

Von der statischen Analyse zur nachhaltigen Qualitätssicherung

Java Forum Stuttgart 2022
Dr. Tobias Röhm (@langelot)





Freelancer

TUM

CQSE



2004 - 2010

2010 - 2015

2015 - Heute



Praxis

Software **Audits**

Kontinuierliches **Quality-**
und **Test-Control**

 **Teamscale**

CQSE GmbH



Forschung

16+ **PhDs** in Software-
engineering

Eigene **Forschung**

Enger Kontakt zu
Universitäten

Kiuwan
Sotoarc
Infer
SofCheck Inspector
ObjectWeb ASM
IBM Rational AppScan
Checkstyle
Moose
FindBugs
Teamscale*
Xanitizer
AgileJ StructureViews
Imagix 4D
Clang
Codyze
Yasca
DMS Software Reengineering Toolkit
SonarQube
Coverity SAVE
PVS-Studio
SQuORE
Codyze
CAST Application Intelligence Platform
Gamma
Klocwork Insight
Visual Studio Team System
PMD
ConQAT*
ThreadSafe
Parasoft
App-Ray
Protecode
Simian - Similarity Analyser
BugScout
Oversecured
MALPAS Software Static Analysis Toolset
Kalistick
RIPS
Copy/Paste Detector (CPD)
JDepend
Cigital SecureAssist
Hammurapi
Squale
Lattix
LDRA Testbed
Feram
ResourceMiner
Soot
Fluctuat
Axivion Bauhaus Suite
GrammaTech CodeSonar
Structure101
Polyspace

* von CQSE entwickelt, Quelle: wikipedia.de



Klone

java.lang

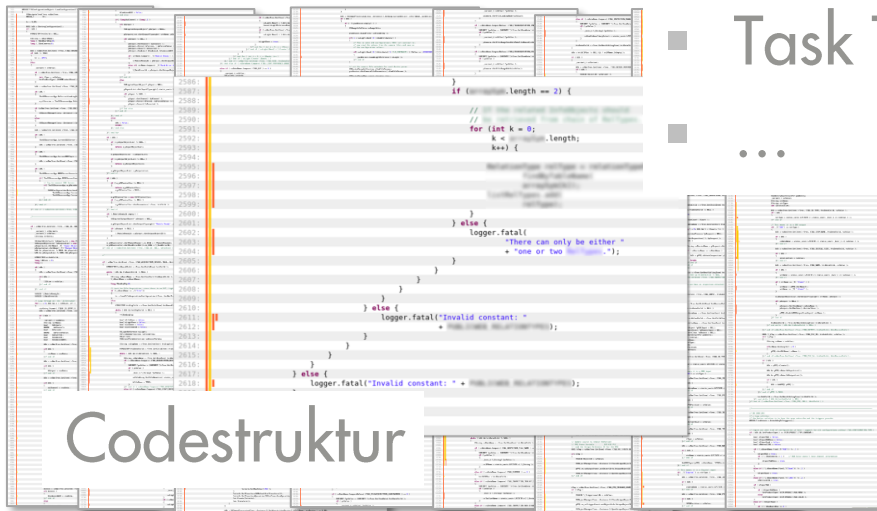
Bug Patterns

Class NullPointerException

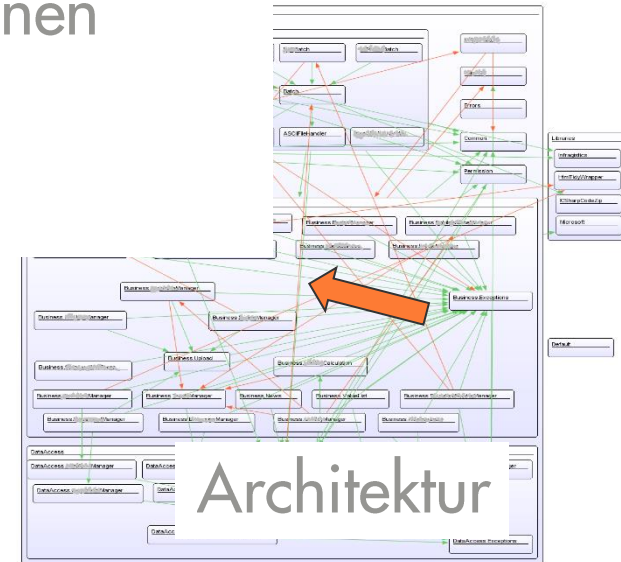
java.lang.Object
java.lang.Throwable

RuntimeException
lang.NullPointerException

- Kommentarvollständigkeit
- Richtlinienverletzungen
- Auskommentierter Code
- Namenskonventionen
- Task Tags
- ...



Codestruktur



Architektur

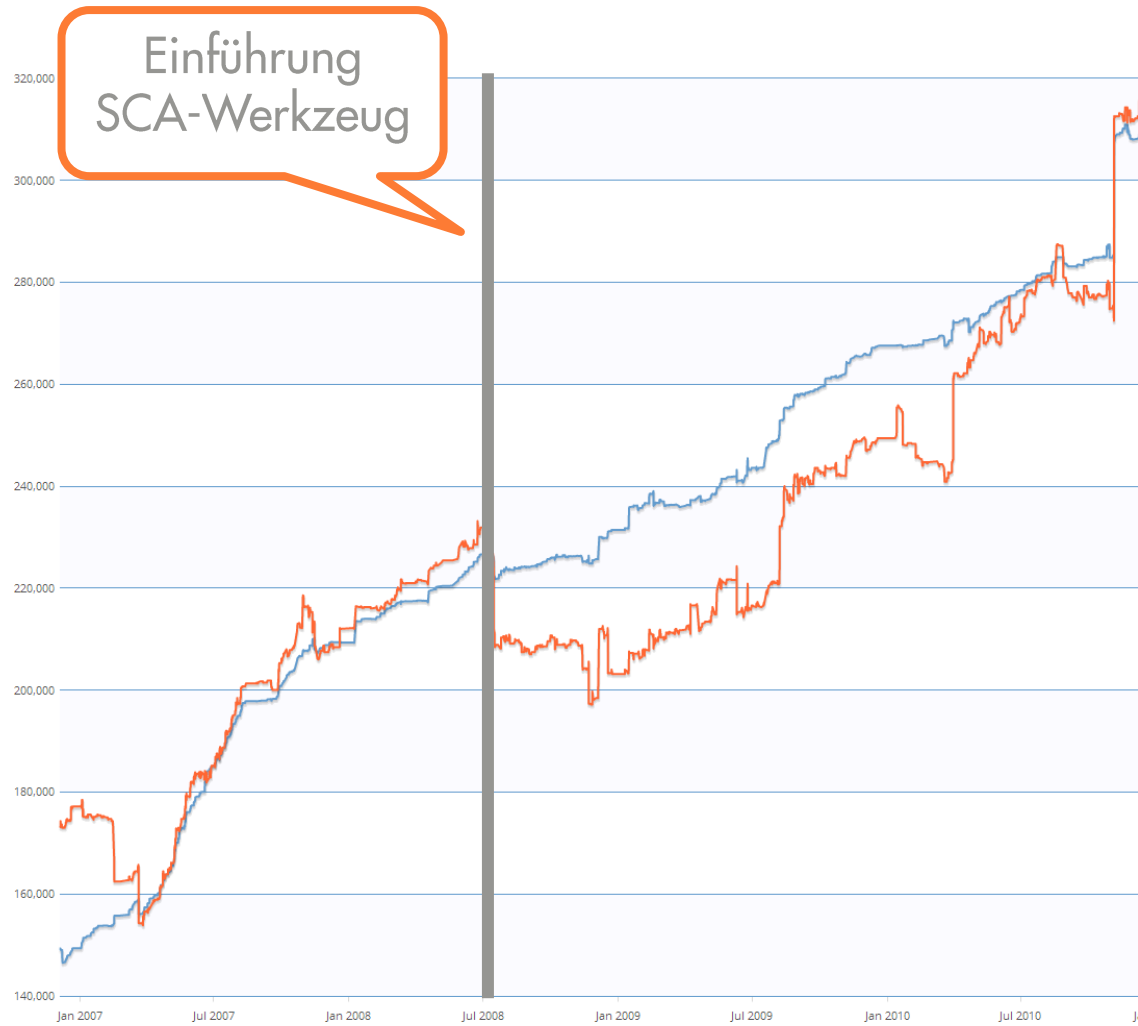
→ Statische Analyse schafft Transparenz über technische Schulden und Qualitätsdefizite



