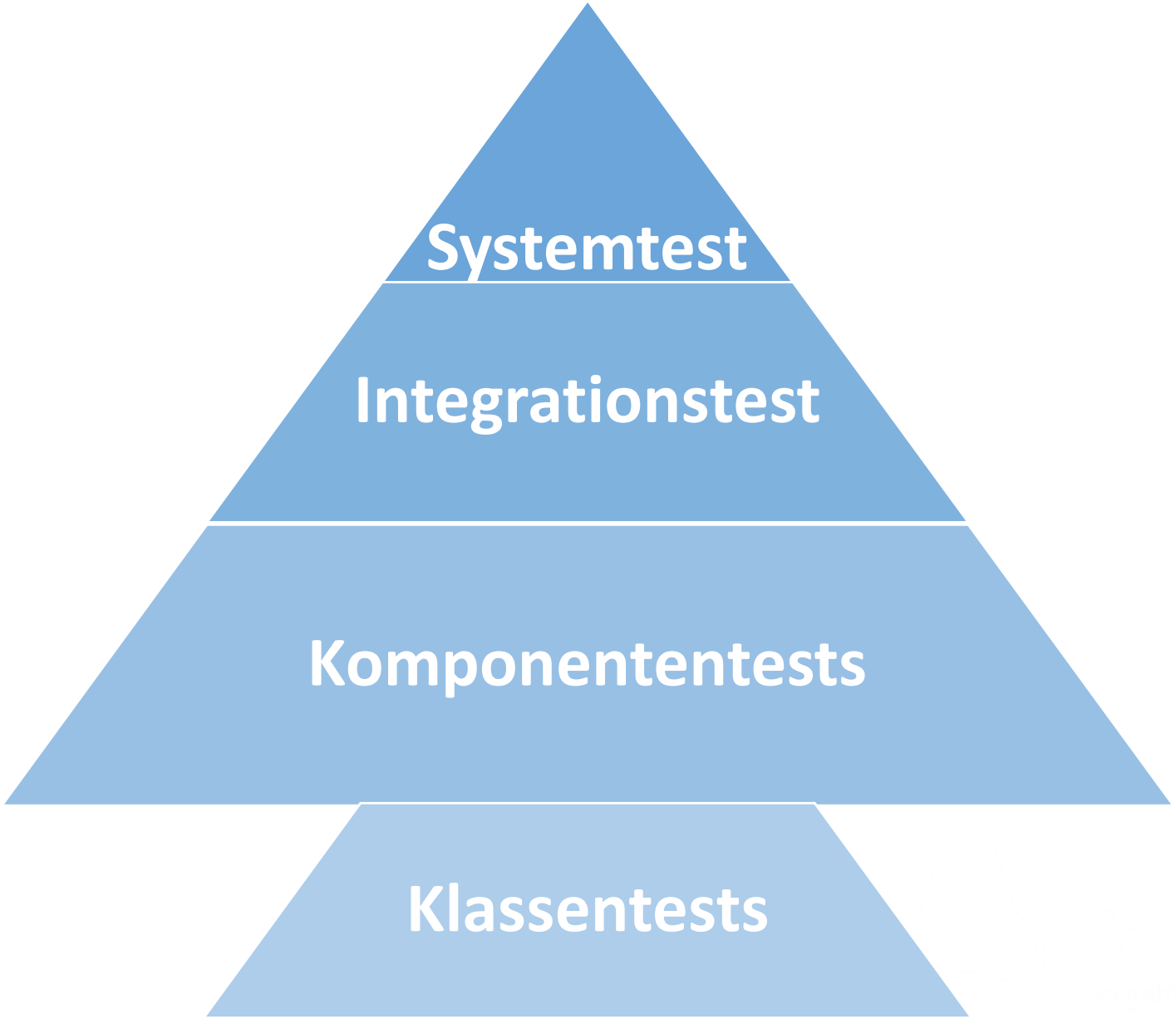
Saxonia Systems
So geht Software.

