

SURVIVAL KIT FÜR CODEHAUSMEISTER



Saxonia Systems
So geht Software.

Der Sprecher



Hendrik Lösch

Senior Consultant

Hendrik.Loesch@saxsys.de

@HerrLoesch

Hendrik-Loesch.de



ReSharper lernen
Effektiver programmieren mit der Erweiterung zu
Visual Studio
[Hendrik Lösch](#)



WPF-Anwendungen mit MVVM und Prism
Modulare Architekturen verstehen und umsetzen
[Hendrik Lösch](#)



Windows 8 Store Apps mit MVVM und Prism
XAML-Entwurfsmuster, Bootstrapping, Navigation,
Messaging
[Hendrik Lösch](#)



LINQ Grundkurs
Wichtige Spracheigenschaften von C#, Joins,
Variablen und andere Operationen, erweiterte
Techniken
[Hendrik Lösch](#)

LinkedIn Learning



**Grundlagen der Programmierung: Test Driven
Development**
Business-Anwendungen testgetrieben entwickeln
[Hendrik Lösch](#)



**Inversion of Control und Dependency
Injection – Grundlagen**
Prinzipien der modernen Software-Architektur ...
[Hendrik Lösch](#)



C#: Test Driven Development
Grundlagen, Frameworks, best Practices
[Hendrik Lösch](#)



**Grundlagen der Programmierung:
Codemetriken**
Softwarequalität einschätzen, sicherstellen und ...
[Hendrik Lösch](#)



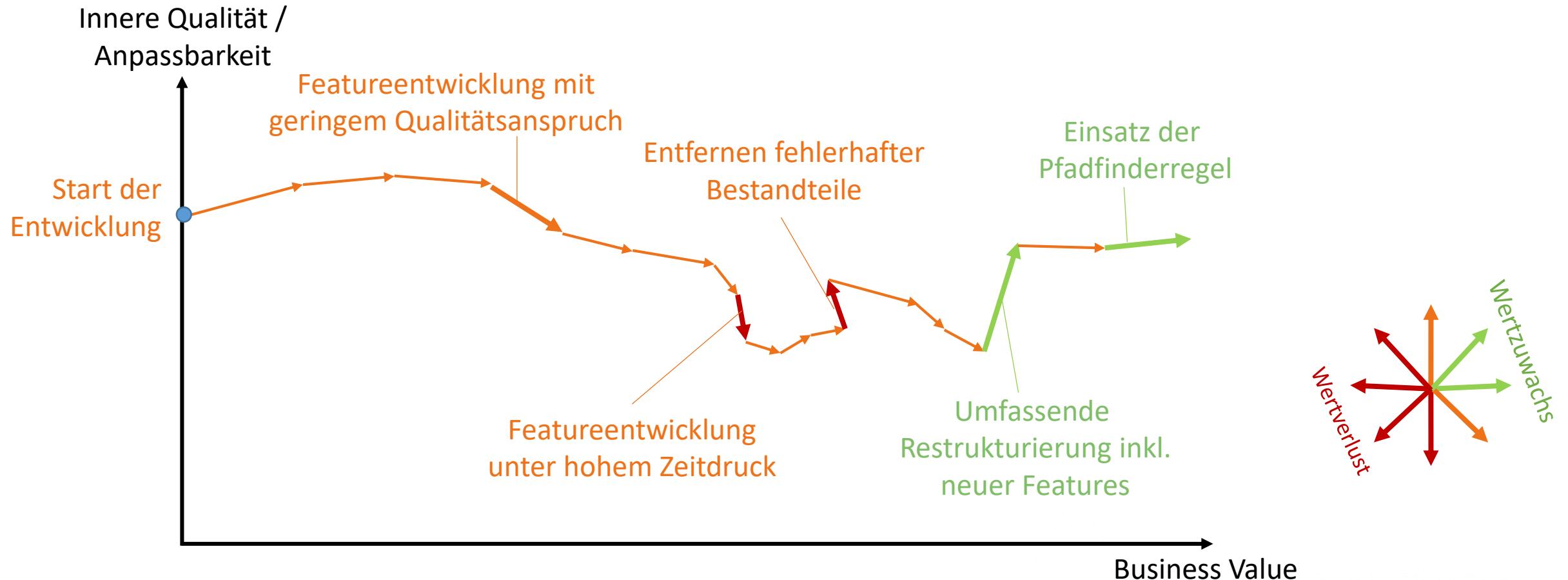
Saxonia Systems
So geht Software.

AUSGANGSSITUATION

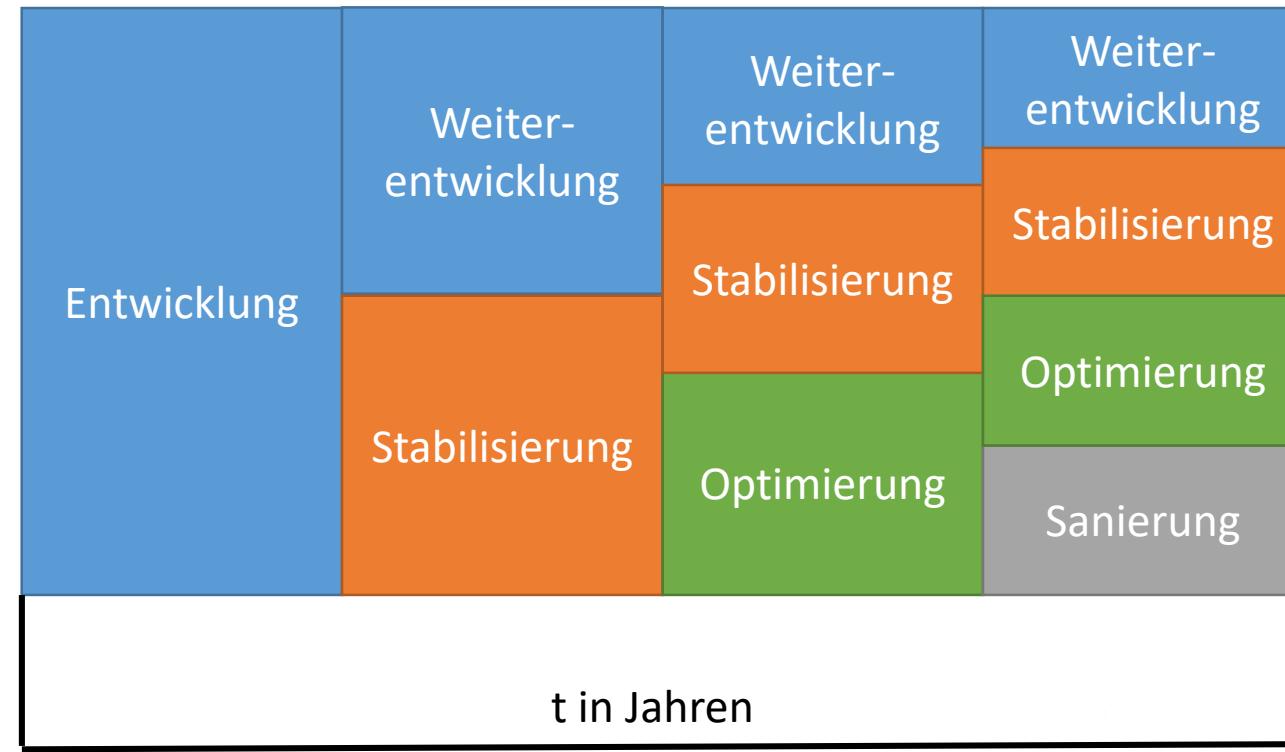


Saxonia Systems
So geht Software.

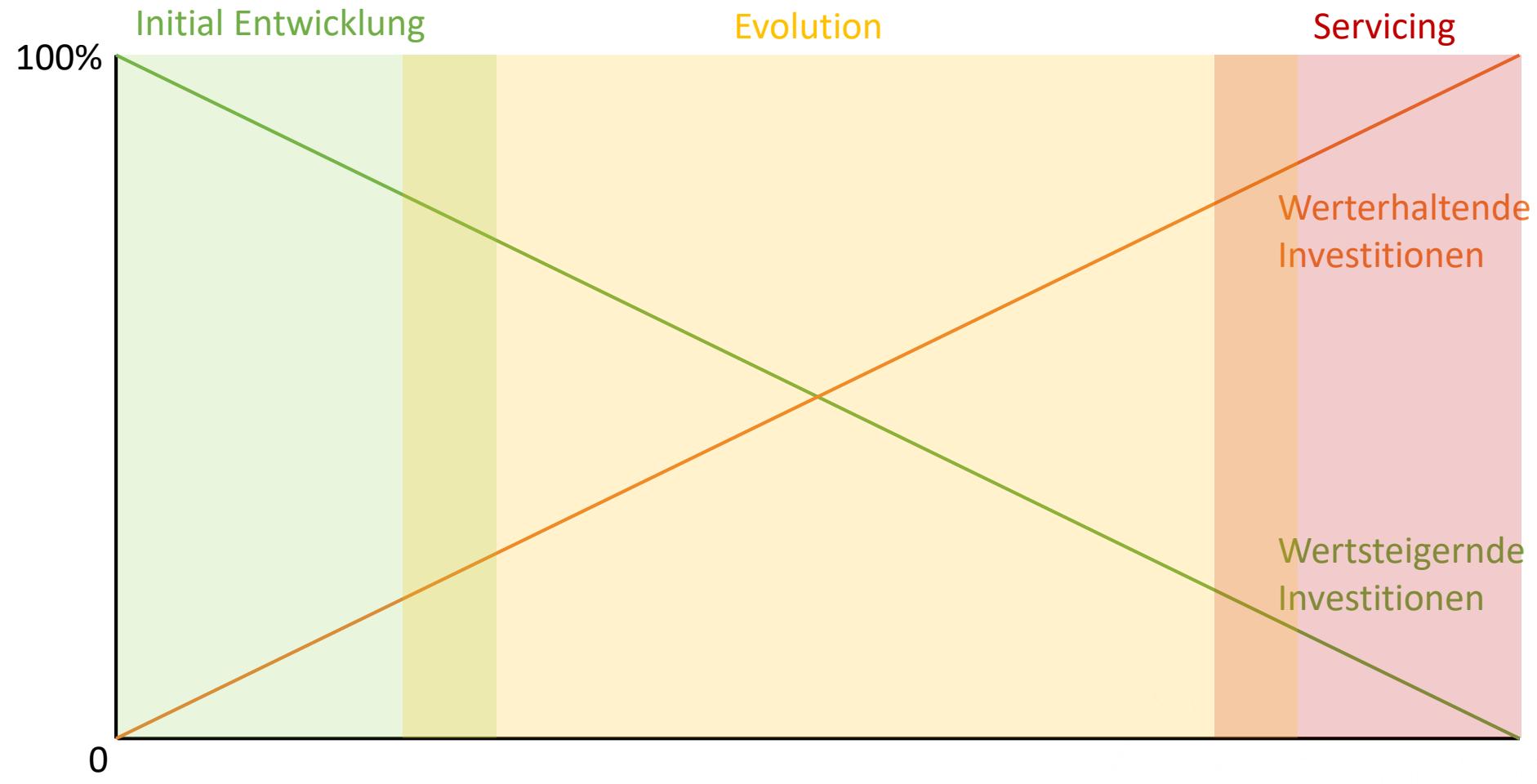
Anpassbarkeit vs. Business Value



Werterhaltung



Werterhaltung



Saxonia Systems
So geht Software.



CODEHAUSMEISTER



ZIELE

- Instandhaltung der Immobilie
- Zufriedene Bewohner
- Vermeidung und Beseitigung von Schäden an der Anlagen



AUFGABEN

- Betreuung und Verwaltung einer Immobilie im Bezug auf Zustand und Technik
- Ansprechpartner für die Bewohner und Besitzer
- Wartung von Anlagen
- Einleitung von Instandhaltungs-, Renovierungs- und Wartungsarbeiten ggf. durch Subunternehmer



Vertrauen



Feedback

Aufwand



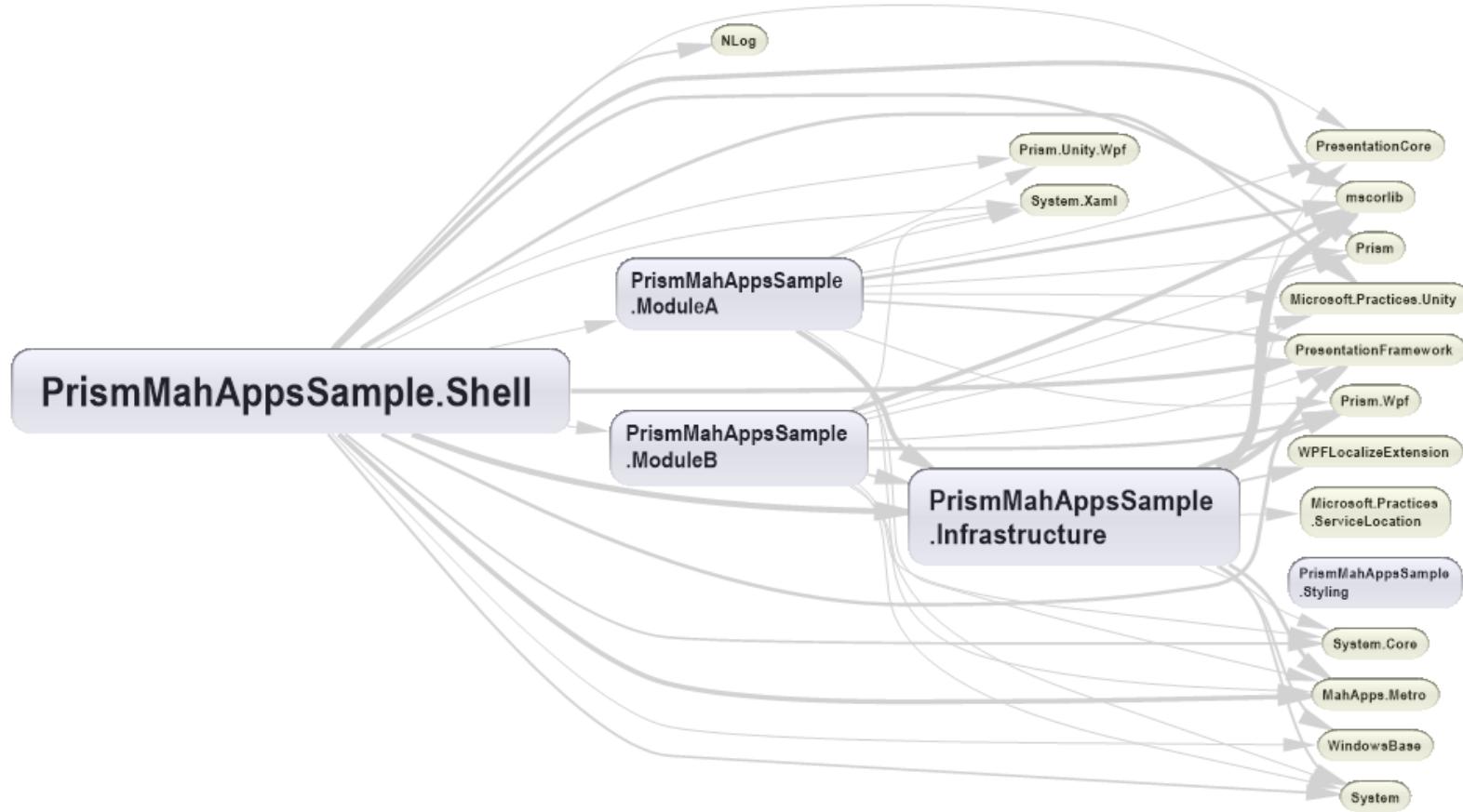
Saxonia Systems
So geht Software.