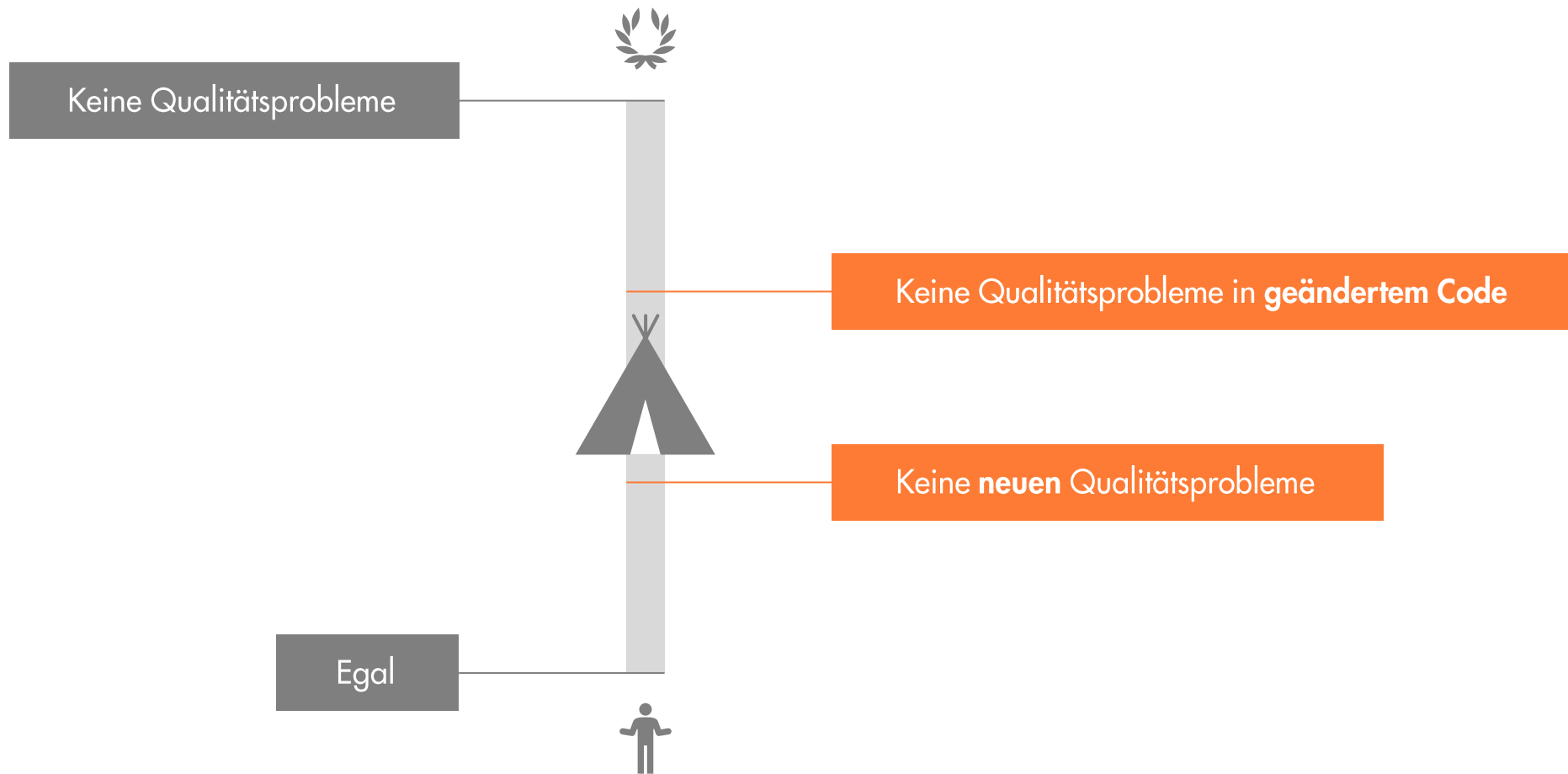


Wie können wir uns verbessern?

**»Es geht nicht nur um die Funktion von Analysetools,
sondern darum, wie ich mit den Ergebnissen umgehe!«
C. Finkbeiner, Softwarearchitekt, SEW-Eurodrive**



→ Die reine Einführung eines statischen Analysewerkzeugs hat häufig nur einen kurzfristigen Effekt.



→ Erfolgsfaktor »Realistisches Qualitätsziel«



→ Erfolgsfaktor »Keine Qualitätspolizei«

Der Analysescope beinhaltet ...