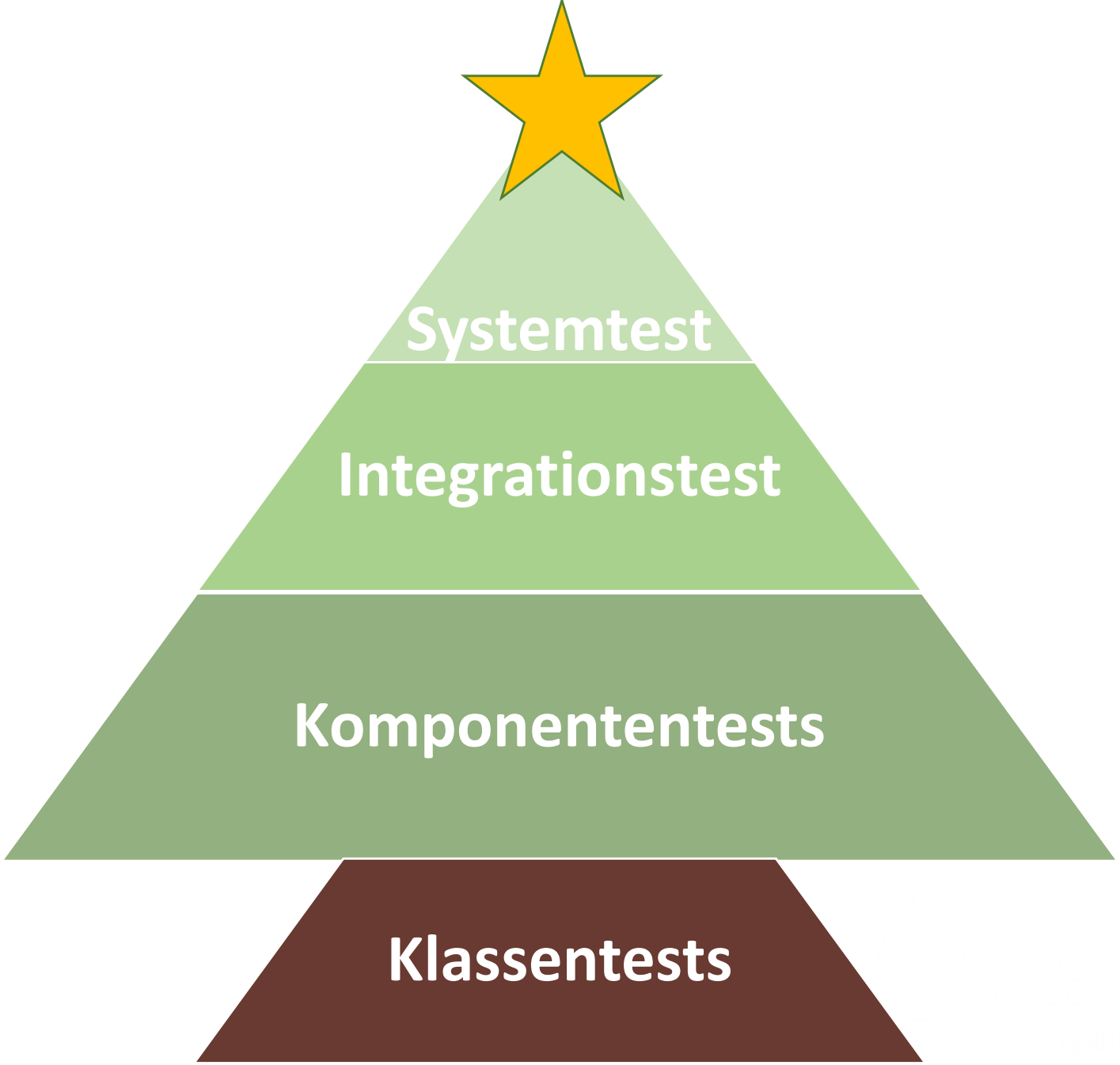


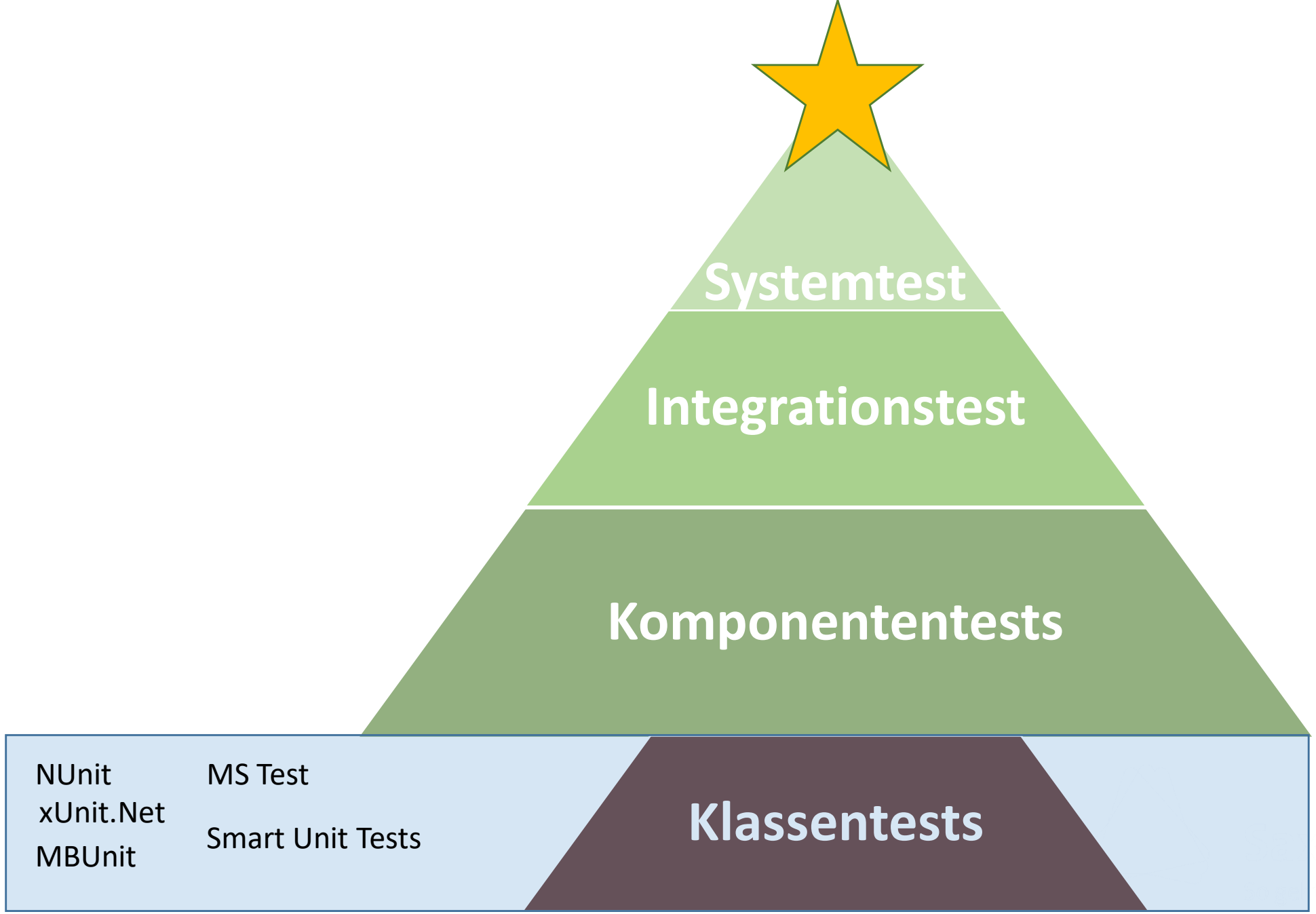
Saxonia Systems
So geht Software.