STRUKTUR & ARCHITEKTUR

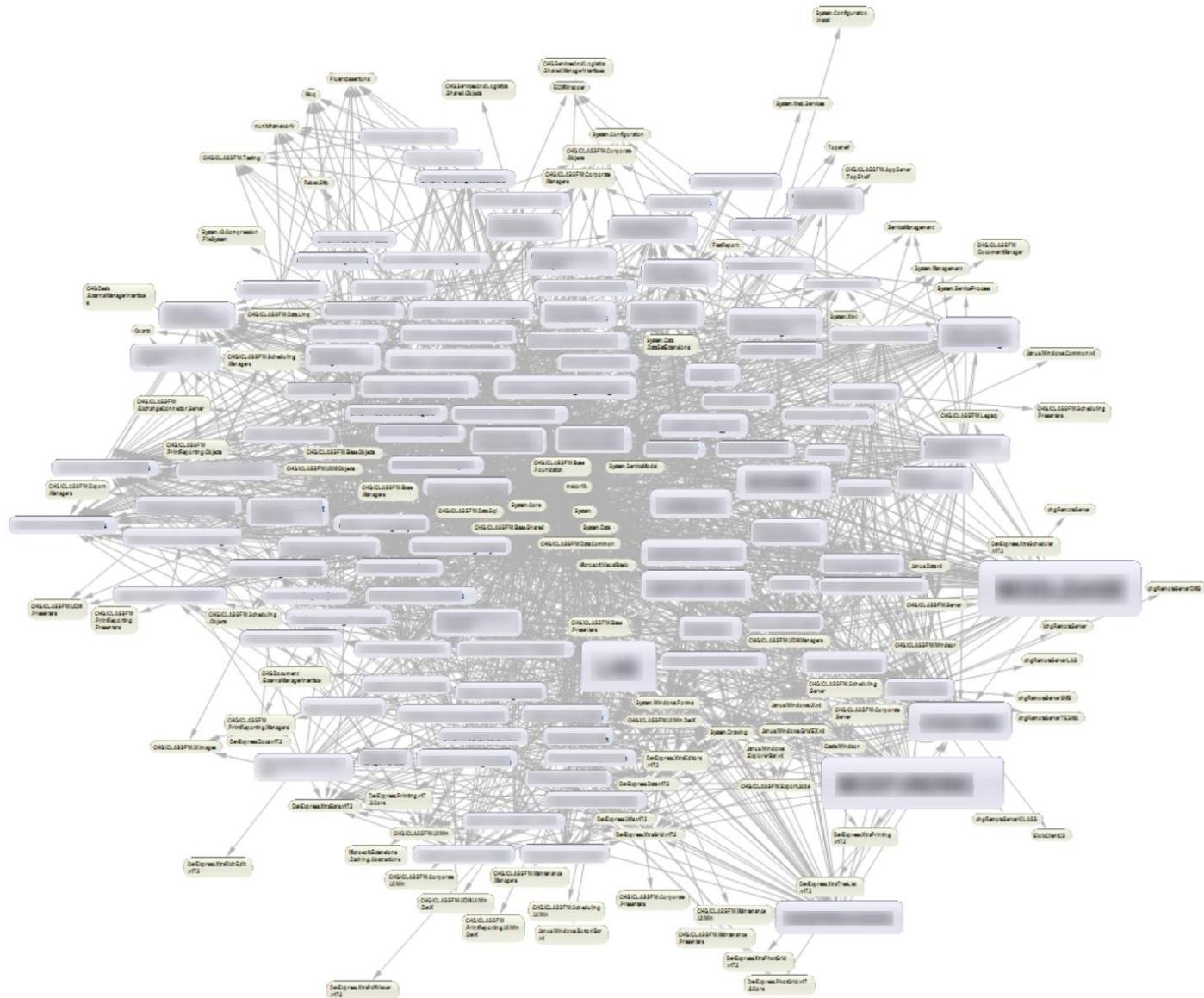


Saxonia Systems
So geht Software.

Abhängigkeitsgraph



Abhangigkeitsgraph



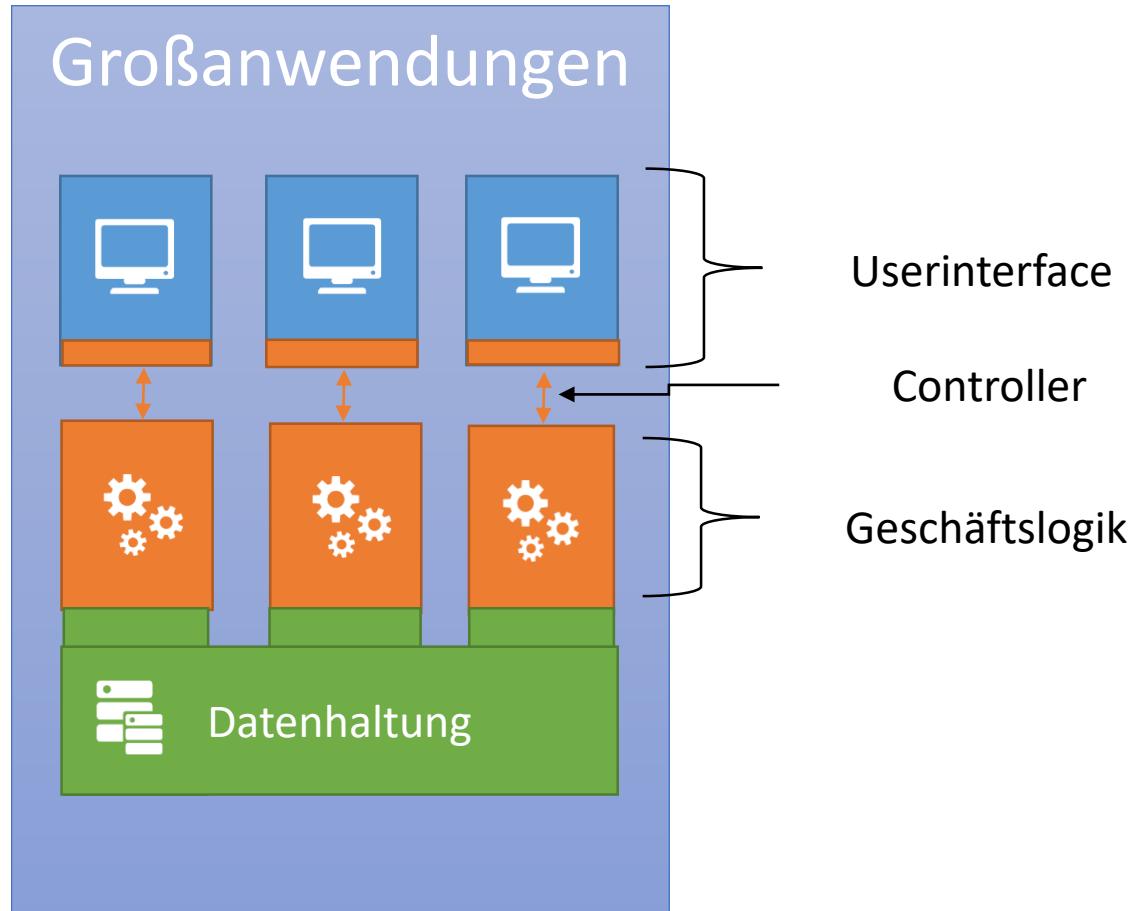
Saxonia Systems

So geht Software.