Anwendungscode

Testcode

... und exkludiert

~~Generierten Code~~

~~3rd-Party-Code~~

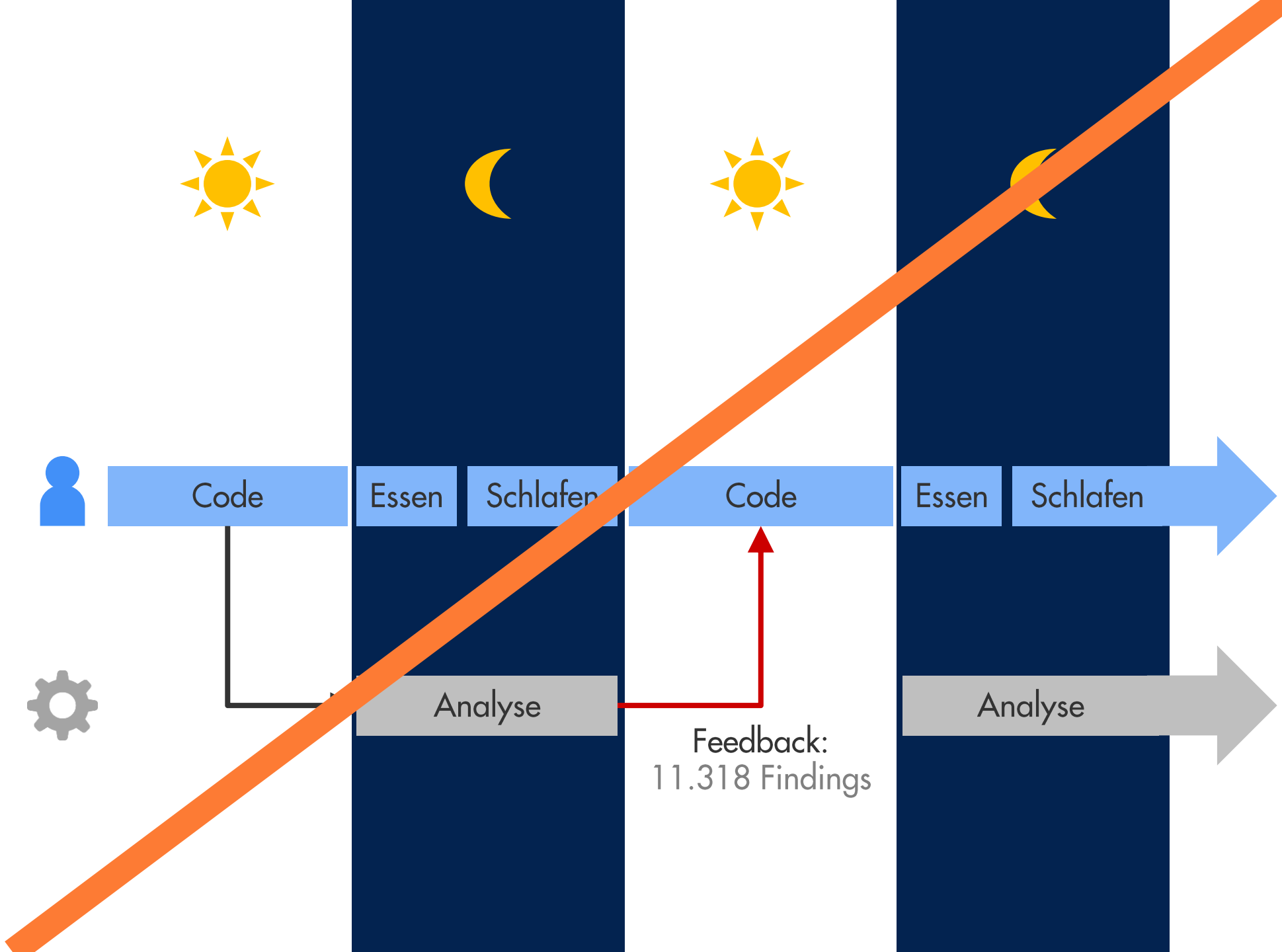
~~Experimentellen Code~~

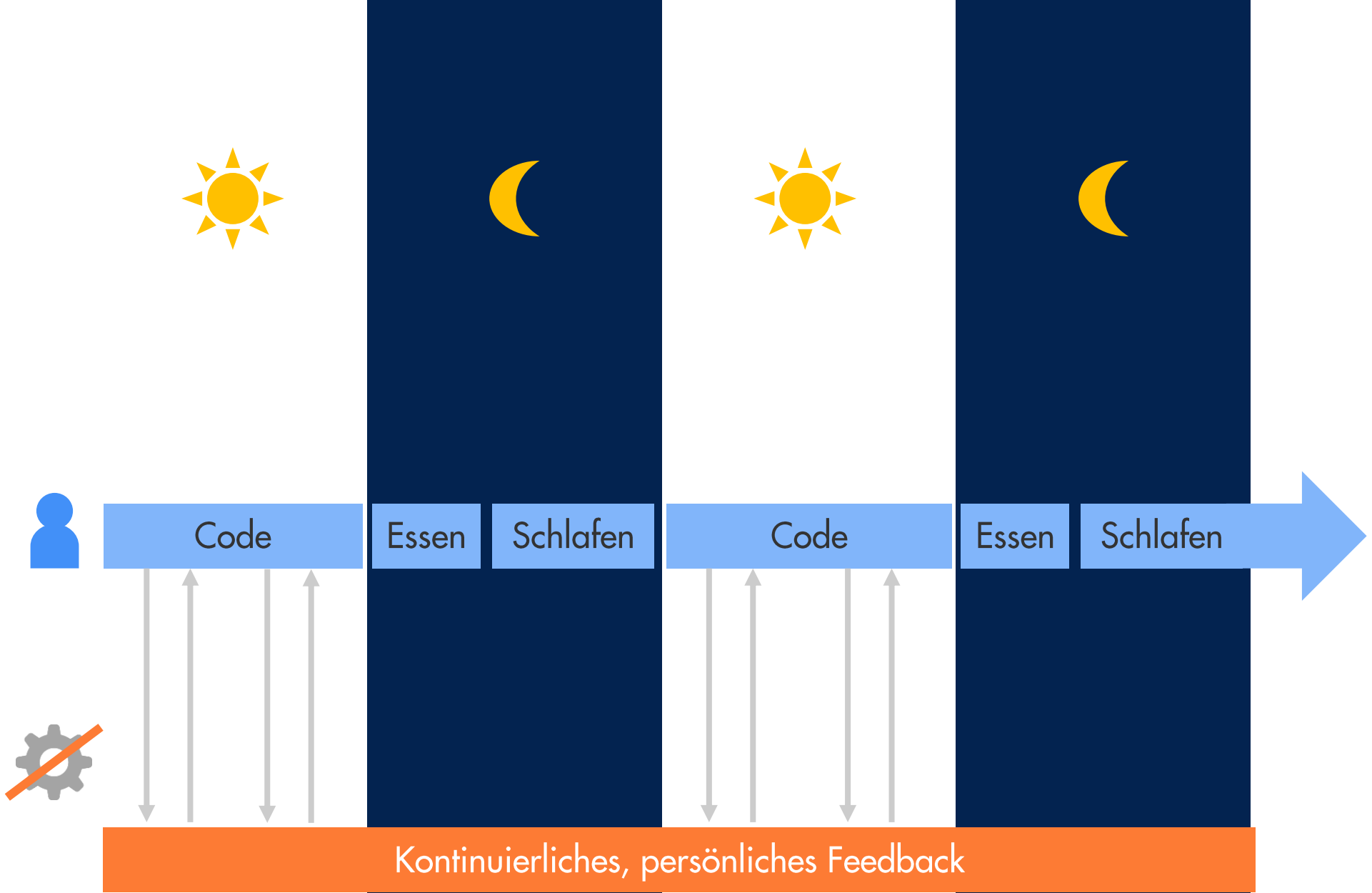
~~Ungenutzten Code~~

→ Erfolgsfaktor »Individuelle Analysekonfiguration«











Improvements

by Dennis Pagano as revision c41014dc in precommit-client

Files: 4 changed

Aug 22 2018
22:53



First version, for real

by Dennis Pagano as revision 567347b9 in precommit-client

Files: 5 changed

Findings: 🔴 2 🔵 2 🟢 4

Aug 22 2018
20:22



First version of precommit client

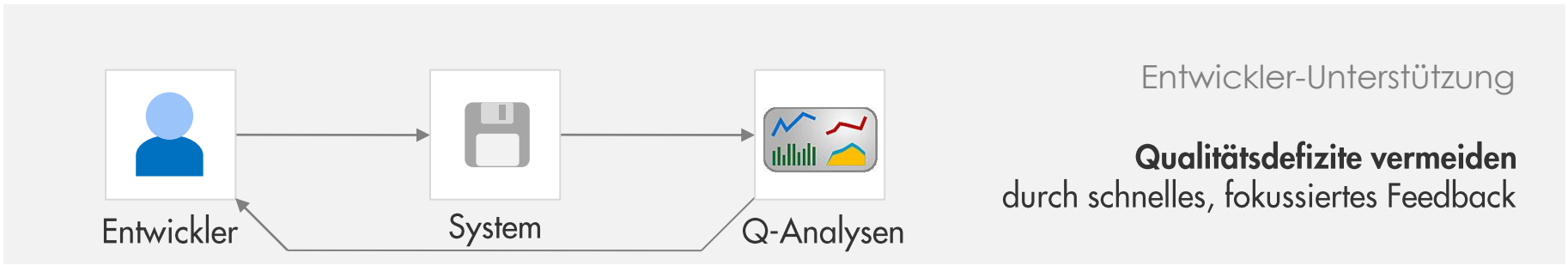
by Dennis Pagano as revision ea87f8e5 in precommit-client