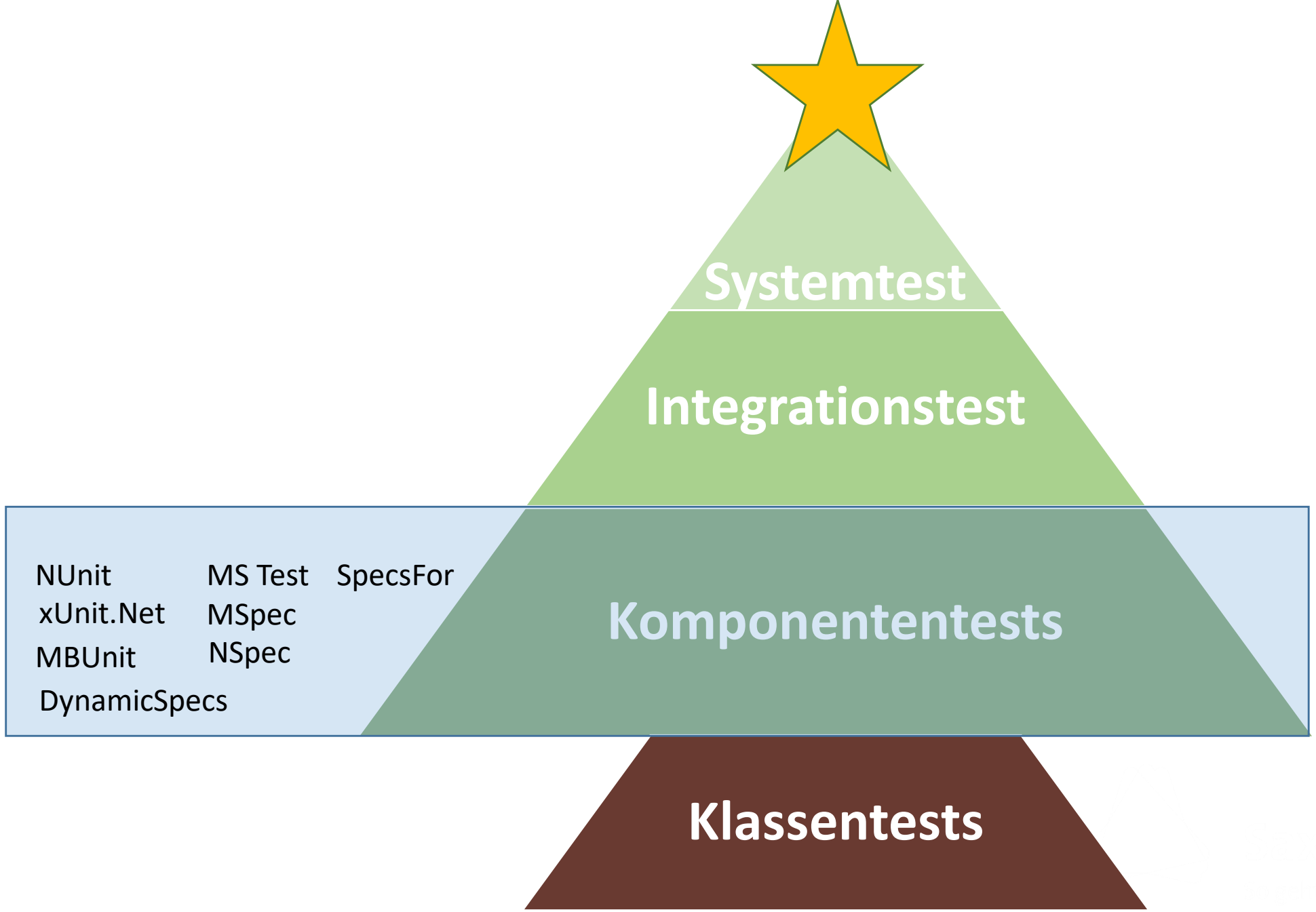



System
Integrations-
test
Komponententest
Klassentests
Unit-Tests


Systemtest
Integrationstest
Komponententests
Klassentests


Systemtest
Integrationstest
Komponententests
Klassentests

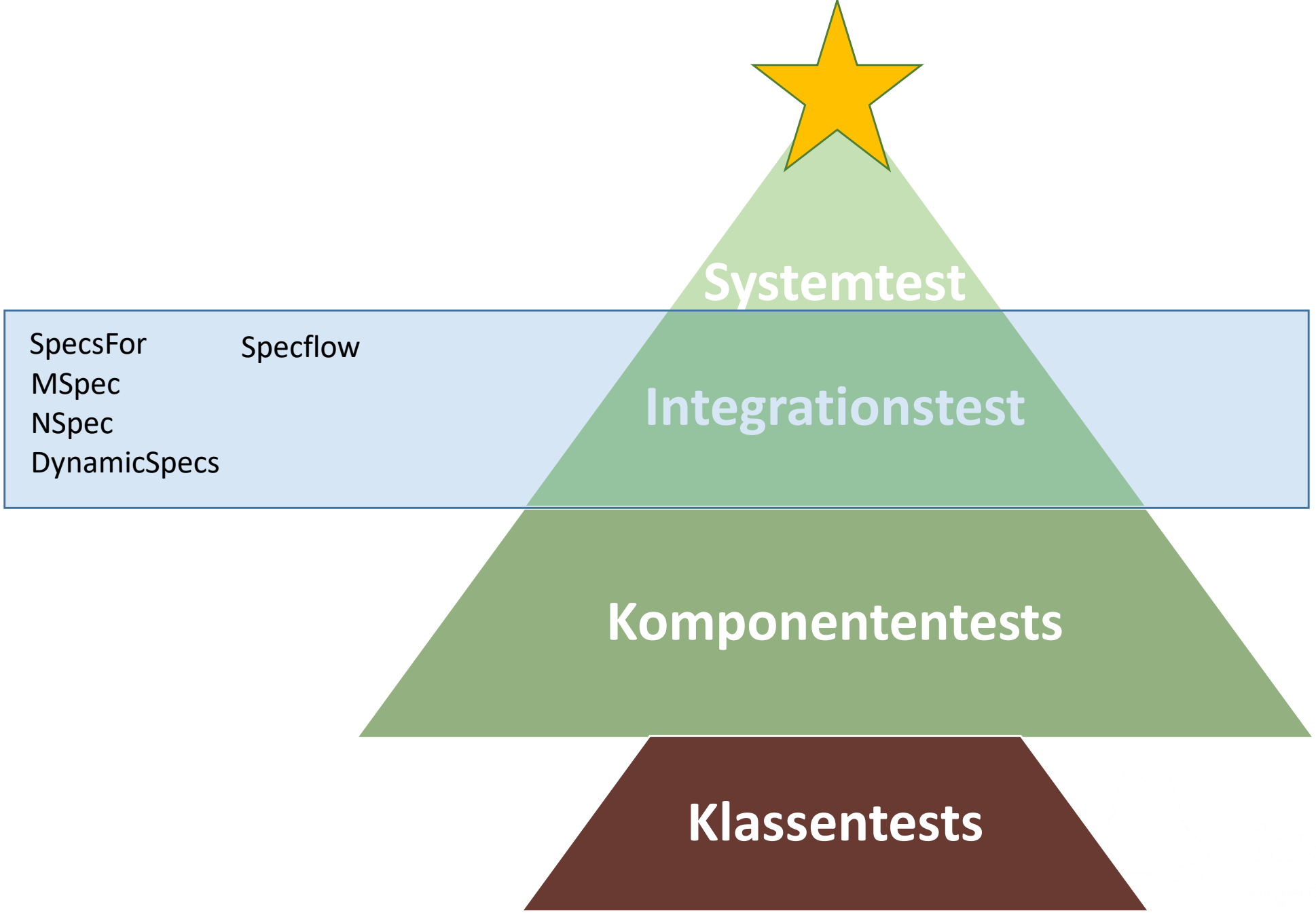



Systemtest
Integrationstest
Komponententests
Klassentests
NUnit
xUnit.Net
MBUnit
MS Test
MSpec
NSpec
SpecsFor
DynamicSpecs