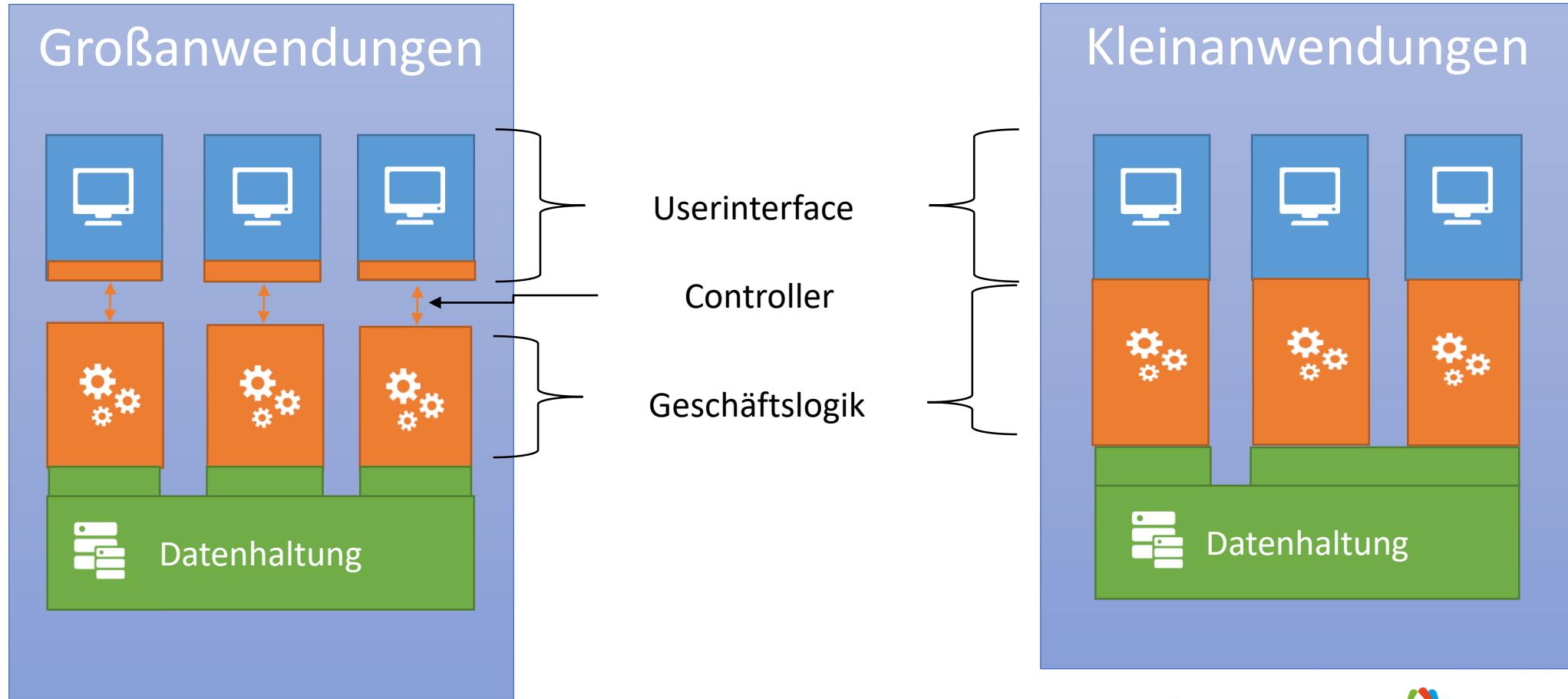
Abhängigkeiten



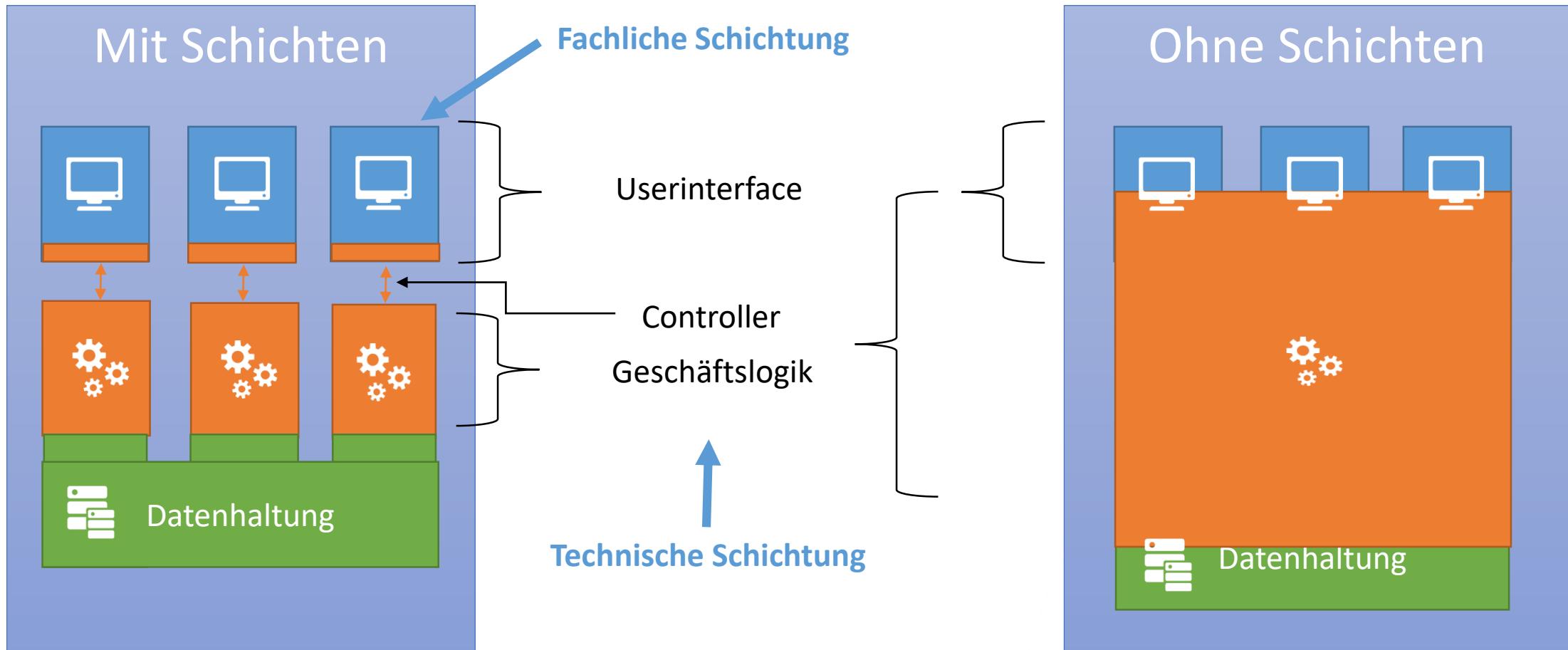
Schichtentrennung



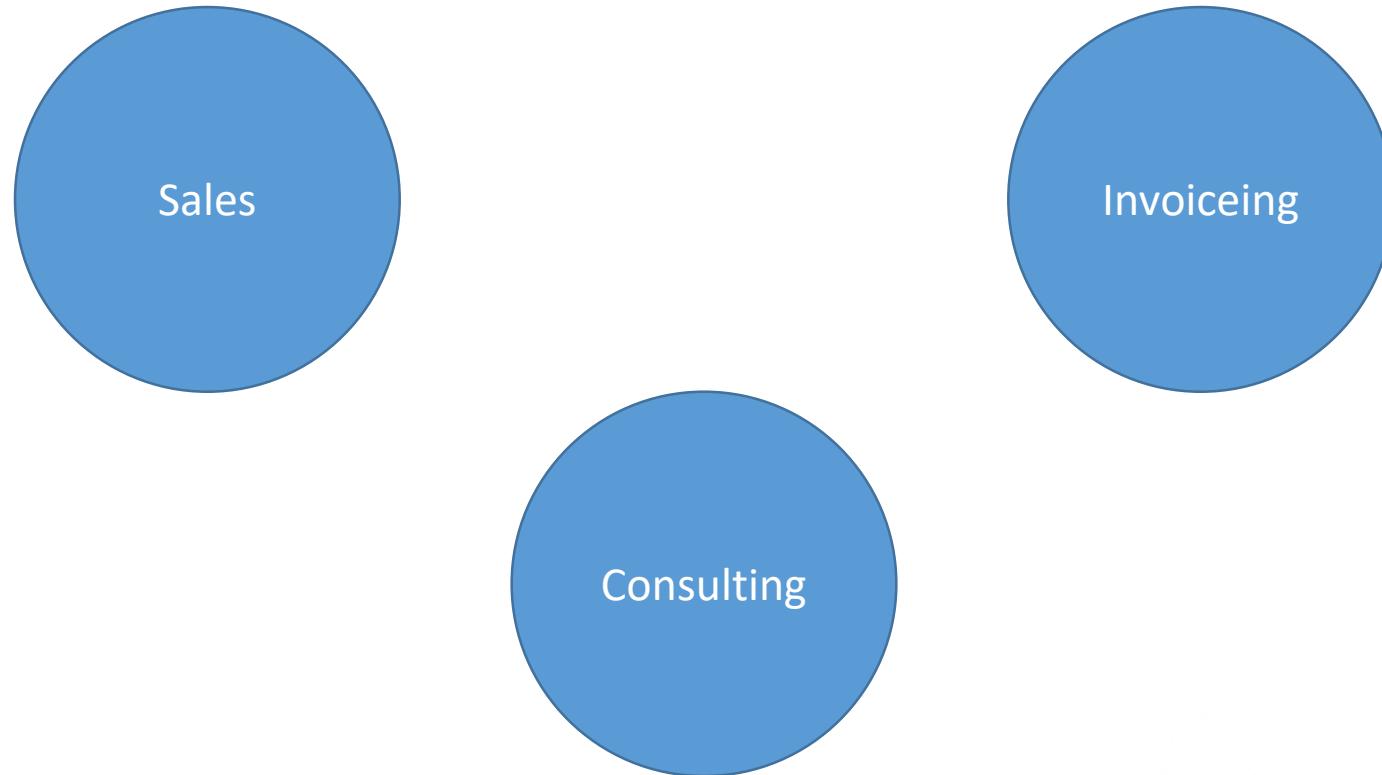
Schichtentrennung



Schichtentrennung

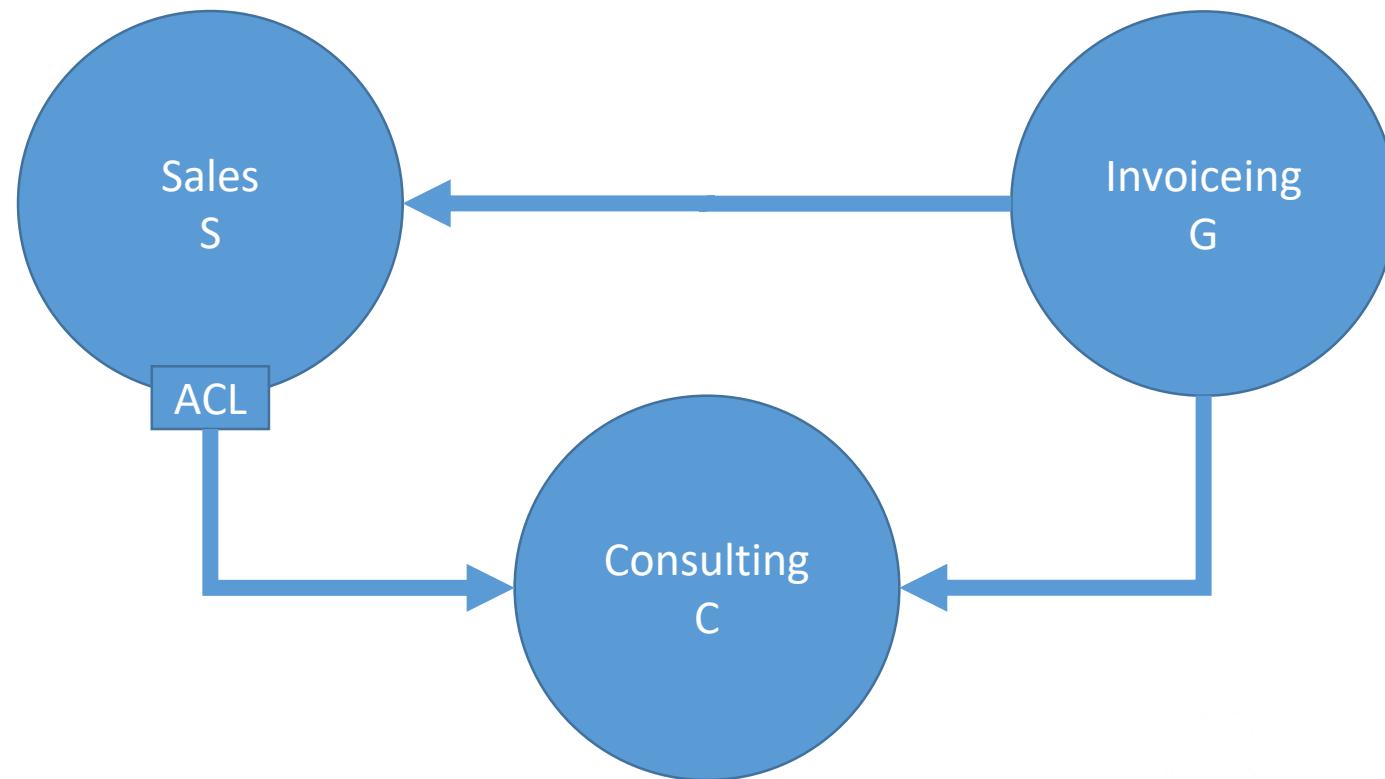


Context Map



Saxonia Systems
So geht Software.

Context Map



C – Core Domain

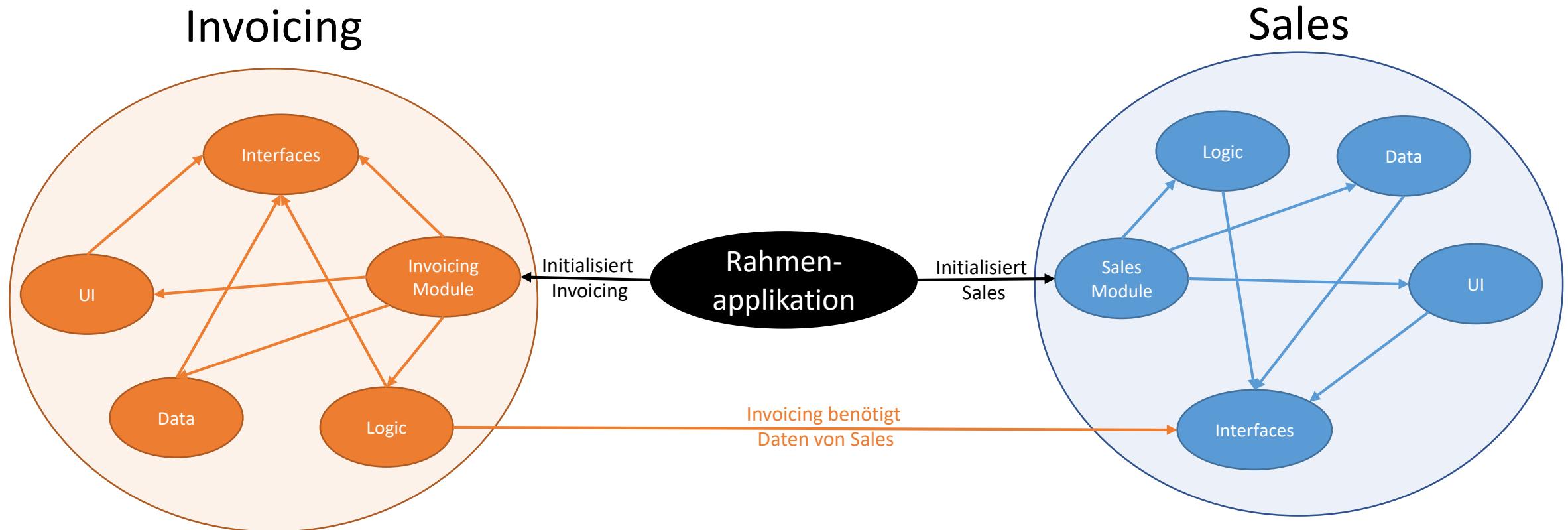
S – Support Domain

G – Generic Domain

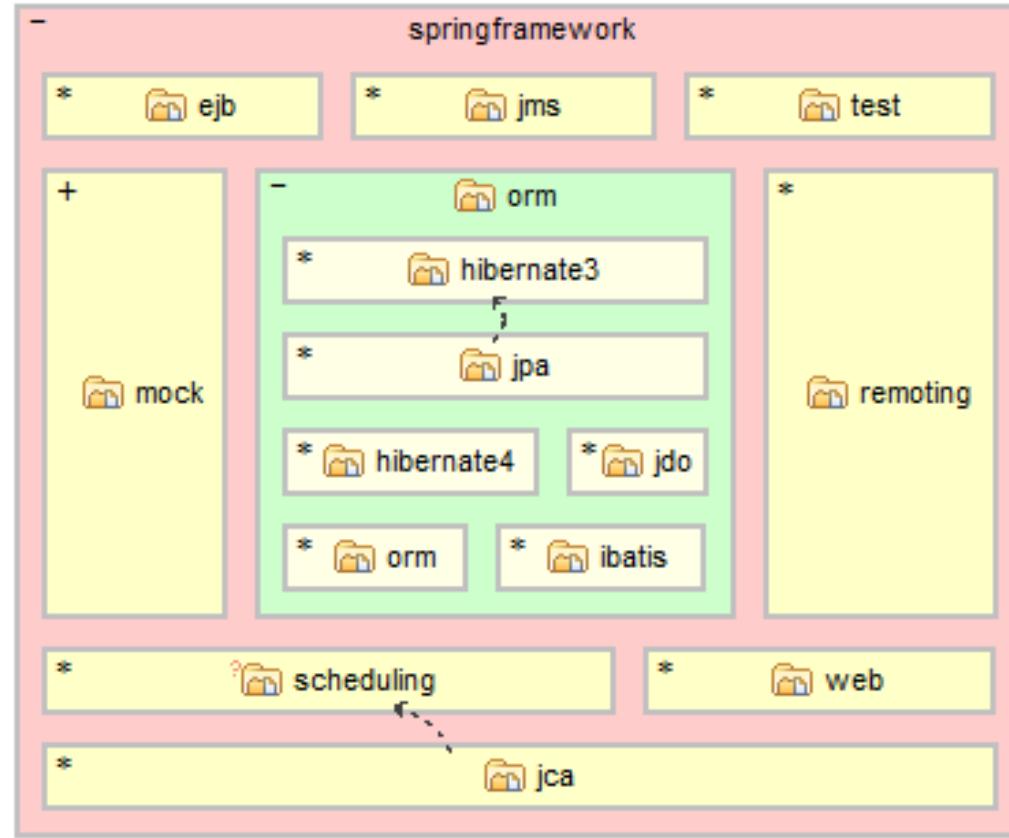


Saxonia Systems
So geht Software.

Zielzustand



Architekturdiagramm



Projektstruktur



Quelle: <http://butunclebob.com/Articles.UncleBob.PrinciplesOfOOD>

QUALITÄTSSICHERUNG

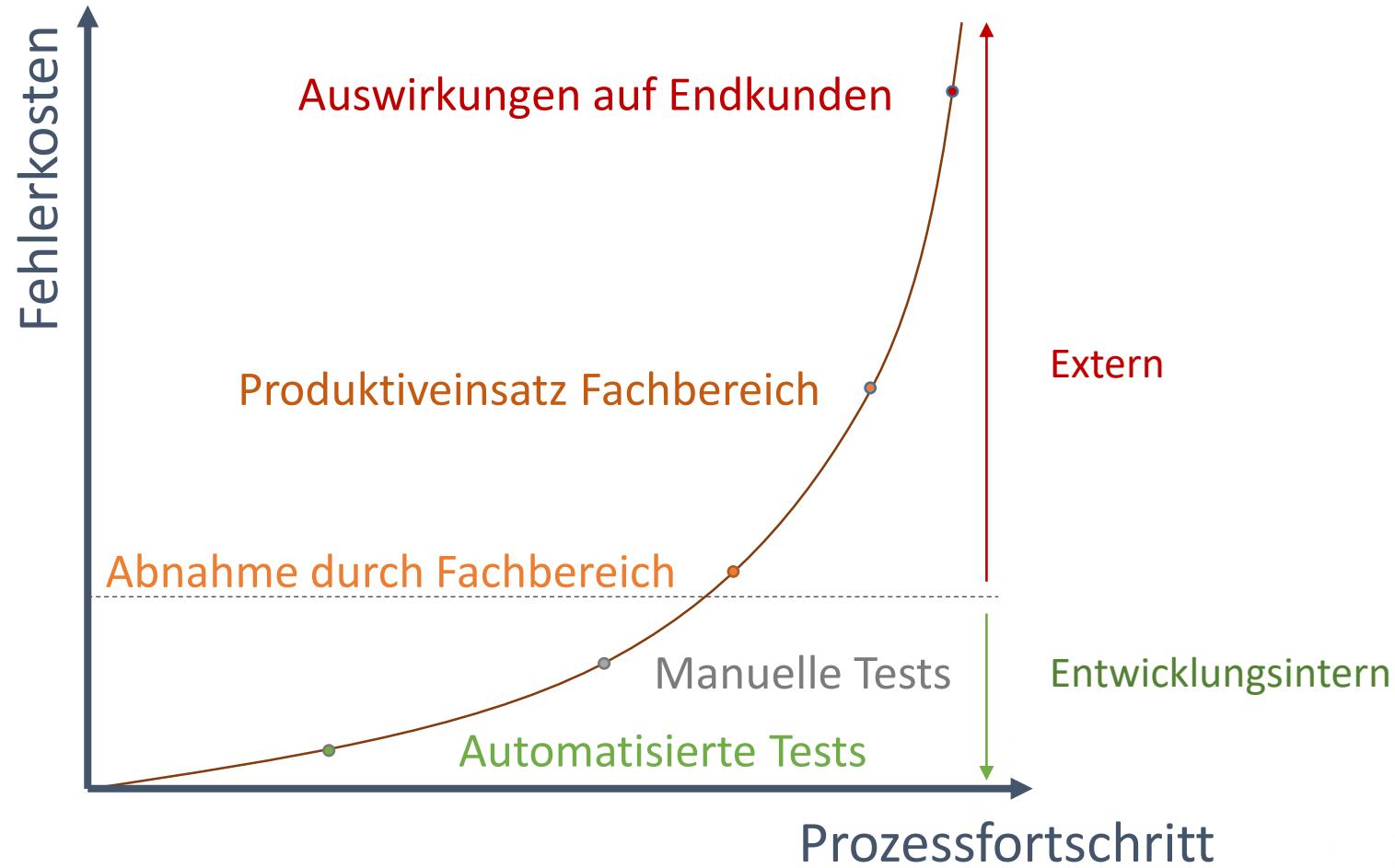
Verifizierung und Validierung

Bei der Verifizierung wird geprüft, ob ein Produkt bei seiner Entwicklung mit den spezifizierten Anforderungen, festgehalten im Pflichtenheft eines Unternehmens, übereinstimmt, während die Validierung eine Art Feldexperiment ist, bei dem kontrolliert wird, ob festgelegte Nutzungsziele erfüllt sind und somit die Anforderungen des Kunden auf Tauglichkeit überprüft.