Files: 4 added

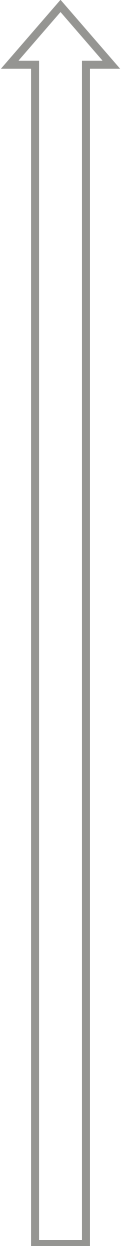
Findings: 🔴 5

Aug 22 2018
20:12

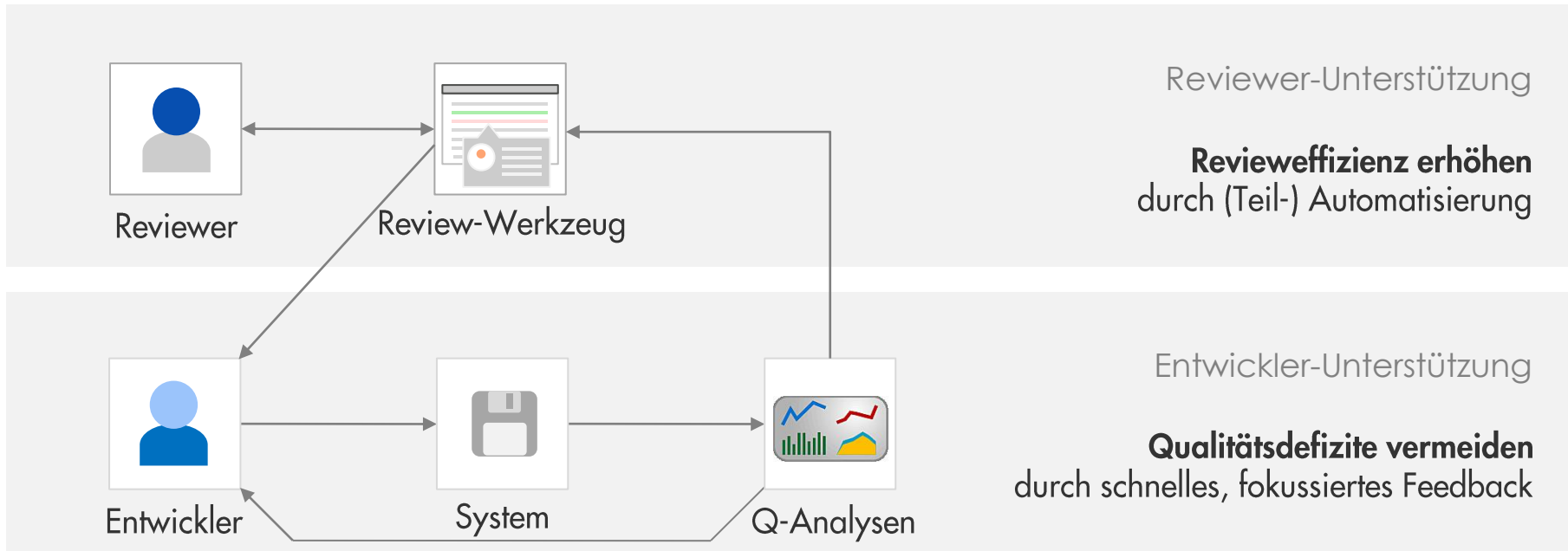
→ Erfolgsfaktor »Schnelles, änderungsfokussiertes Feedback«



Sekunden

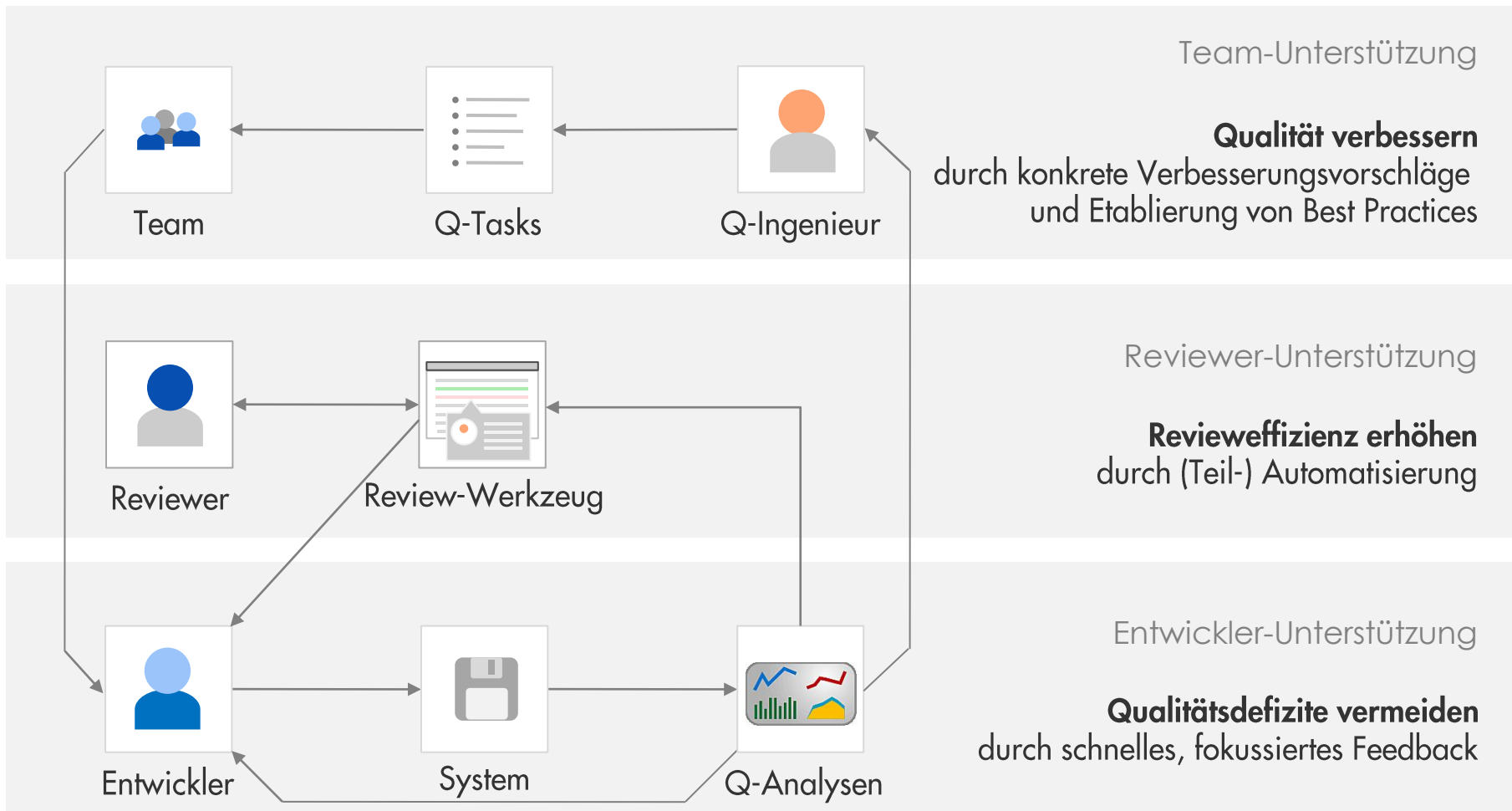


Zeit



The screenshot shows the Azure DevOps interface for a pull request. The breadcrumb navigation at the top reads 'cqse / Test / Repos / Pull requests / Test'. The pull request title is 'Fix for Serialization Bug' by user 'Khater', with a status of 'ACTIVE' and '0/3 resolved'. The description section contains the text 'findings: 3' highlighted with a red box, followed by 'Fixed the Serialization bug:' and a bulleted list of changes: 'Added @JsonProperty annotations' and 'Added @JsonCreator annotations to constructors'. The right-hand sidebar displays 'Status' with '3 new findings', 'Work Items' with 'No related work items', 'Reviewers' with 'No reviewers', and 'Labels' with an 'Add label' button.