MSpec

```
public class When_no_data_is_entered_for_person
{
   static Person sut;

   Establish context = () => sut = new Person();

   Because of = () =>  { /* nothing todo because we simply don't set any data */ };

   It should_provide_an_error_text_for_BirthDate = () =>  sut["BirthDate"].ShouldNotBeEmpty();

   It should_provide_an_error_text_for_FirstName = () => sut["FirstName"].ShouldNotBeEmpty();

   It should_provide_an_error_text_for_LastName = () =>  sut["LastName"].ShouldNotBeEmpty();
}
```

Saxonia Systems
So geht Software.



Saxonia Systems
So geht Software.

Gherkin mit Specflow

specflow
Cucumber for .NET

```
Funktionalität: Programmstart und Laufzeit
    Um die Applikation nutzen zu können
    muss sie über einen längeren Zeitraum hinweg ohne Absturz funktionieren

Szenario: Programmstart
Angenommen die Applikation wurde gestartet
Wenn 5 Sekunden vergangen sind
Dann sollte die Applikation noch immer aktiv sein

@ignore @Lasttest
Szenario: Laufzeit
Angenommen die Applikation wurde gestartet
Wenn die Applikation 8 Stunden genutzt wird
Dann sollte die Applikation noch immer aktiv sein
```

Gherkin mit Specflow

specflow
Cucumber for .NET

```
[Binding]
public class StartAndRuntimeSteps : StepBase
{
    [Given(@"die Applikation wurde gestartet")]
    public static void AssertThatApplicationWasStarted()
    {
        UiMap.UIFotoMaXWindow.WaitForControlReady(2000);
        UiMap.AssertThatMainWindowsExists();
    }

    [When(@"(.*) Sekunden vergangen sind")]
    public static void Wait(int seconds)
    {
        Playback.Wait(seconds*1000);
    }

    [Then(@"sollte die Applikation noch immer aktiv sein")]
    public static void AssertThatApplicationExists()
    {
        UiMap.AssertThatMainWindowsExists();
    }
}
```

Saxonia Systems
So geht Software.