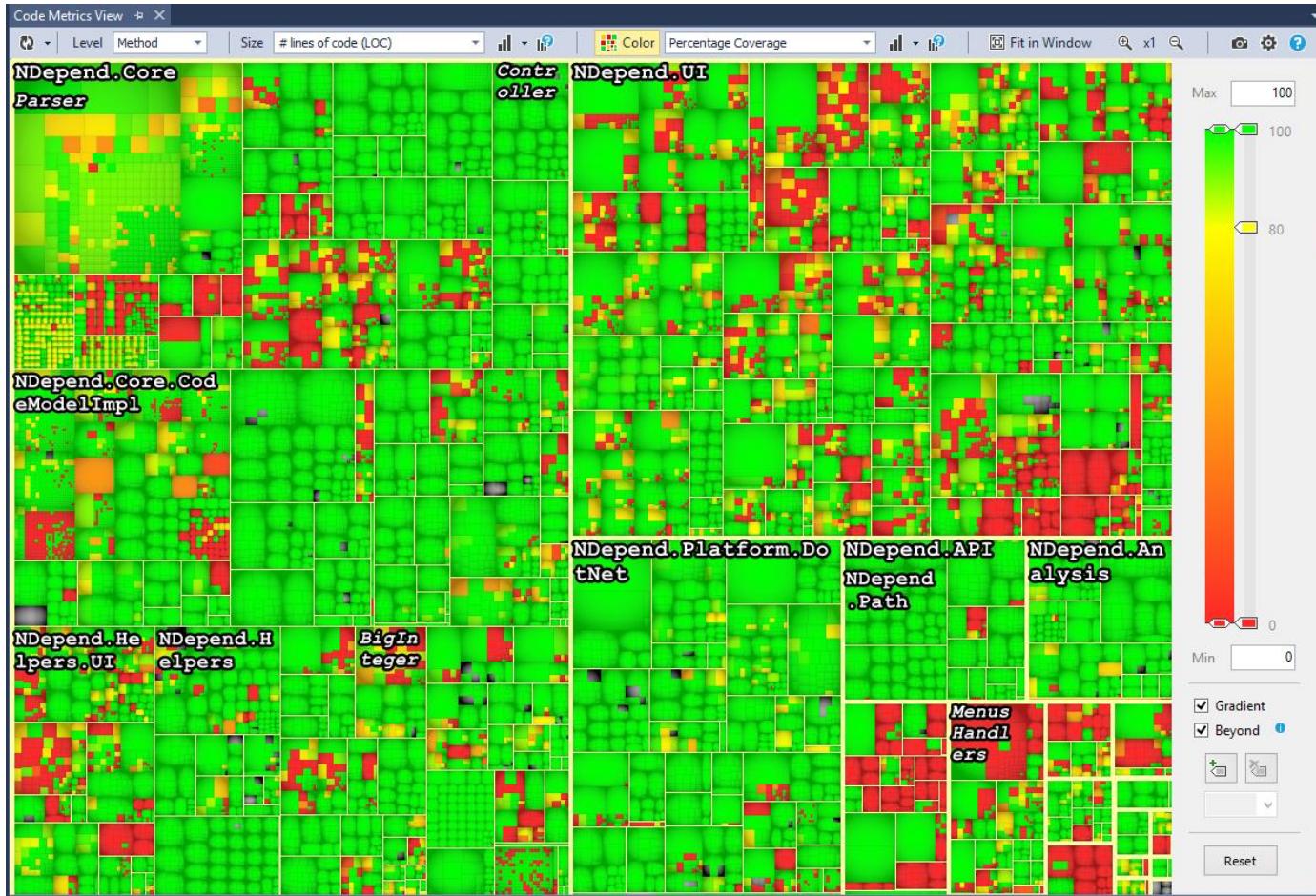
W [Mehr auf Wikipedia \(DE\)](#)



Fehlerkosten

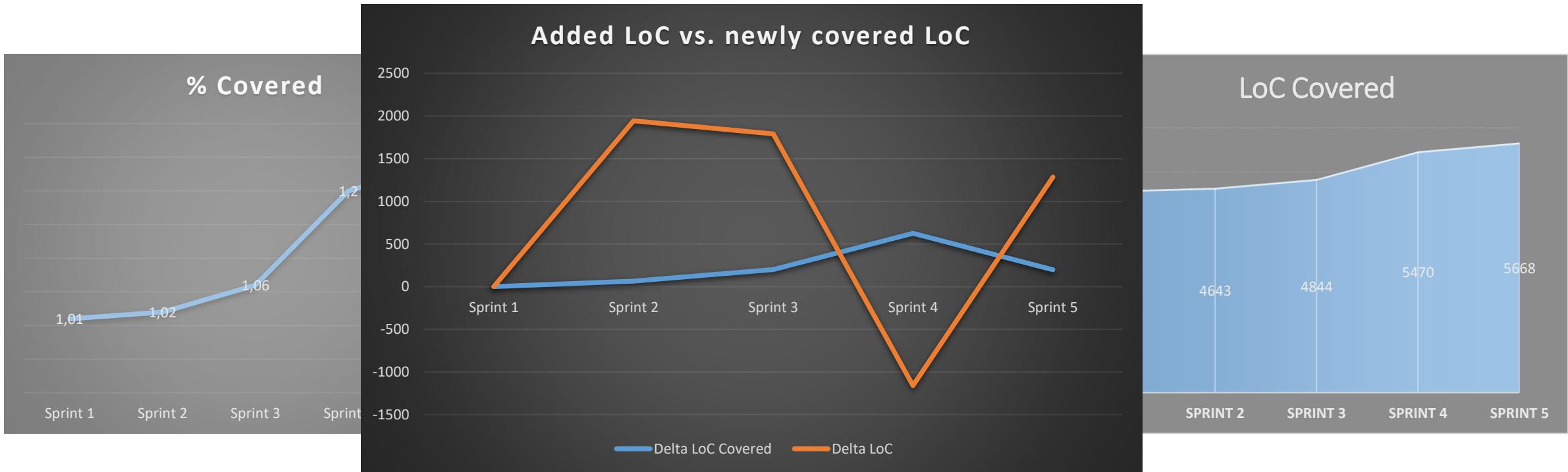


Code Coverage



Saxonia Systems
So geht Software.

Code Coverage ist nicht alles



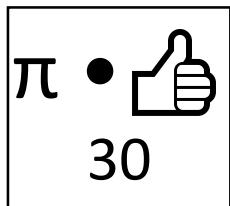
	% Covered
Sprint 1	1,01
Sprint 2	1,02
Sprint 3	1,06
Sprint 4	1,2
Sprint 5	1,24



Saxonia Systems
So geht Software.

C.R.A.P.

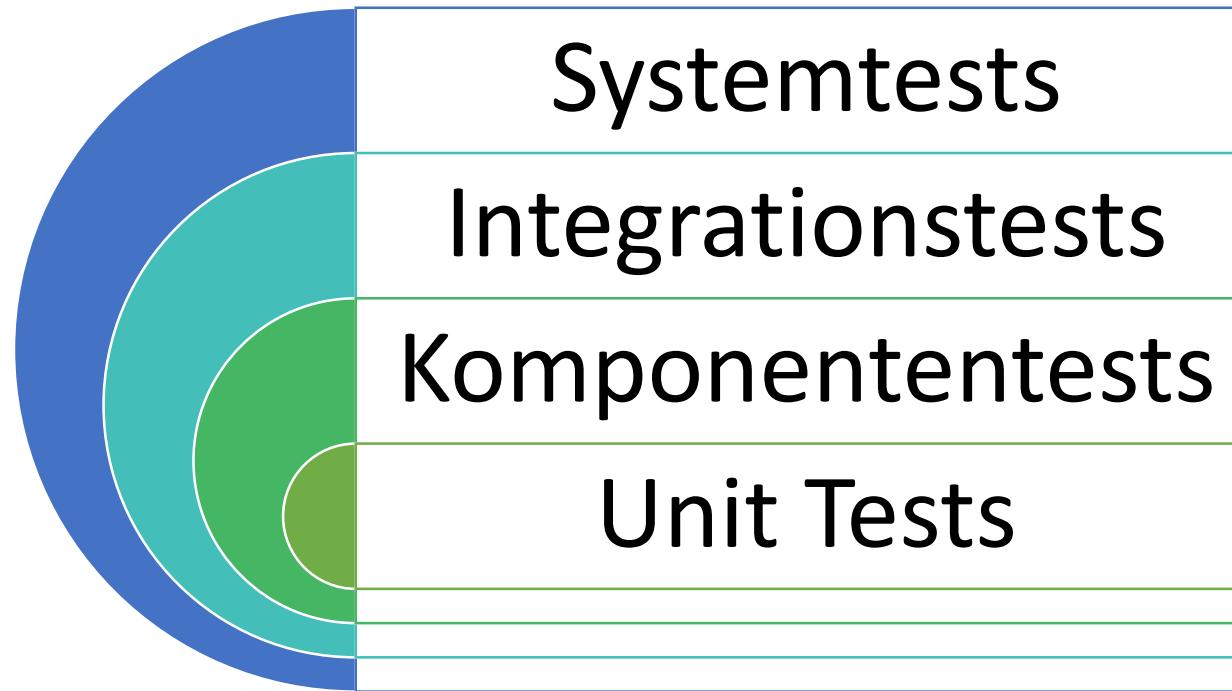
$$C.R.A.P.(m) = CC(m)^2 * (1 - Coverage(m)/100)^3 + CC(m)$$



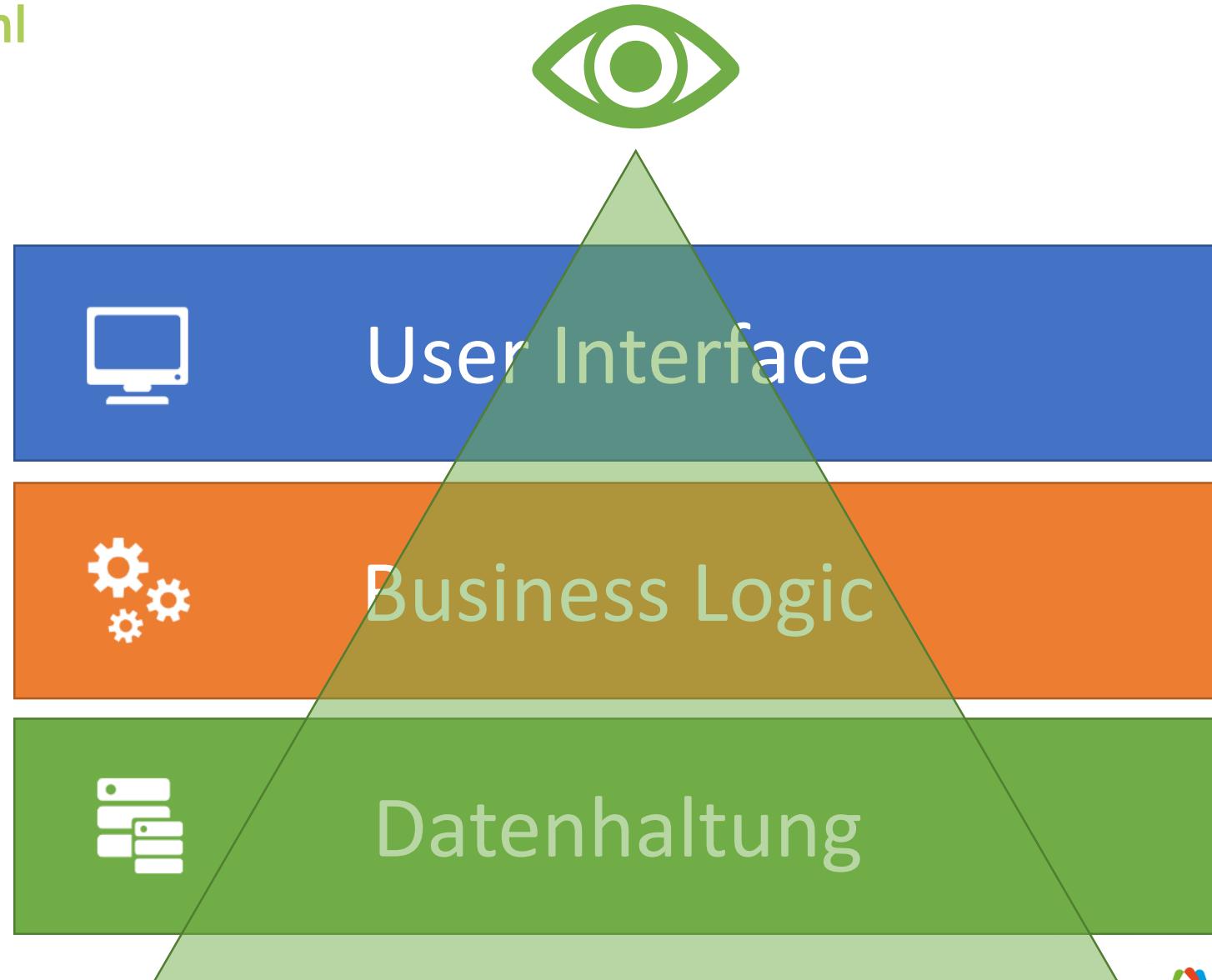
Method's CC	% of coverage required to be below CRAPpy threshold
0 – 5	0%
10	42%
15	57%
20	71%
25	80%
30	100%
31+	No amount of testing will keep methods this complex out of CRAP territory.