→ Erfolgsfaktoren »Review(teil-)automatisierung« und »Workflow-Integration«



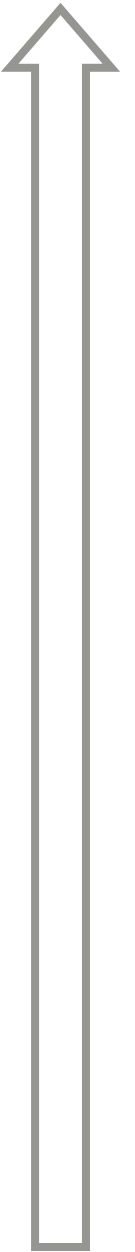
Monate

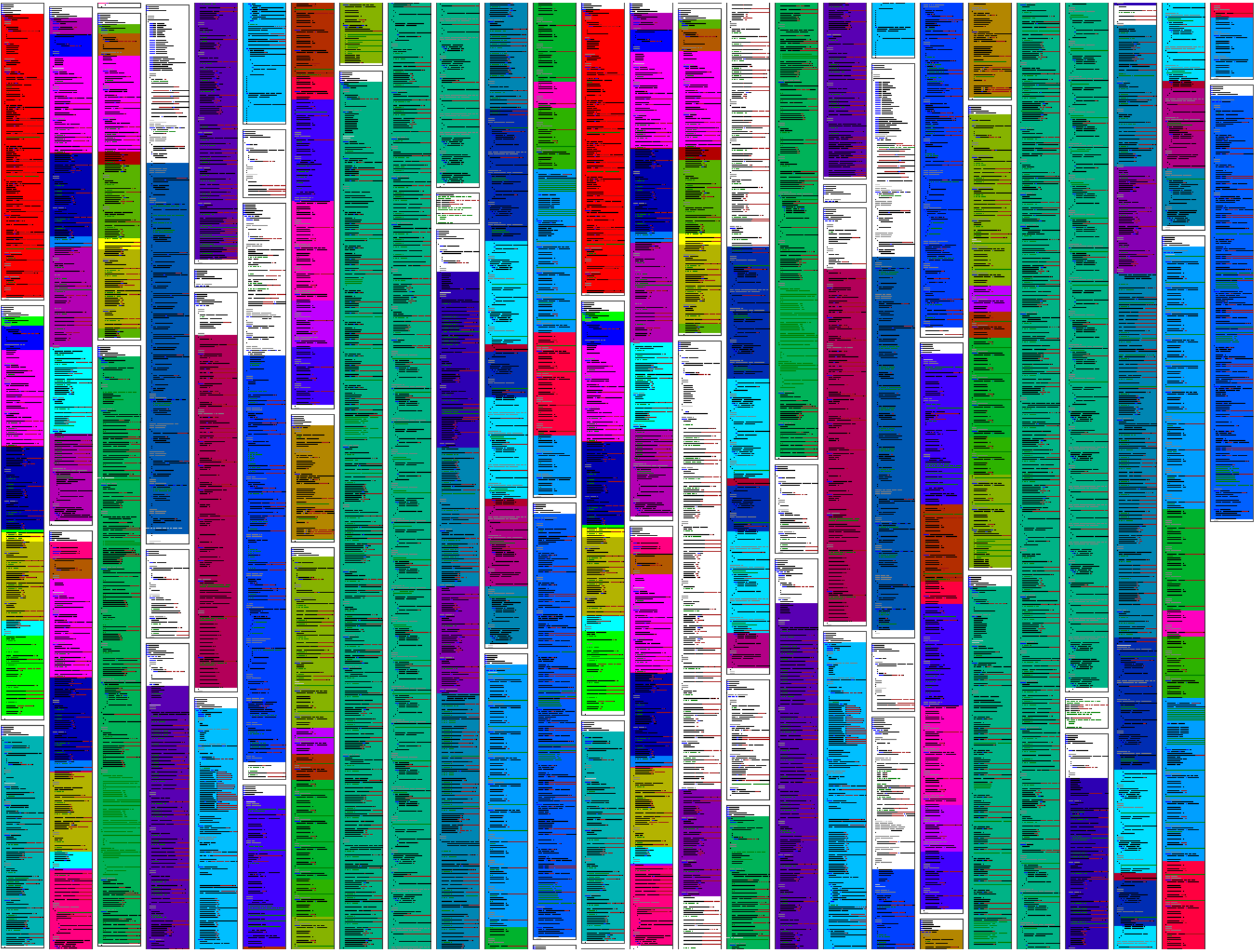
Wochen

Minuten

Sekunden

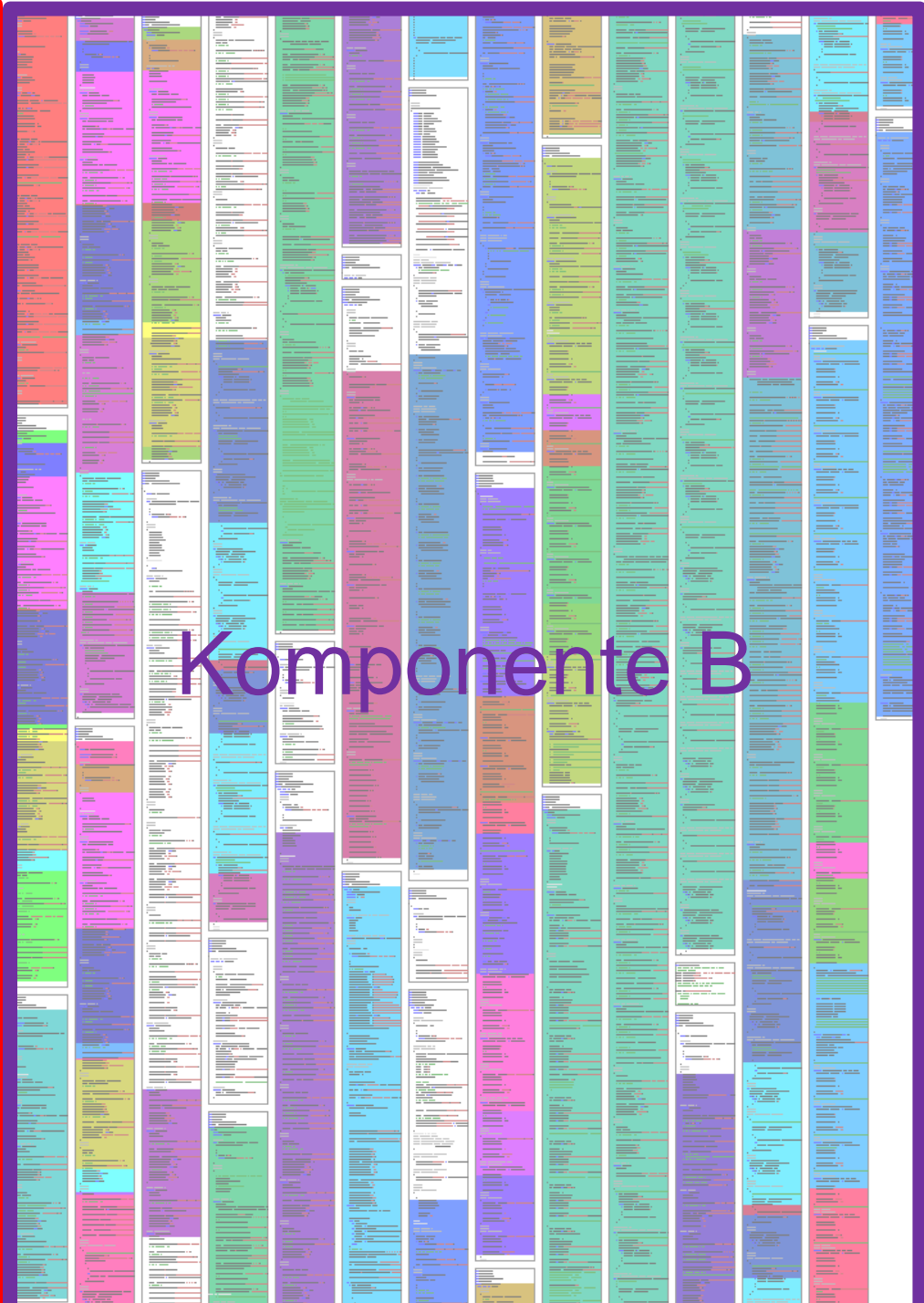
Zeit



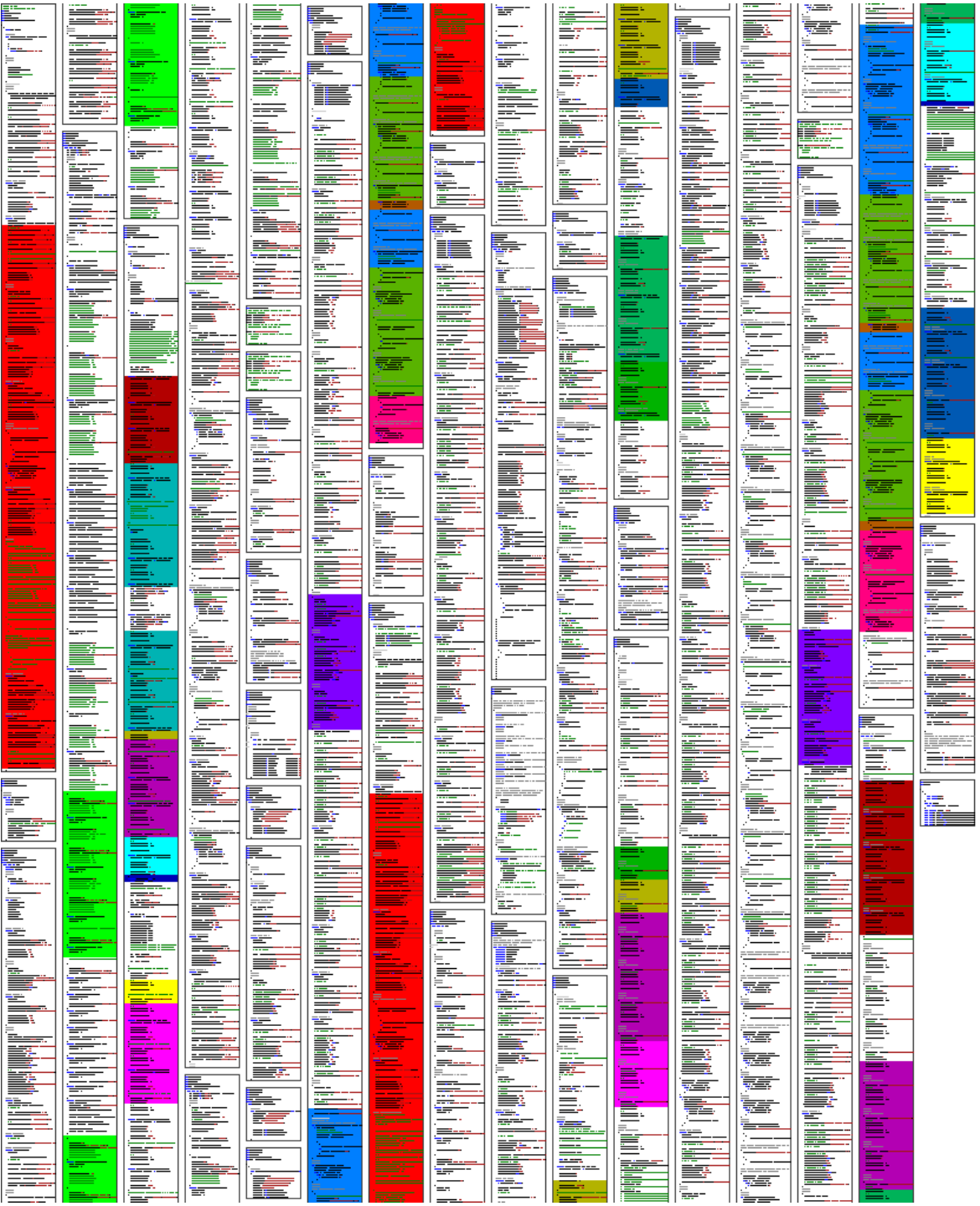


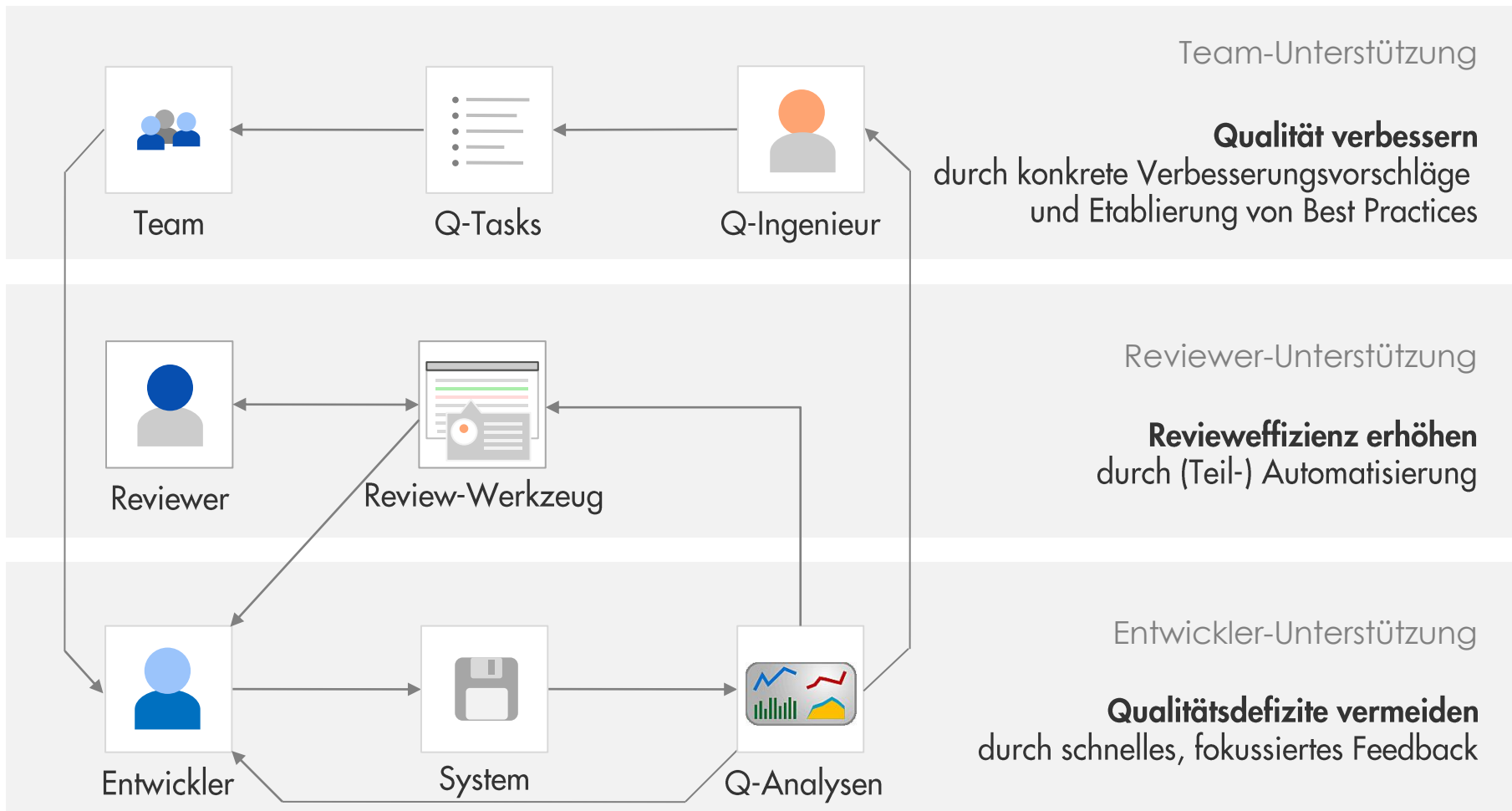


Komponente A



Komponente B





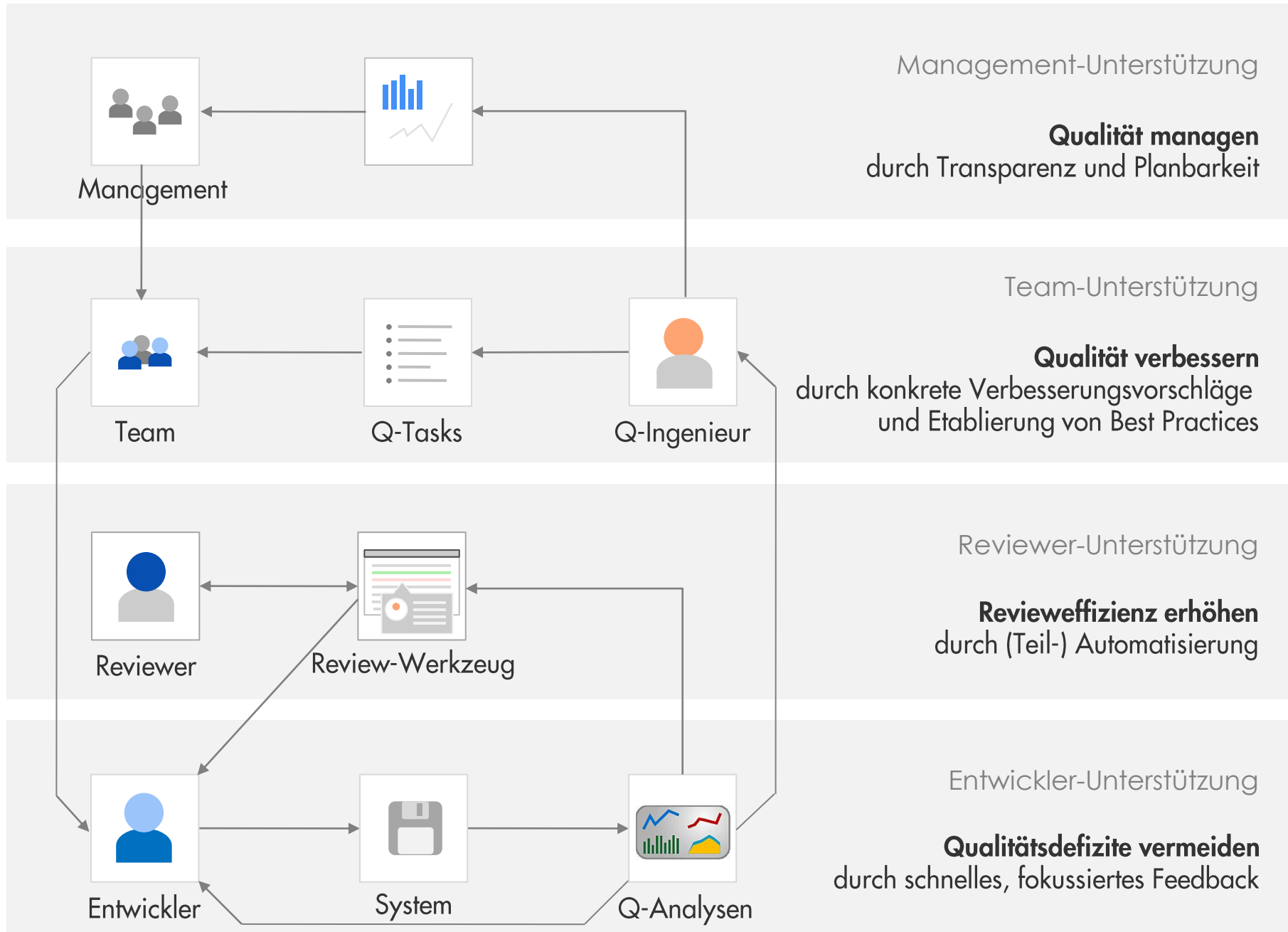