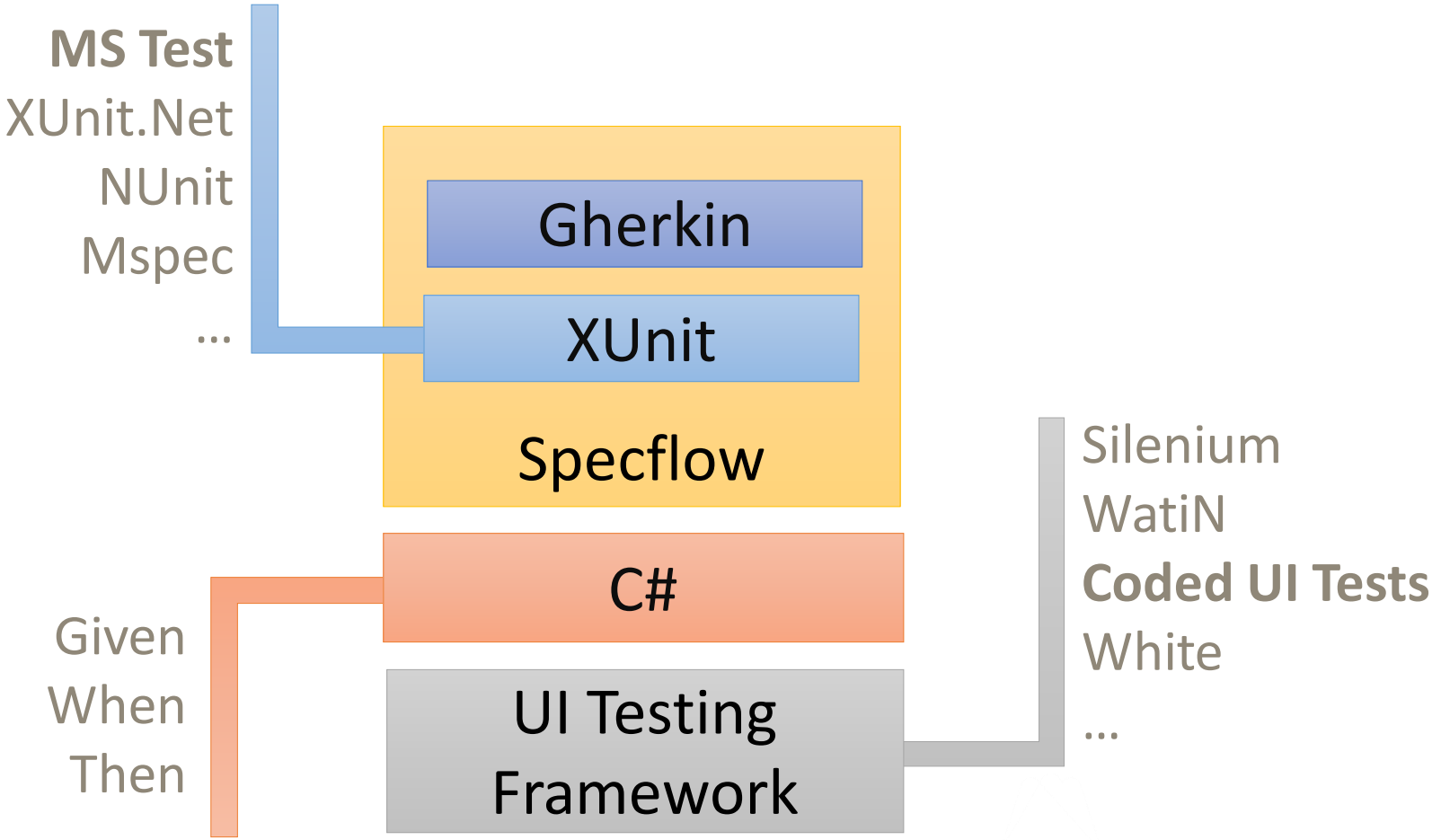
Gherkin mit Specflow

specflow
Cucumber for .NET

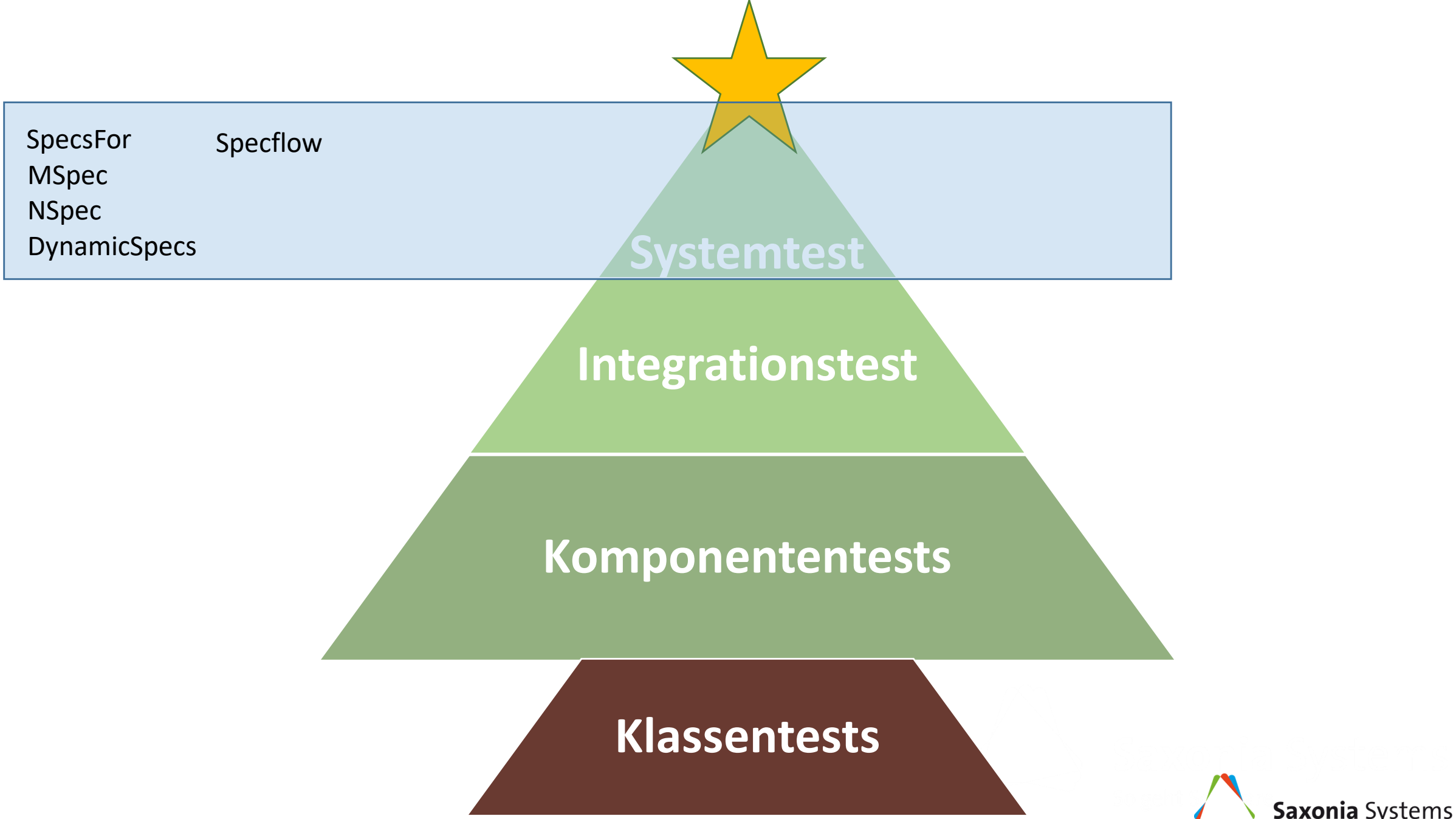
```
Funktionalität: Auftragserteilung
    Um einzelne Bilder entwickeln lassen zu können
    muss ein Auftrag zur Entwicklung dieser Bilder erteilt werden


Szenario: Kostenaufstellung für ein Bild
Angenommen die Entwicklung eines Bildes kostet 10 Cent
Und eine Datenquelle mit Bildern wurde ausgewählt
Wenn ein Bild ausgewählt wird
Und die Parameter der ausgewählten Bilder nicht verändert werden
Und folgende Persönliche Daten eingegeben werden
| Nachname | Vorname | Telefonnummer | Straße       | PLZ   | Ort     |
| Meier    | Karl    | 03510815123   | Grubenweg 6 | 01077 | Dresden |
Und eine Bestellung ausgelöst wird
Dann muss die Kostenaufstellung für die Bestellung ein Bild enthalten
Und der Rechnungsposten für die Entwicklung aller Bilder muss 10 Cent betragen
Und die Rechnung muss einen Rechnungsposten über 2 Euro für die Entwicklung enthalten
Und der Gesamtbetrag muss 2,10 Euro betragen
Und die Kontaktdaten müssen folgenden Inhalt haben
| Nachname | Vorname | Telefonnummer | Straße       | PLZ   | Ort     |
| Meier    | Karl    | 03510815123   | Grubenweg 6 | 01077 | Dresden |
```

Saxonia Systems
So geht Software.