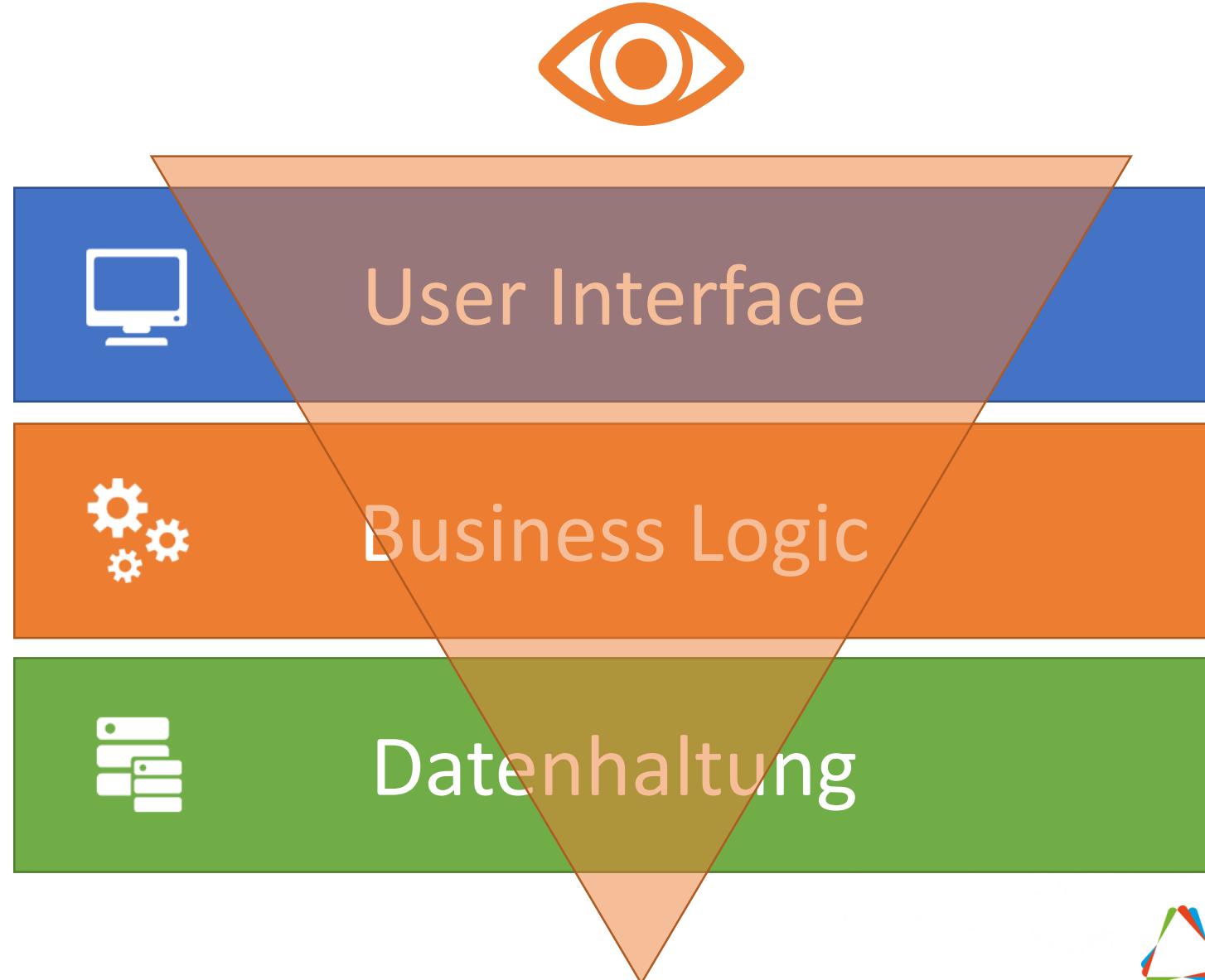
Teststrategie



Szenarienauswahl

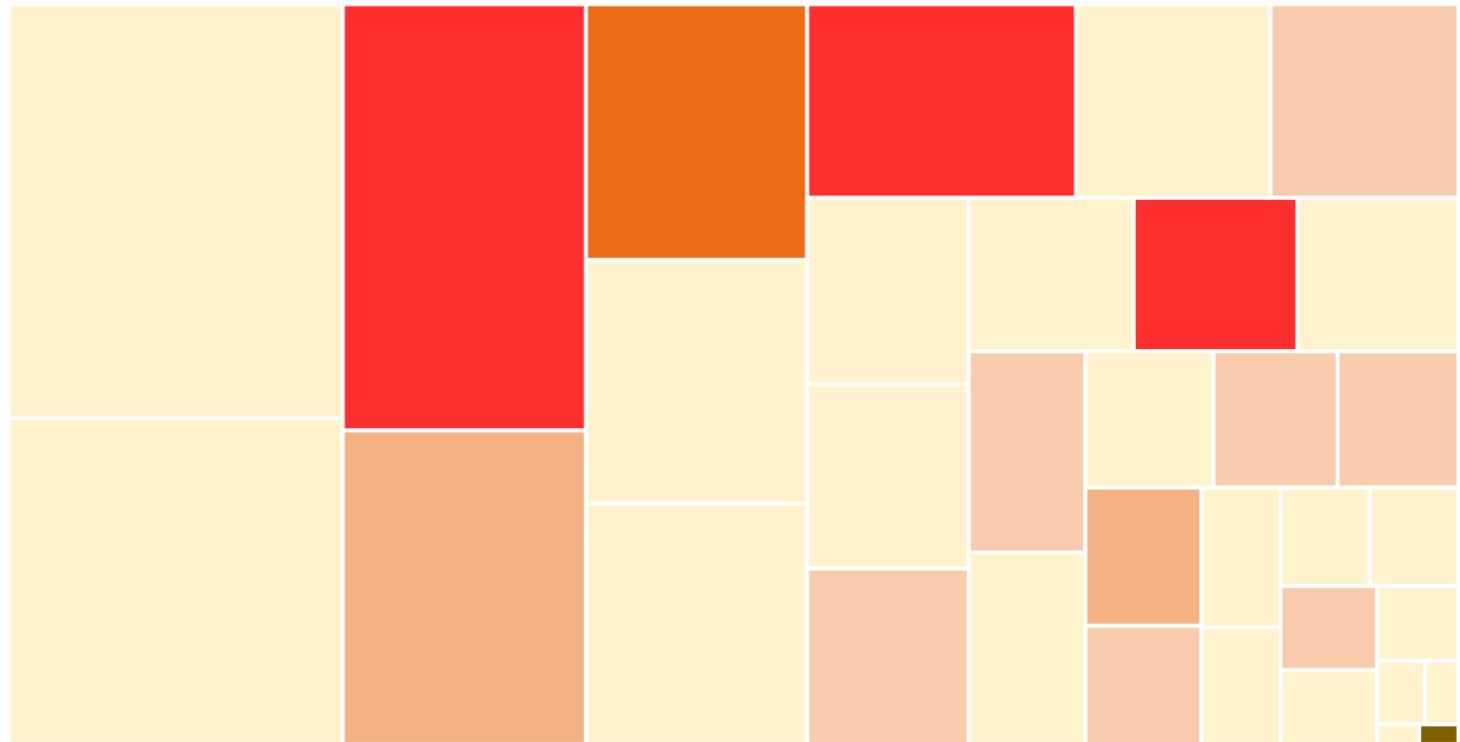


Szenarienauswahl



Hot Spot Analyse

Komplexität (CC) & Änderungshäufigkeit



Anzahl der Änderungen
von Januar bis März 2018

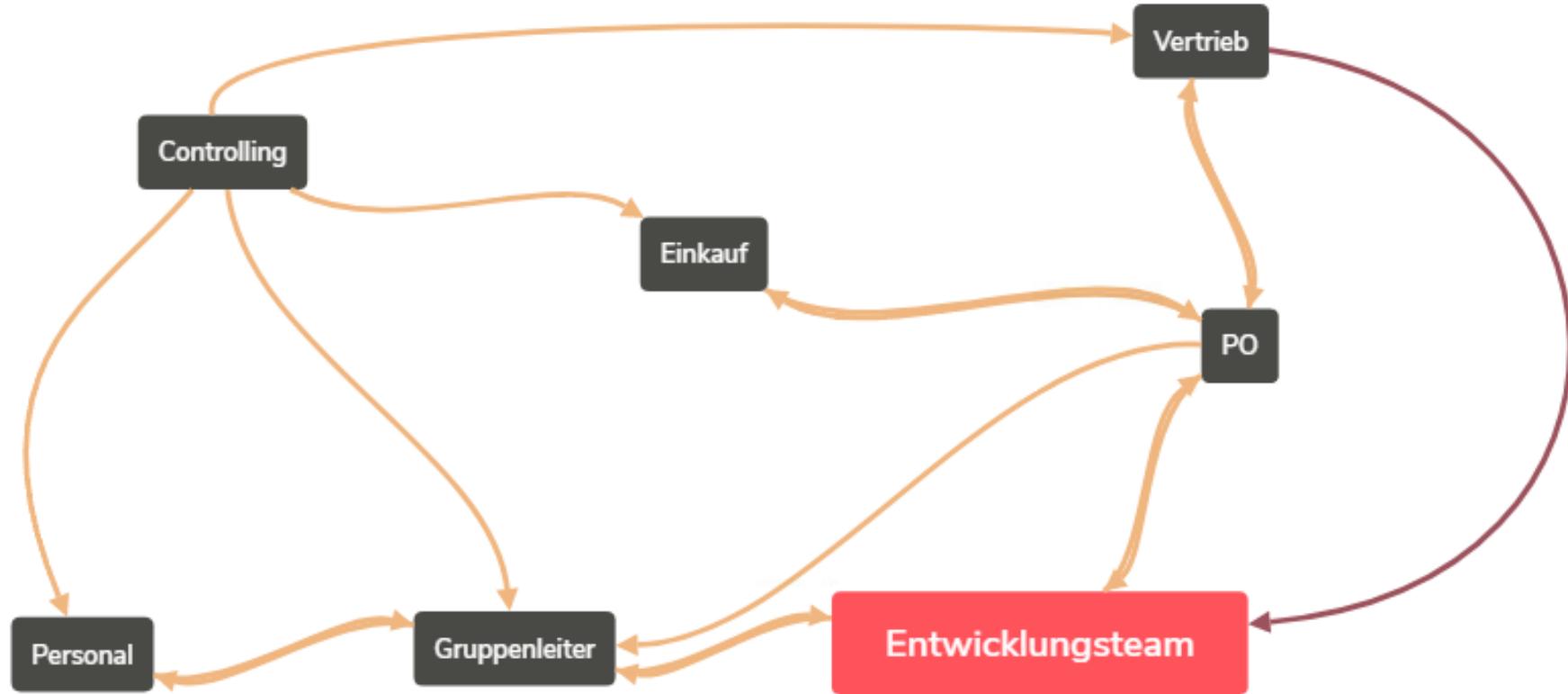
- 0 - 50
- 50 - 100
- 100 - 150
- 150 - 200
- 200 - 250
- > 250
- Sonstiges



Saxonia Systems
So geht Software.

KOLLABORATION

Stakeholder Map

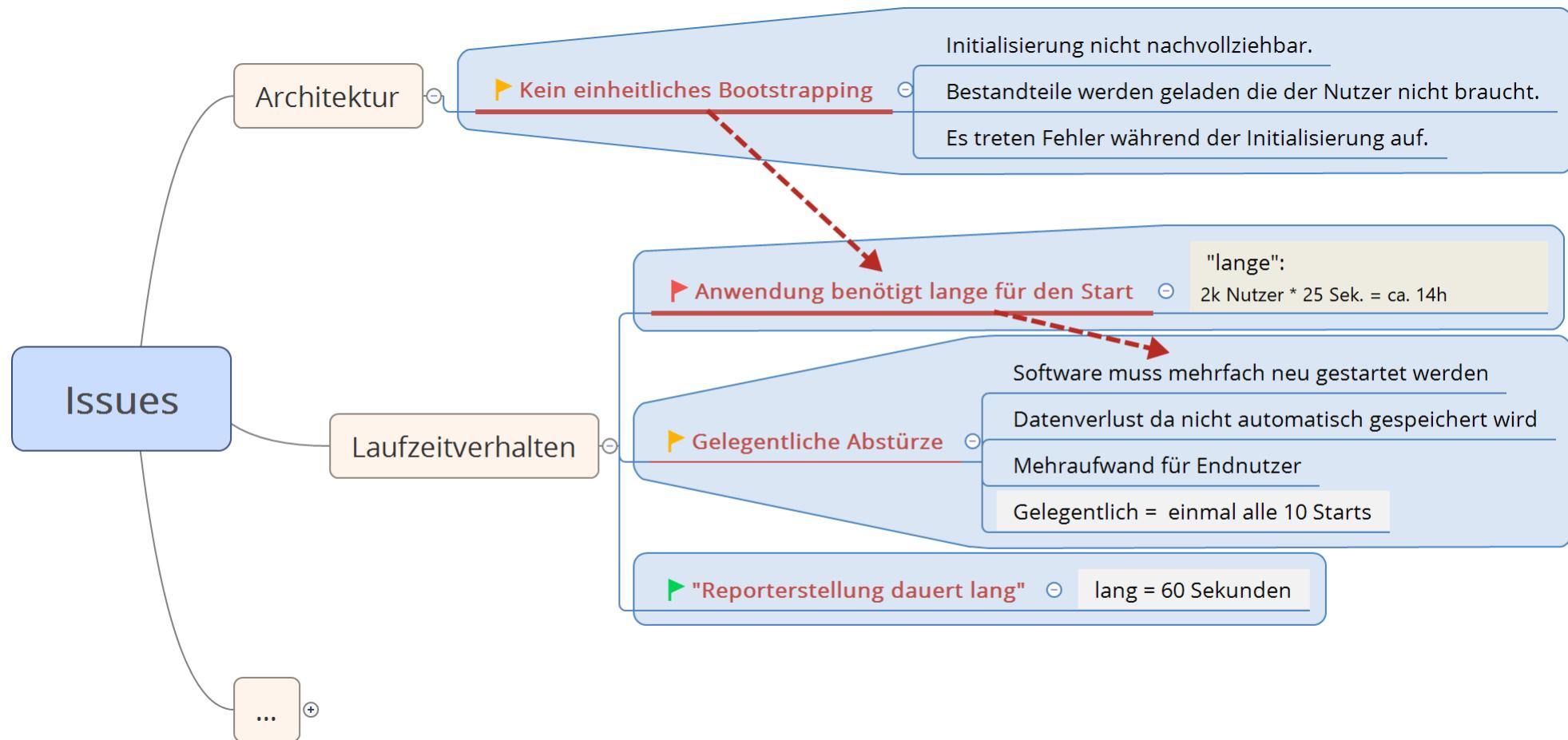


Issue Backlog

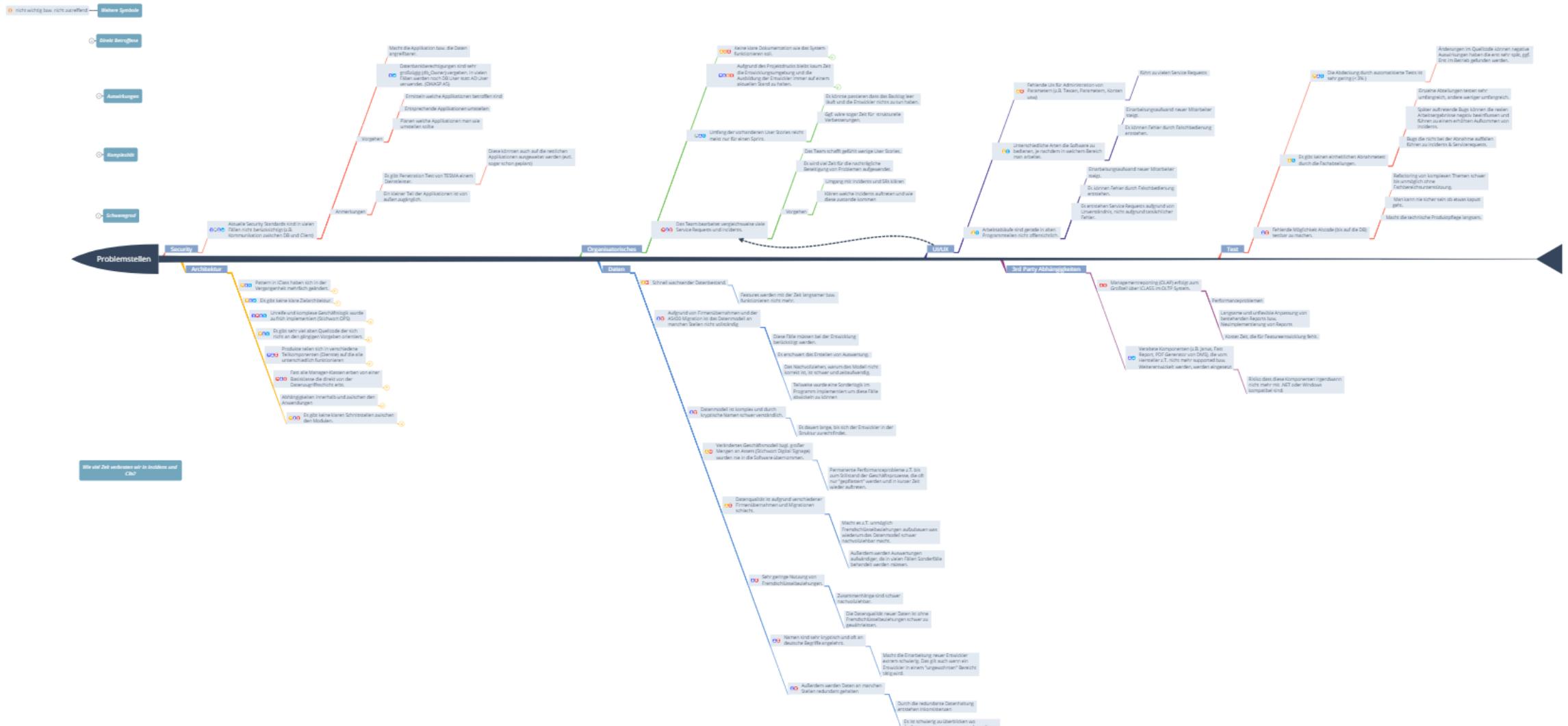
Problem	Auswirkungen	Betroffene Stakeholder
Keine Zielarchitektur vorhanden	Änderungen werden unstrukturiert vorgenommen. Unterschiedliche Muster werden eingesetzt. Es besteht eine beschleunigte Architekturerrosion.	Entwicklungsteams werden in ihrer direkten Arbeit behindert. Fachabteilungen merken, dass ähnliche Vorgänge sehr unterschiedliche Arbeitsabläufe aufweisen.
Geringe Testabdeckung	Sehr hoher Testaufwand.	IT Abteilung bindet Ressourcen die in der Entwicklung gebraucht werden. Fachabteilungen müssen Personal für den Test abstellen.
	Viele Fehler schlagen bis in die Produktivumgebungen durch.	Entwicklungsteams brauchen lange um Fehler zu beheben und verlieren Leistungsfähigkeit. Fachabteilungen werden in ihrer Arbeit behindert.
Entwickler verbringen viel Zeit mit Bugfixing.	Es besteht kaum strukturierte Weiterentwicklung weshalb Anforderungen als Service Requests formuliert werden.	Fachabteilungen verlieren Zeit, da sie für Kernaufgaben die IT konsultieren müssen. Entwicklungsteams werden mit zusätzlichen SRs blockiert.