Monate

Wochen

Minuten

Sekunden

Zeit



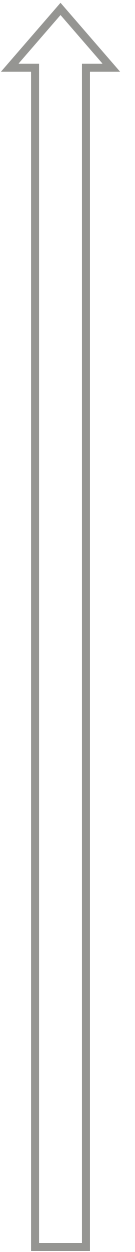
Monate

Wochen

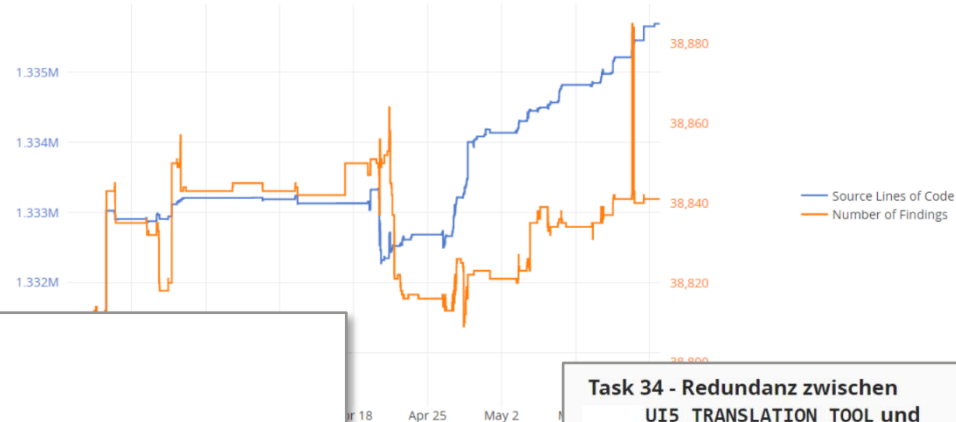
Minuten

Sekunden

Zeit



Trend 22.März - 17. Mai



System Quality Overview

Quality Indicator (QI)	Trend	Quality Indicator (QI)	Trend
Security		Redundanz	
Rote Security-Findings	→	Clone Coverage	↘
Security-Findings je 1.000 Codezeilen	↗	Codestruktur	
Codeanomalien		Struktur: Prozedurlänge	→
Korrektheit	↗	Struktur: Schachtelungstiefe	→
Codeanomalien je 1000 SLOC	→		

Task 34 - Redundanz zwischen UI5_TRANSLATION_TOOL und TRANSLATION_TOOL

created by [User] Sep 16 2021 15:22, last updated [User] Sep 16 2021 15:22

Assignee: [User]

Status:

Resolution:

Description: Der Code im neuen Paket UI5_TRANSLATION_TOOL. Da beide Pakete hier reduziert werden, z. B. durch Ver...