Specflow unter der Haube

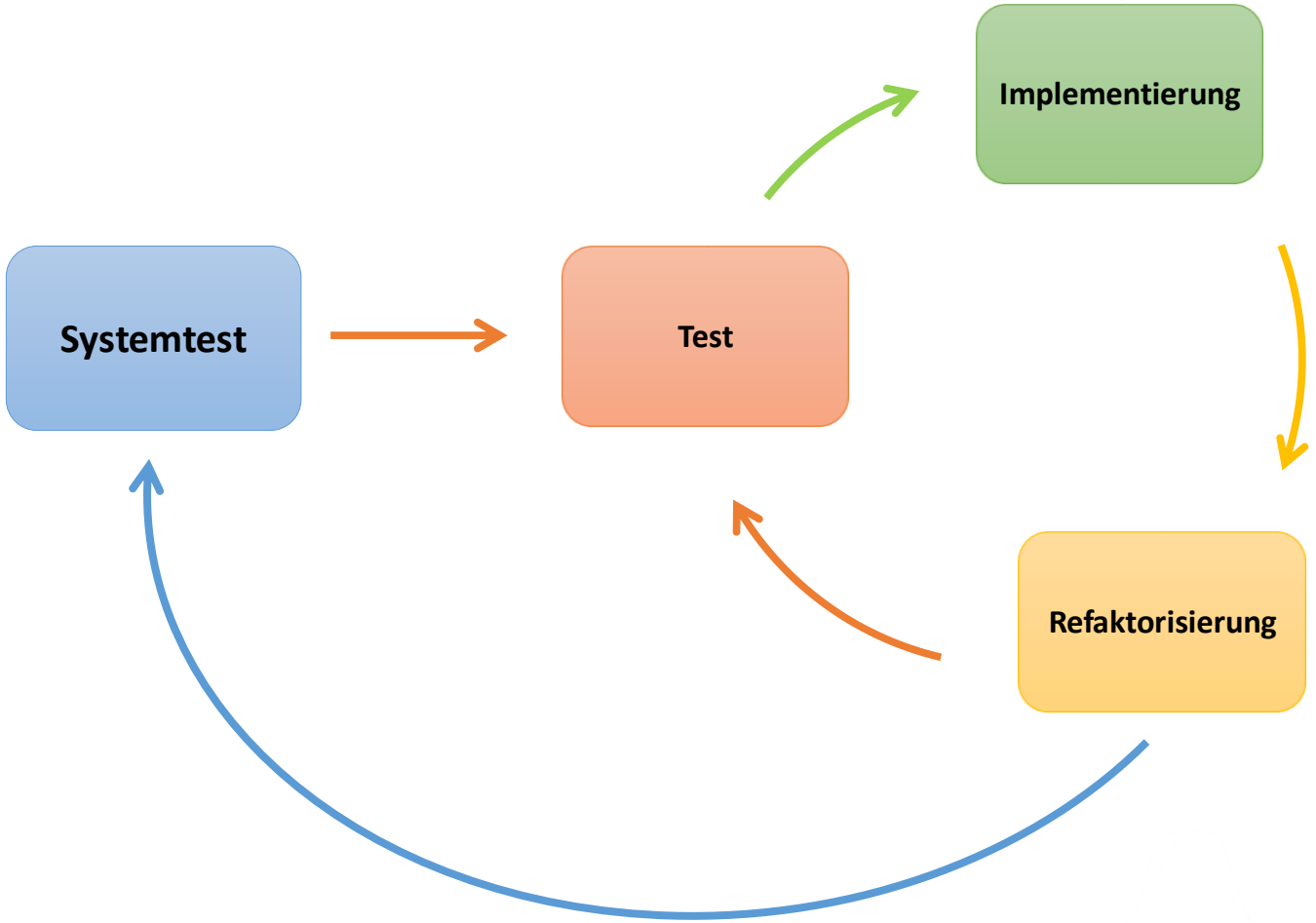


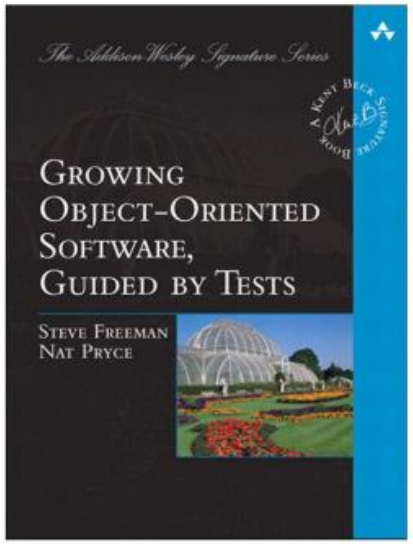
DEMO



Saxonia Systems
So geht Software.