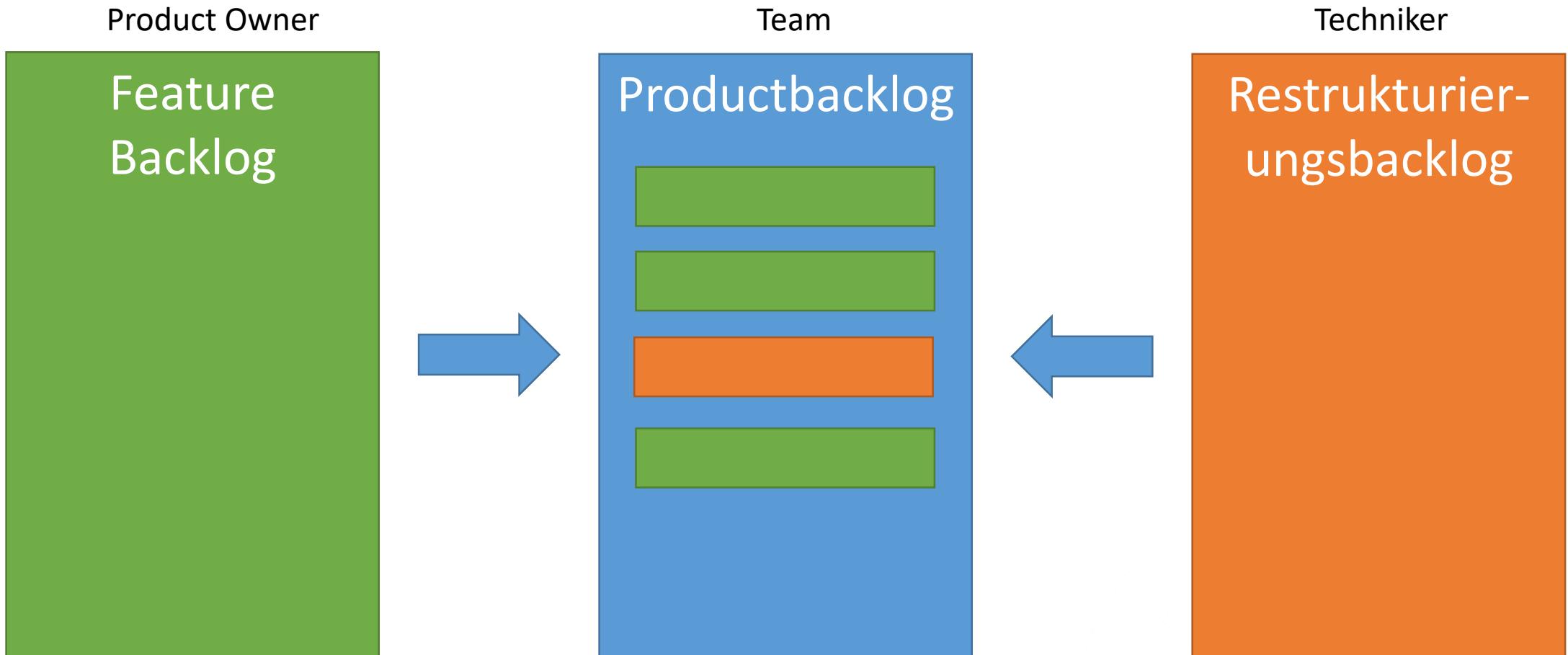
Issue Map



Issue Map

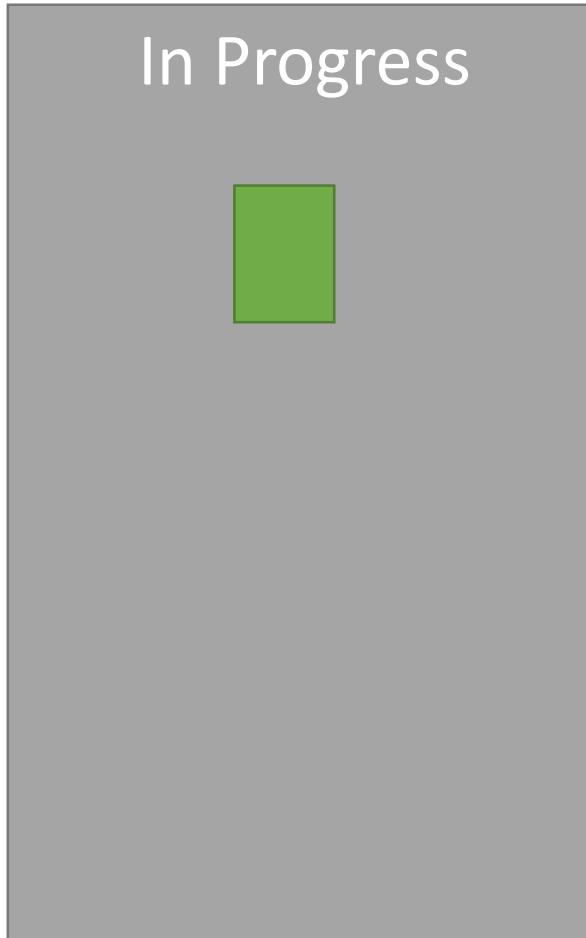
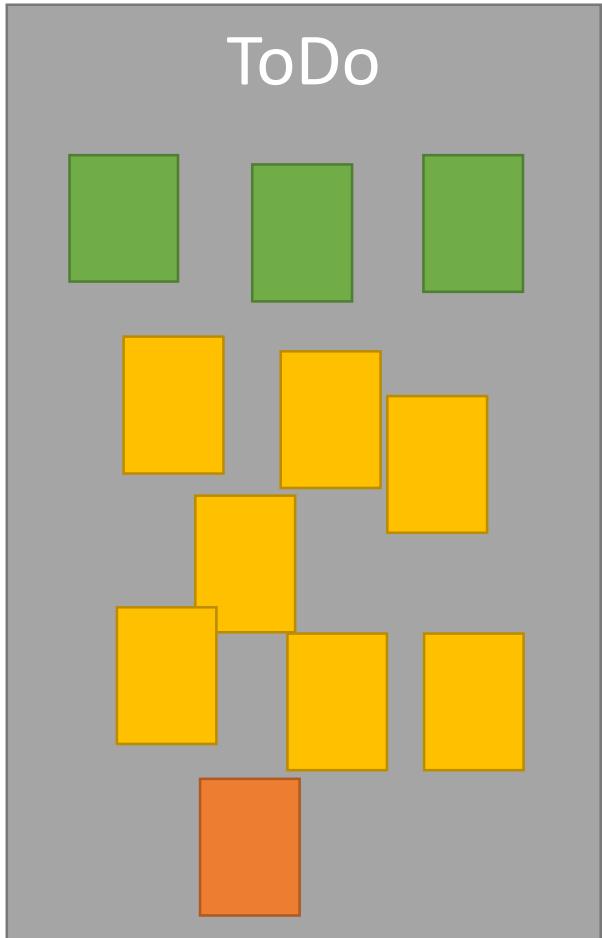


Backlogs

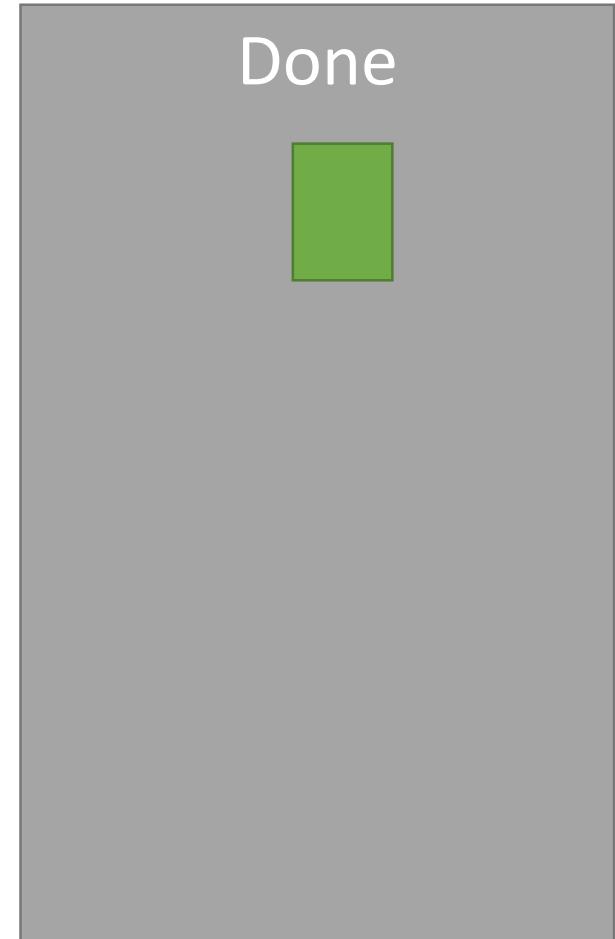
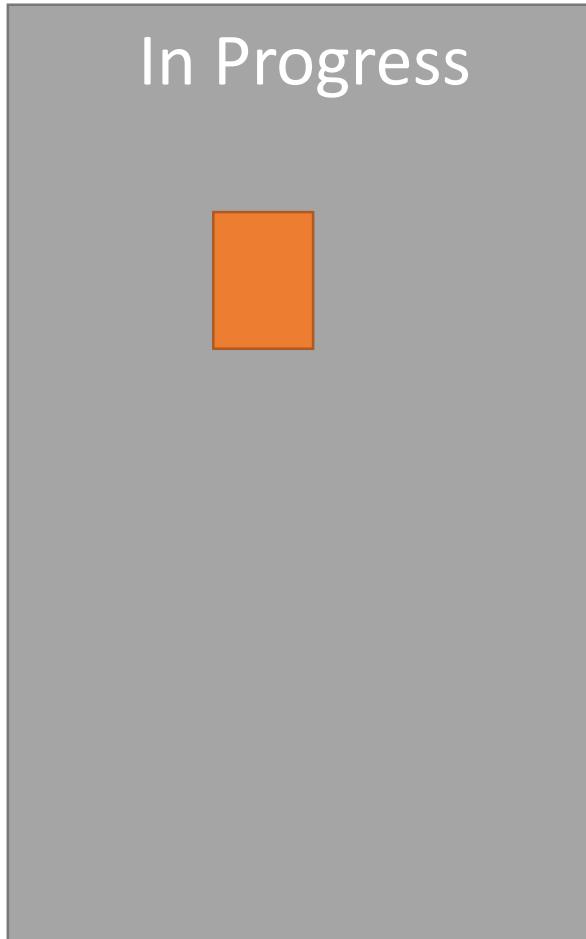
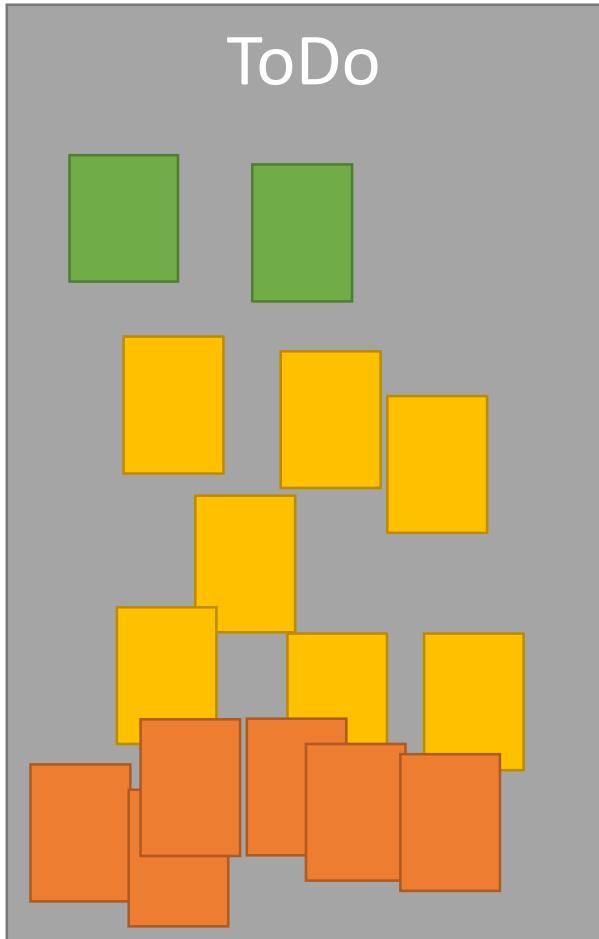


PRIORISIERUNG

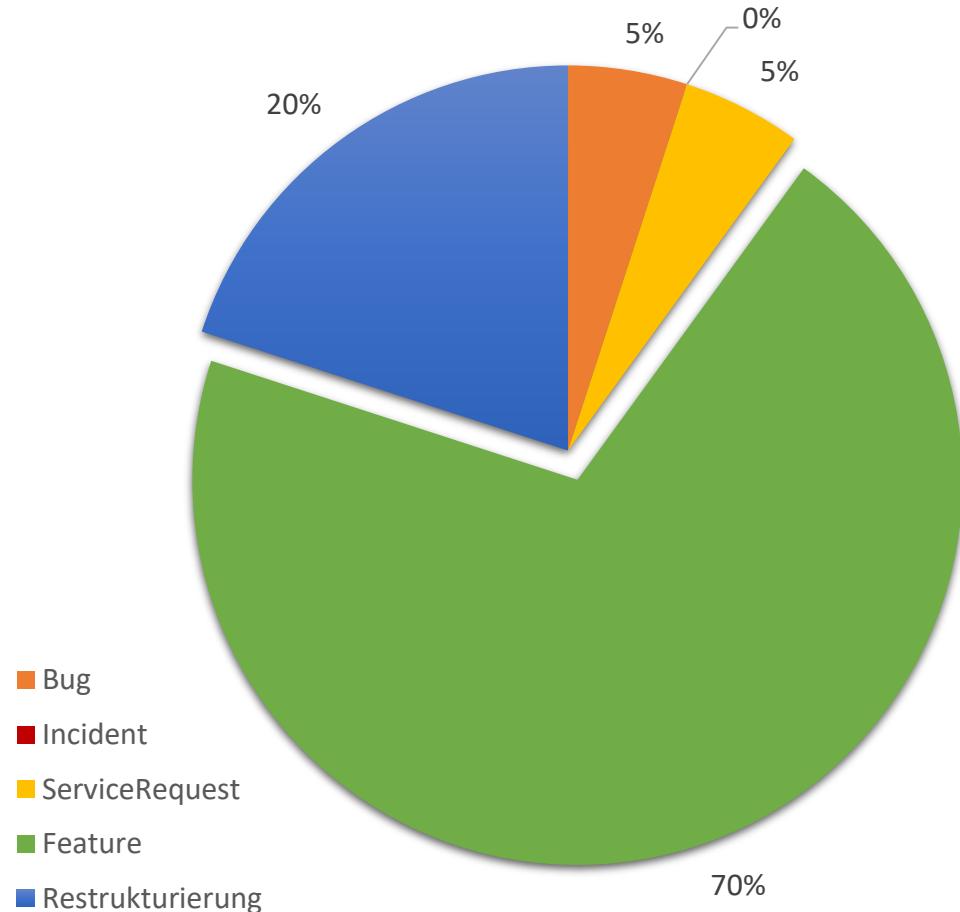
Mein Scrum ist kaputt...



Mein Scrum ist kaputt...