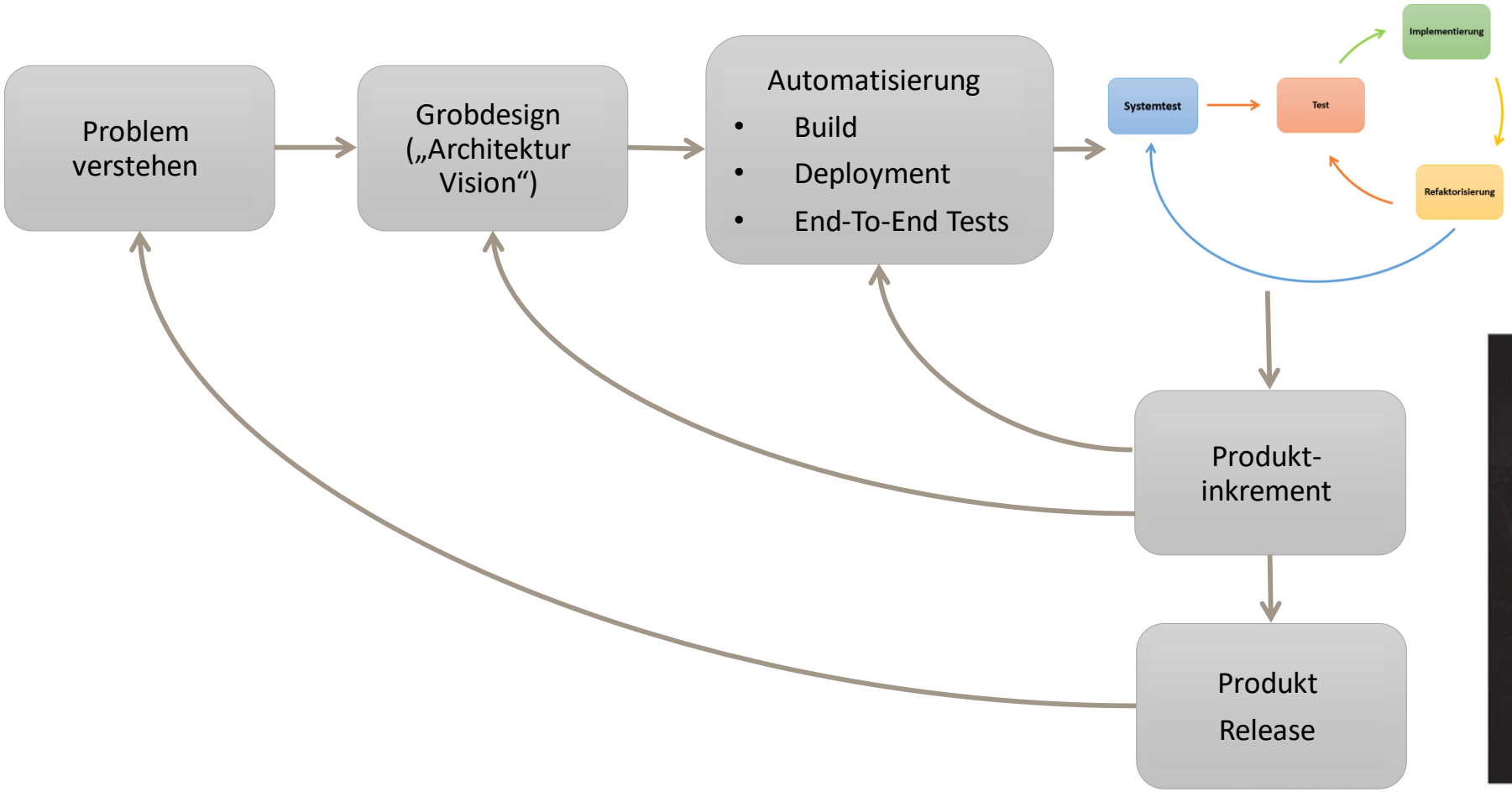
Acceptance Test Driven Development

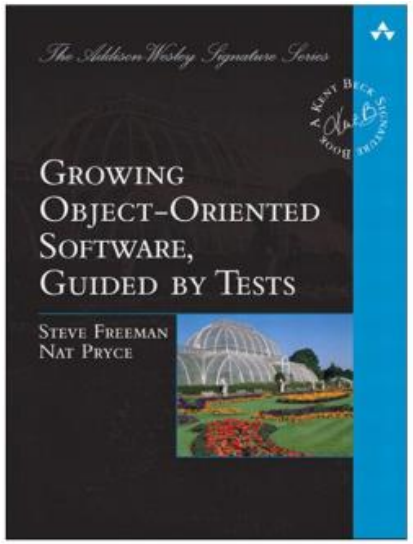




Saxonia Systems
So geht Software.

ATDD im Projektablauf





Saxonia Systems
So geht Software.

Reporting == Dokumentation

Summary

Features	Success rate	Scenarios	Success	Failed	Pending	Ignored
2 features	75%	8	6	1	1	0

Feature Summary

Feature	Success rate	Scenarios	Success	Failed	Pending	Ignored
Calculation	100%	2	2	0	0	0
Save and Load	67%	6	4	1	1	0

Feature Execution Details

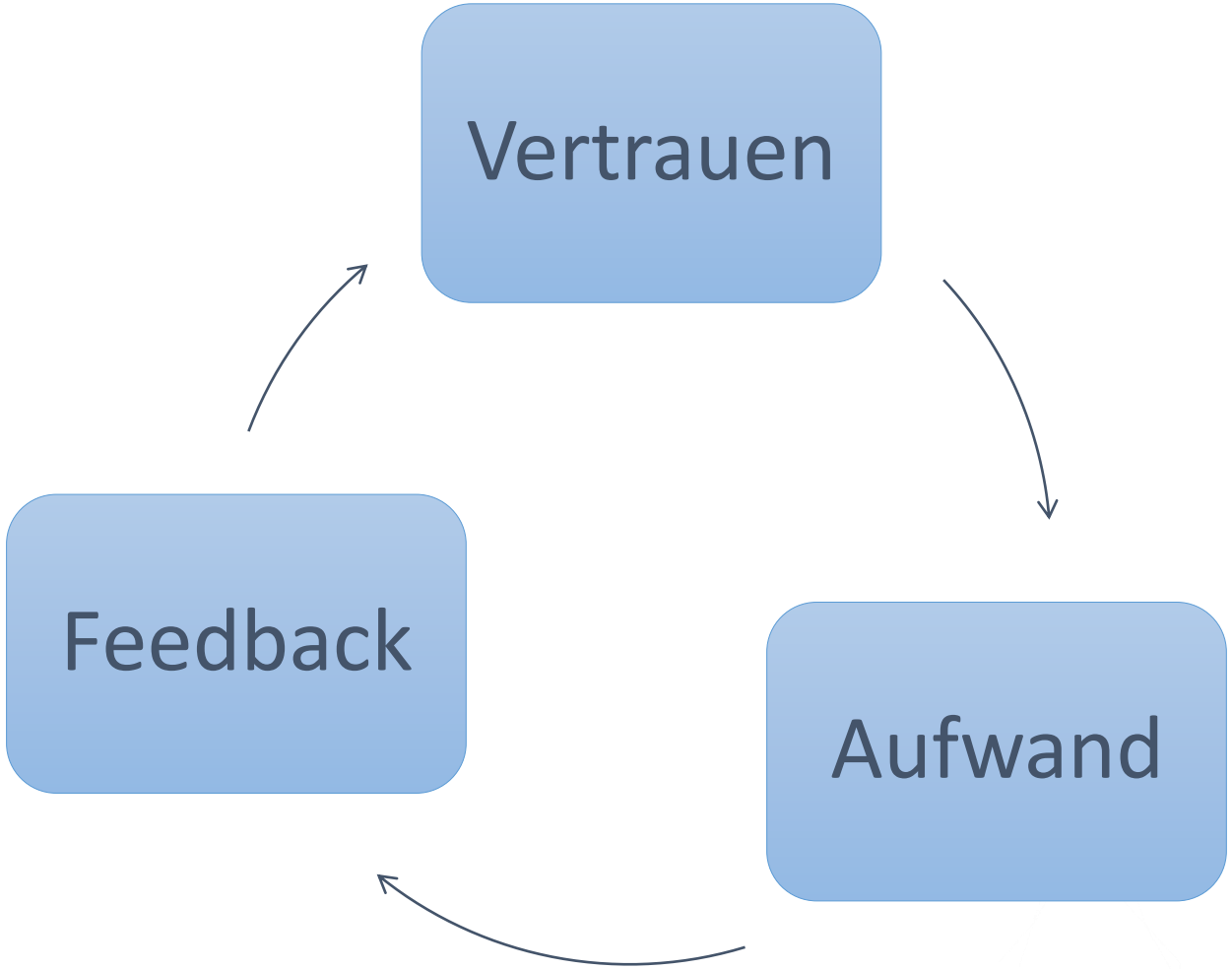
Feature: Calculation

Scenario	Status	Time(s)
multiply two numbers	success	0.102
sum two numbers	success	0.005