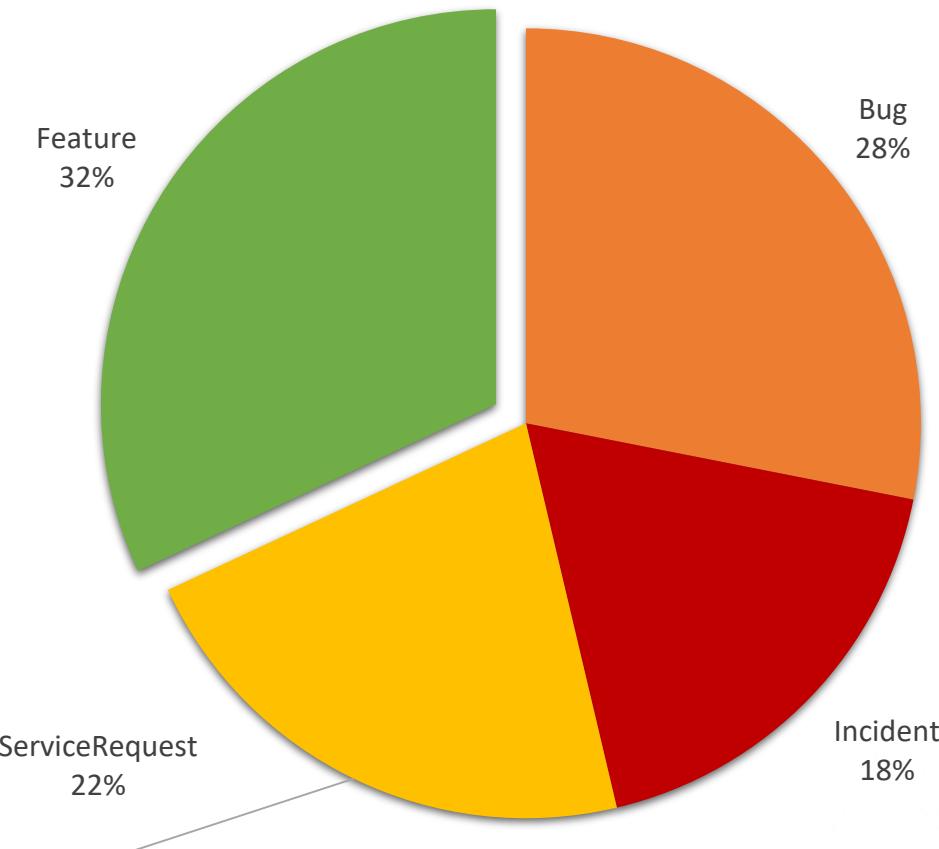
Workitem Analyse



- **Features**
wertsteigernden Maßnahmen
- **Service Requests**
Hilfen für die Endnutzer.
- **Incidents**
Probleme mit der Software, die in kürzester Zeit gelöst werden müssen.
- **Bugs**
Fehlverhalten gegenüber ursprünglich spezifiziertem Verhalten.
- **Restrukturierungen**
werterhaltende Maßnahmen die sich nicht in zusätzlicher Funktionalität wiederspiegeln.

Workitem Analyse

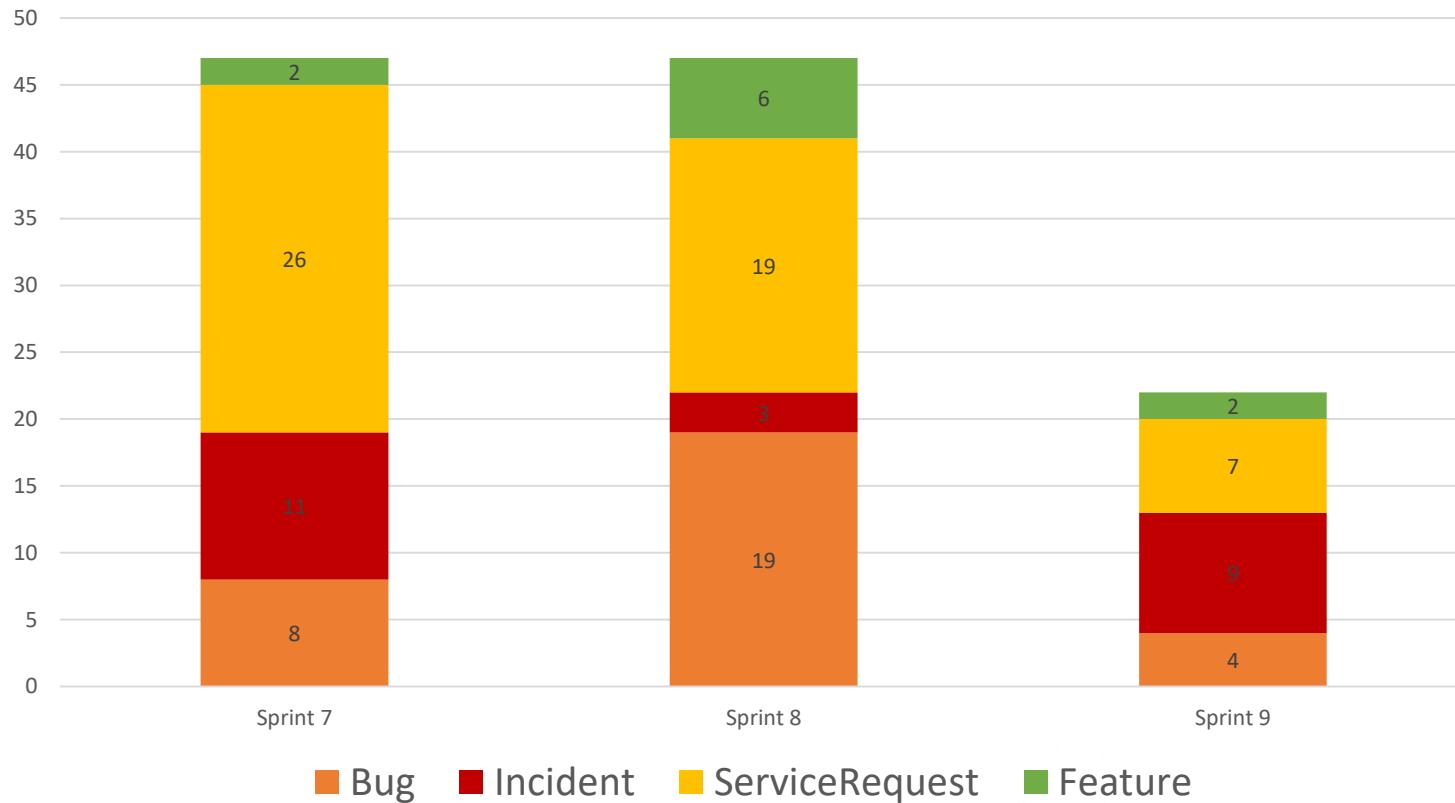
Zeitliche Verteilung



Saxonia Systems
So geht Software.

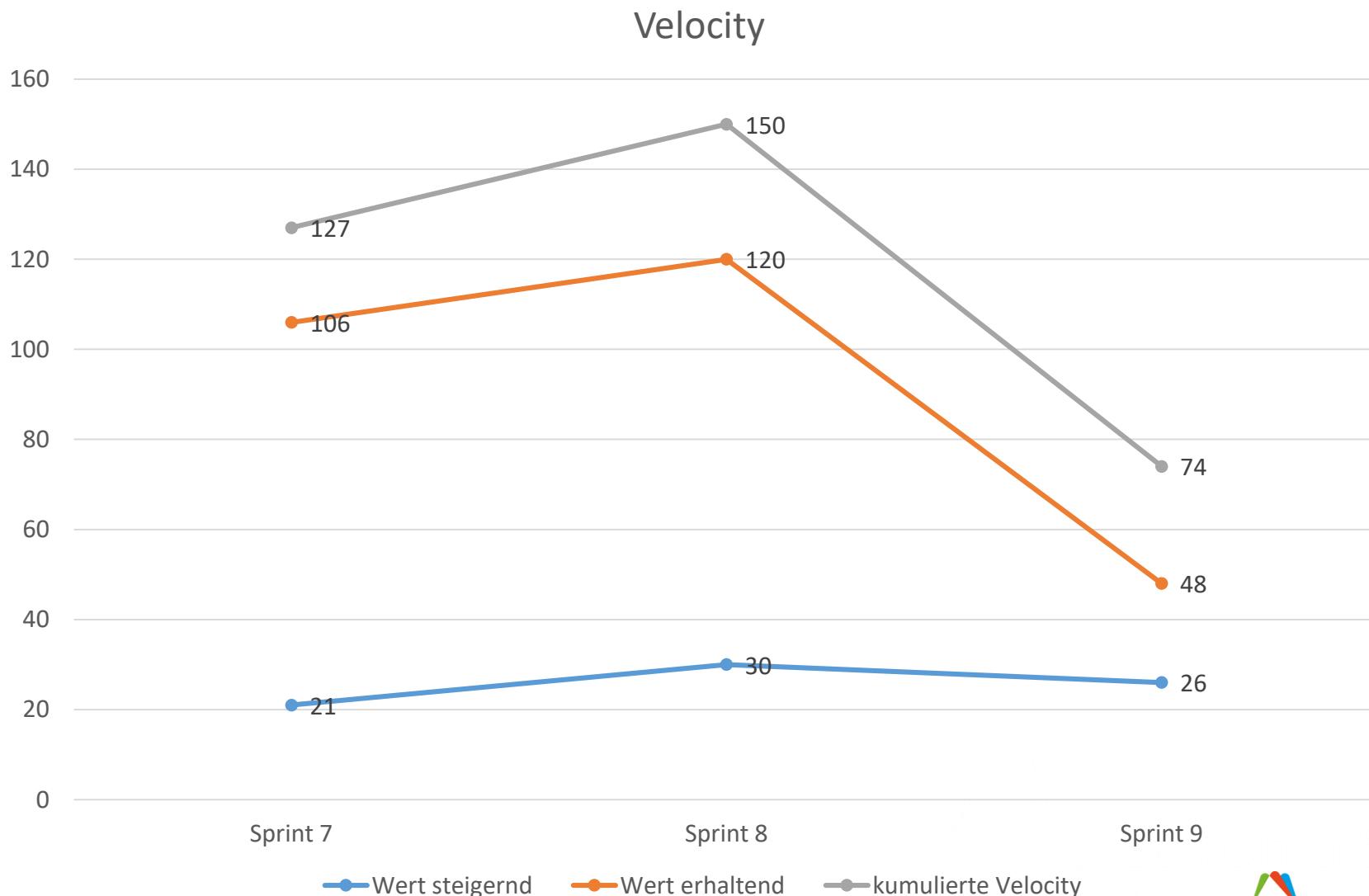
Workitem Analyse

Burnup Chart



Saxonia Systems
So geht Software.

Workitem Analyse



Saxonia Systems
So geht Software.

Sprintbacklog

Incidents

Features

Service Requests

Restrukturierung



Saxonia Systems
So geht Software.

Sprintbacklog zu Beginn des Sprints



Saxonia Systems
So geht Software.

Sprintbacklog gegen Ende des Sprints

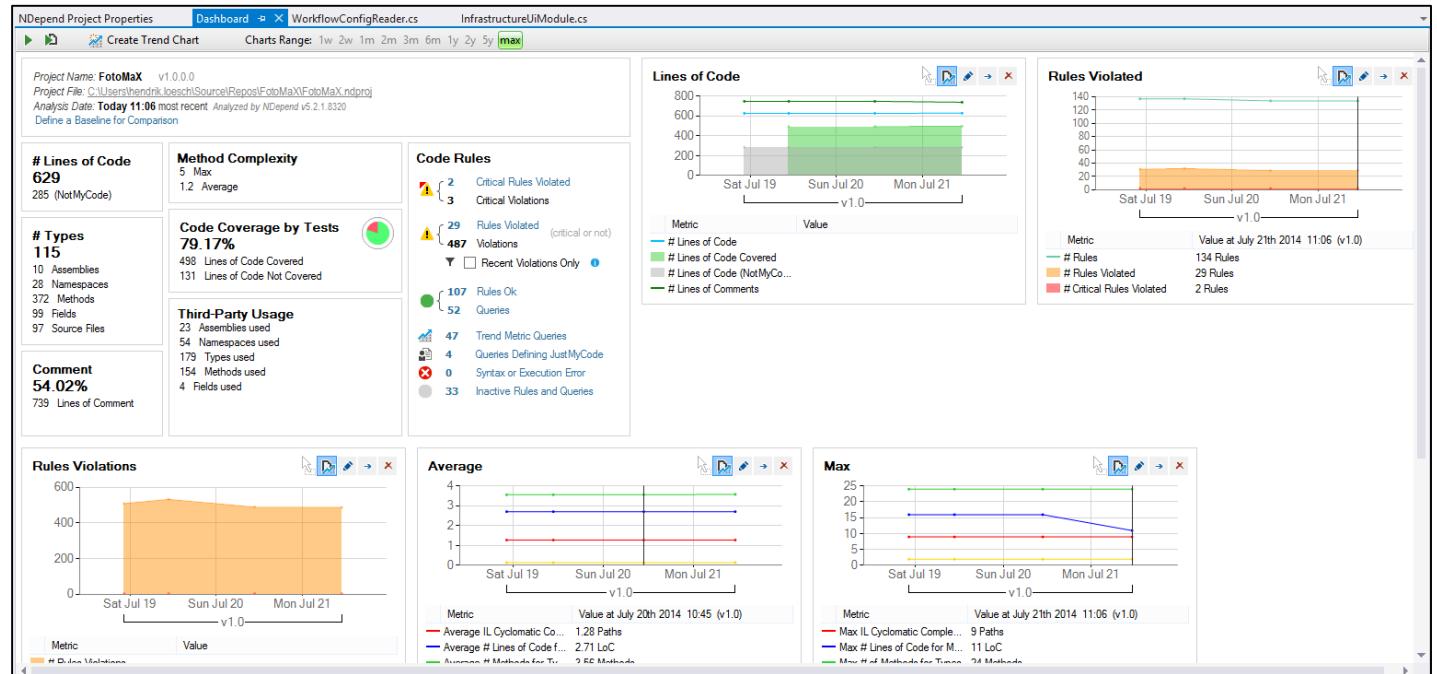


Saxonia Systems
So geht Software.

WERKZEUGE

XDepend

- Verfügbar für:
 - C#
 - VB.Net
 - Java
 - C++
- 14 Tage Testversion
- Einbindbar in den Build
- Verfügt über Html-Reports

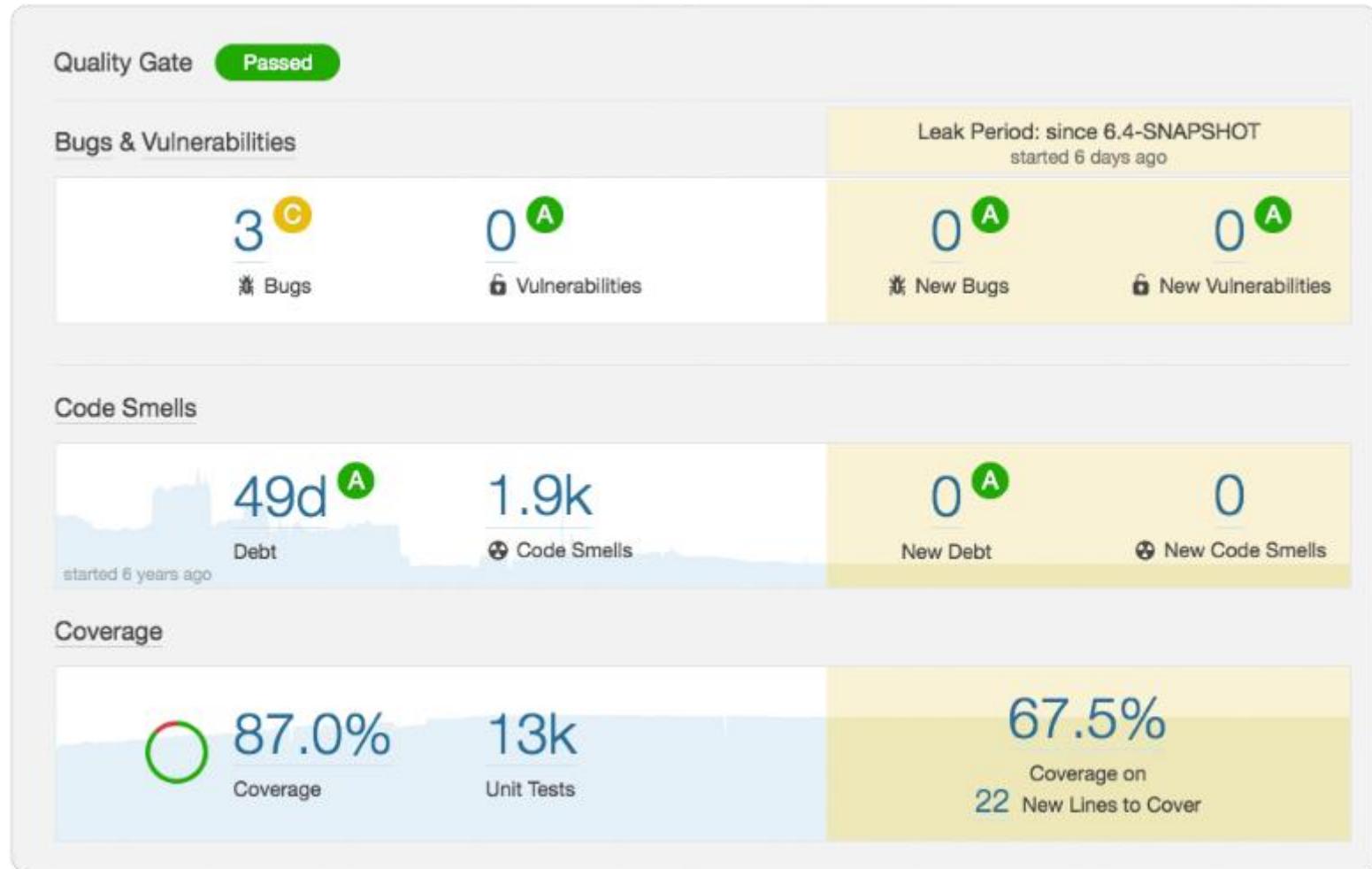


<https://www.ndepend.com/>



Saxonia Systems
So geht Software.