Feature: Save and Load

Scenario	Status	Time(s)
Load data as csv	pending	0.081
Load data as xls	success	0.003
Load data as xml	success	0.003
Save data as csv	failure [hide]	0.008

```
System.Exception : Eine Ausnahme vom Typ "System.Exception" wurde ausgelöst.
bei Beispiel.Specs.CalculationSteps.WhenIDoSomethingelse() in C:\Users\Hendrik.loesch
\Documents\Vorträge\Specflow\Beispiel\Beispiel.Specs\CalculationSteps.cs:Zeile 58.
bei lambda_method(Closure )
bei TechTalk.SpecFlow.Bindings.MethodBinding.InvokeAction(Object[] arguments,
ITestTracer testTracer, TimeSpan& duration)
bei TechTalk.SpecFlow.TestRunner.ExecuteStepMatch(BindingMatch match, Object[]
```


Vertrauen
Aufwand
Feedback

Was sind Units?!?

Das was du daraus machst...

Saxonia Systems
So geht Software.

Der Sprecher

Hendrik Lösch

Senior Consultant & Coach

Hendrik.Loesch@saxsys.de

@HerrLoesch

Just-About.Net

WPF-Anwendungen mit MVVM und Prism

Modulare Architekturen verstehen und umsetzen

Windows 8 Store Apps mit MVVM und Prism

XAML-Entwurfsmuster, Bootstrapping, Navigation, Messaging

Test Driven Development – Praxisworkshop

Business-Applikationen testgetrieben entwickeln

Inversion of Control und Dependency Injection

Prinzipien der modernen Software-Architektur ...

Test Driven Development mit C#

Grundlagen, Frameworks, best Practices

Automatisiertes Testen mit Visual Studio 2012

Grundlagen, Testarten und Strategien

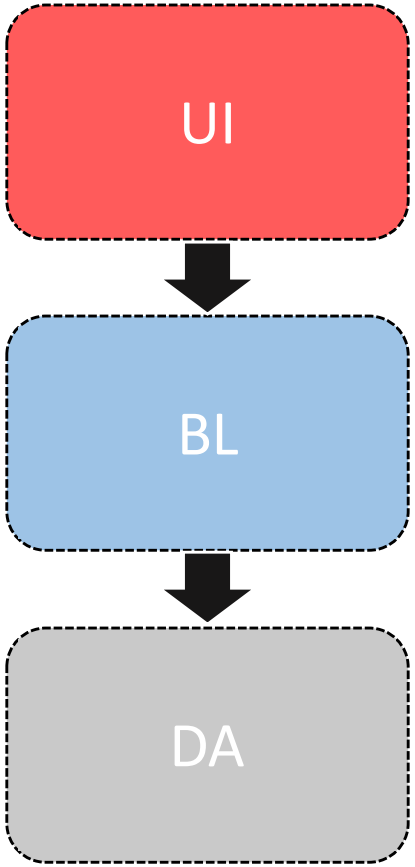
Saxonia Systems

So geht Software.

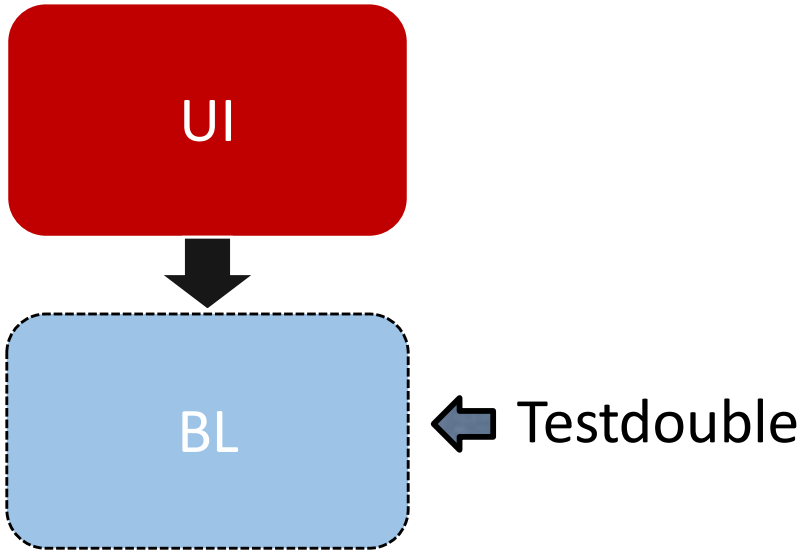
Die 4 Arten von Unit Tests

- Warmup Tests
- Logic Tests
- Solution Tests
- Bug Tests

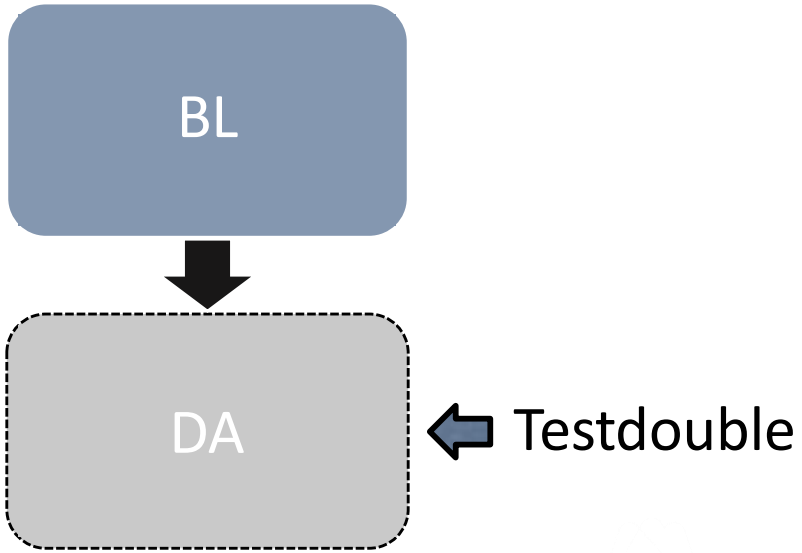
Acceptance Test Driven Development



Acceptance Test Driven Development



Acceptance Test Driven Development



Acceptance Test Driven Development