SonarQube

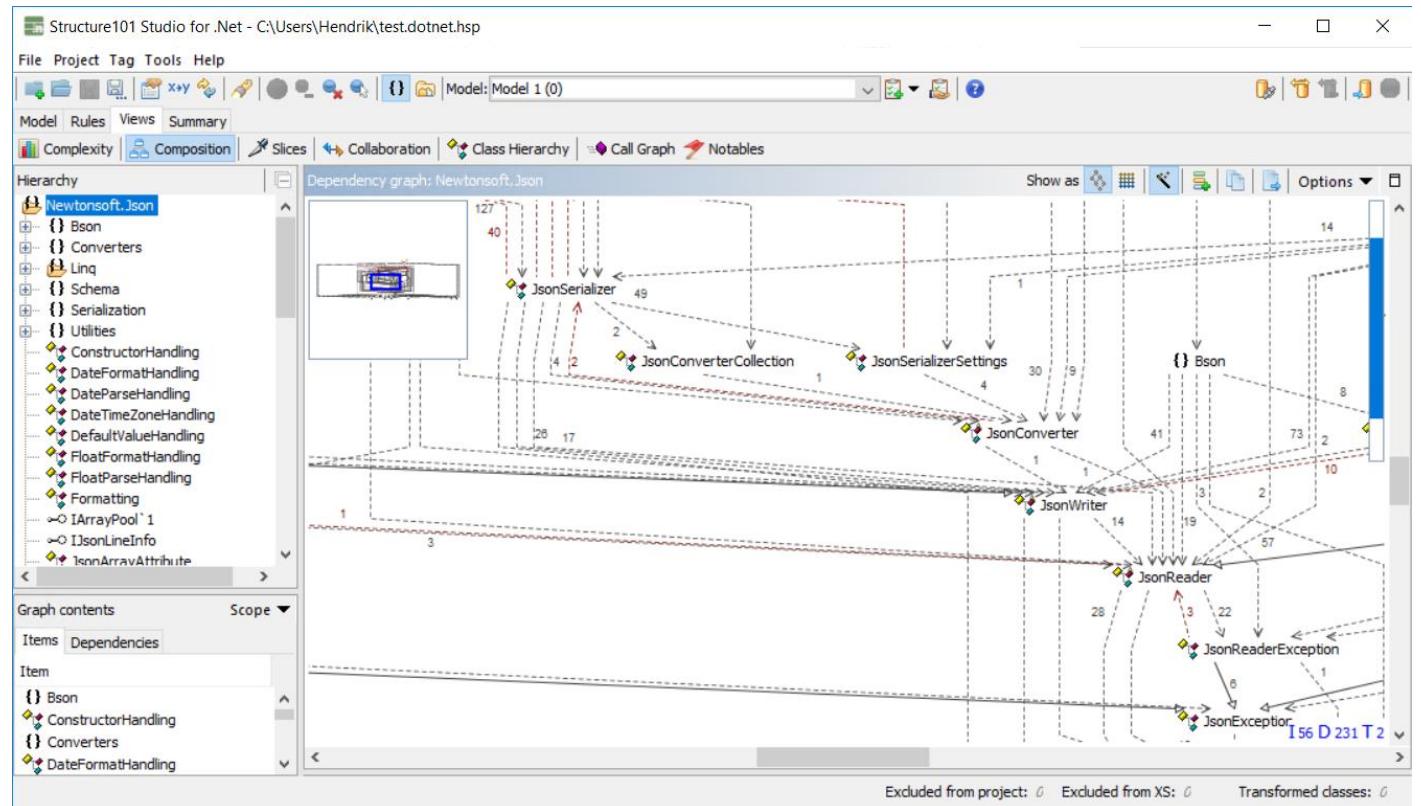


- Analyse der Codequalität während der Entwicklung
- Gibt Hinweise auf Problemstellen im Code.

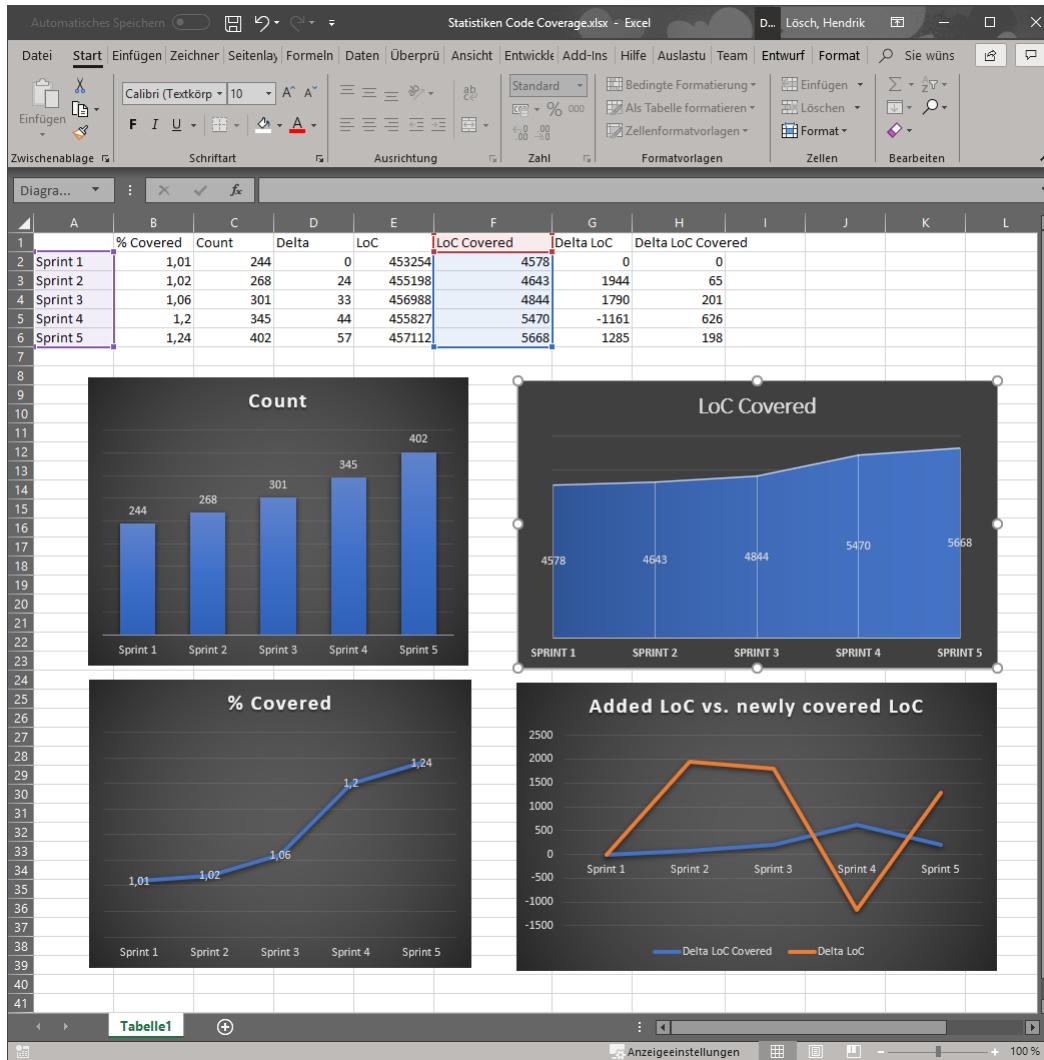
<https://www.sonarqube.org/>

Structure 101 Studio

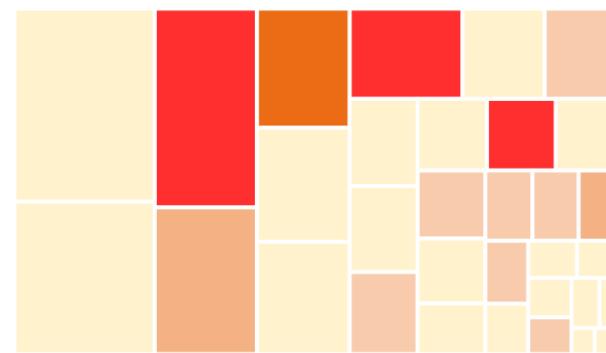
- Verfügbar für:
 - Java
 - C#
 - C++
 - Standalone mit VS Plugins
 - Sehr gut für Architekturanalyse



Excel

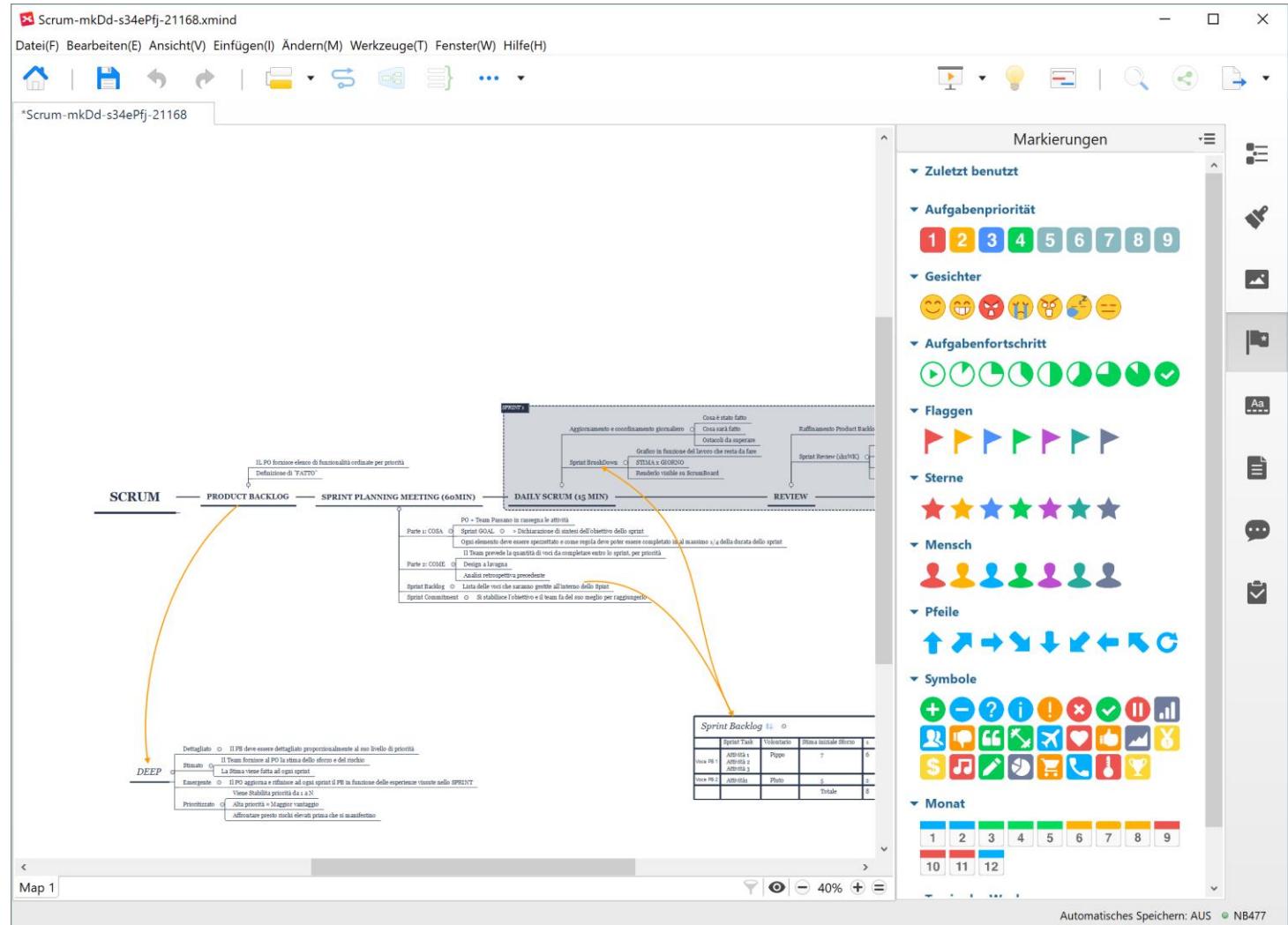


- Besser als sein Ruf...
- Hilfreich um:
 - Statistiken zu erfassen,
 - Daten auszuwerten,
 - Zusammenhänge visualisieren.
- Selten genutzt aber sehr praktisch:
 - Pivot Table
 - Tree Map



XMind

- Mind Mapping Tool für verschiedene Plattformen.
- Viele Extrafeatures und Darstellungsmethoden.
- Version 8 ist kostenlos.



Saxonia Systems
So geht Software.

ZUSAMMENFASSUNG



Saxonia Systems
So geht Software.

Euphorie beginnt ein Vorhaben



Saxonia Systems
So geht Software.

Euphorie beginnt ein Vorhaben,
Disziplin beendet es.

Kleine Schritte aber stetig.

Erst analysieren,
dann planen.



Saxonia Systems
So geht Software.

Erst analysieren,
dann ~~planen~~^{eskalieren}.

Der Sprecher



Hendrik Lösch

Senior Consultant & Coach

Hendrik.Loesch@saxsys.de

@HerrLoesch

Hendrik-Loesch.de



ReSharper lernen

Effektiver programmieren mit der Erweiterung zu Visual Studio

by Hendrik Lösch



WPF-Anwendungen mit MVVM und Prism

Modulare Architekturen verstehen und umsetzen

by Hendrik Lösch



Windows 8 Store Apps mit MVVM und Prism

XAML-Entwurfsmuster, Bootstrapping, Navigation, Messaging

by Hendrik Lösch



LINQ Grundkurs

Wichtige Spracheigenschaften von C#, Joins, Variablen und andere Operationen, erweiterte Techniken

by Hendrik Lösch

LinkedIn Learning



Grundlagen der Programmierung: Test Driven Development

Business-Anwendungen testgetrieben entwickeln

by Hendrik Lösch



Inversion of Control und Dependency Injection – Grundlagen

Prinzipien der modernen Software-Architektur ...

by Hendrik Lösch



C#: Test Driven Development

Grundlagen, Frameworks, best Practices

by Hendrik Lösch



Grundlagen der Programmierung: Codemetriken

Softwarequalität einschätzen, sicherstellen und ...

by Hendrik Lösch



Saxonia Systems
So geht Software.