Tags: 1. Retro Redundanz

Task 16 - Fehlende Best Practice für Berechtigungsprüfungen von Reports und Transaktionen

created by [User] Apr 08 2021 09:14, last updated [User] Sep 19 2021 21:49

Assignee: [User]

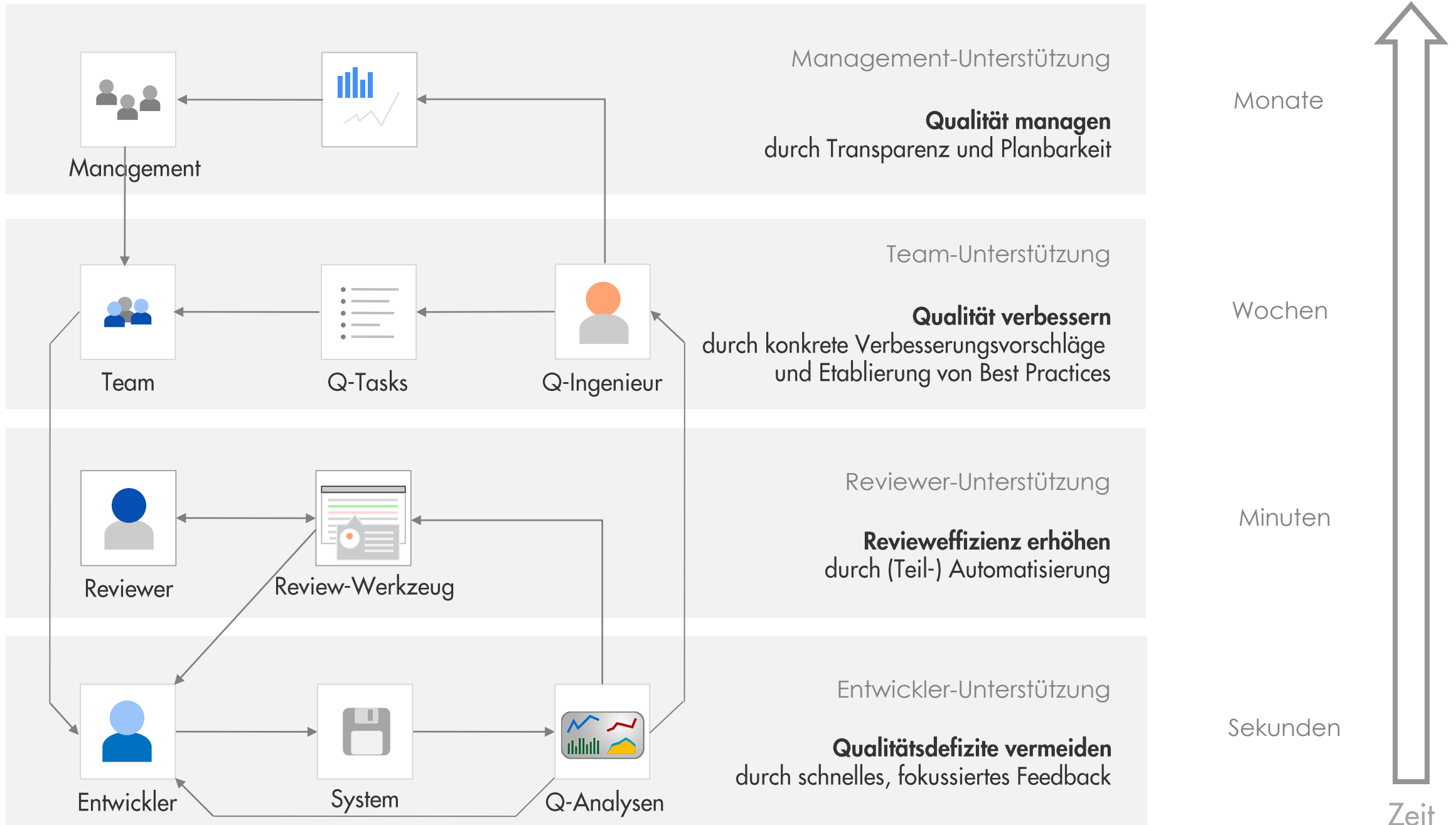
Status:

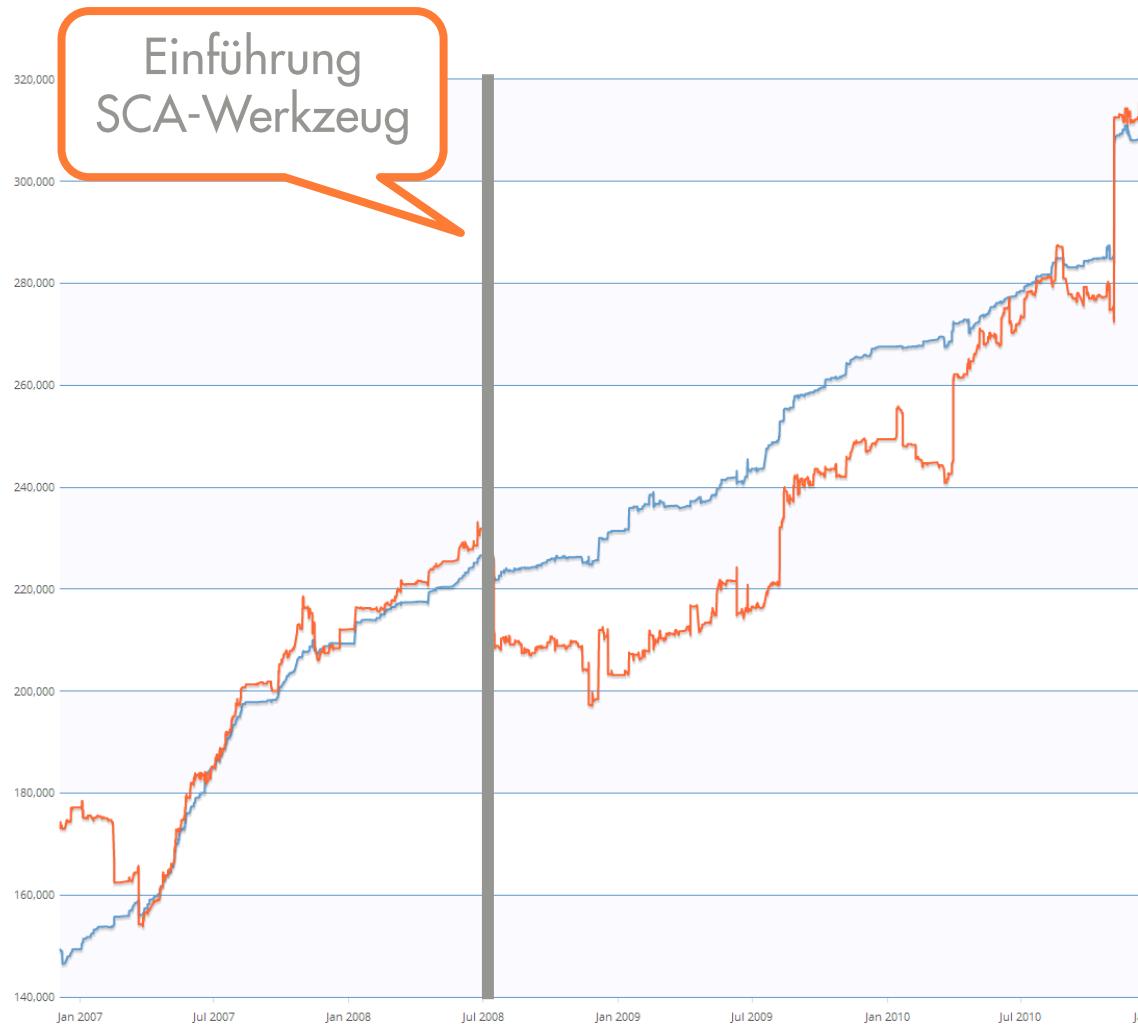
Resolution:

Description: Aufgrund des Feedbacks des Teams wurden die Teamscale-Prüfungen für die Existenz von Berechtigungsprüfungen von Reports und Transaktionen (genauer: von Aufrufen einer Transaktion im Code mittels CALL_TRANSACTION) deaktiviert. Deshalb fehlt aktuell eine Best Practice für die Berechtigungsprüfung für Reports und Transaktionen. Wir empfehlen, eine solche Best Practice zu definieren und mit Teamscale nachzuhalten, ob diese eingehalten wird.

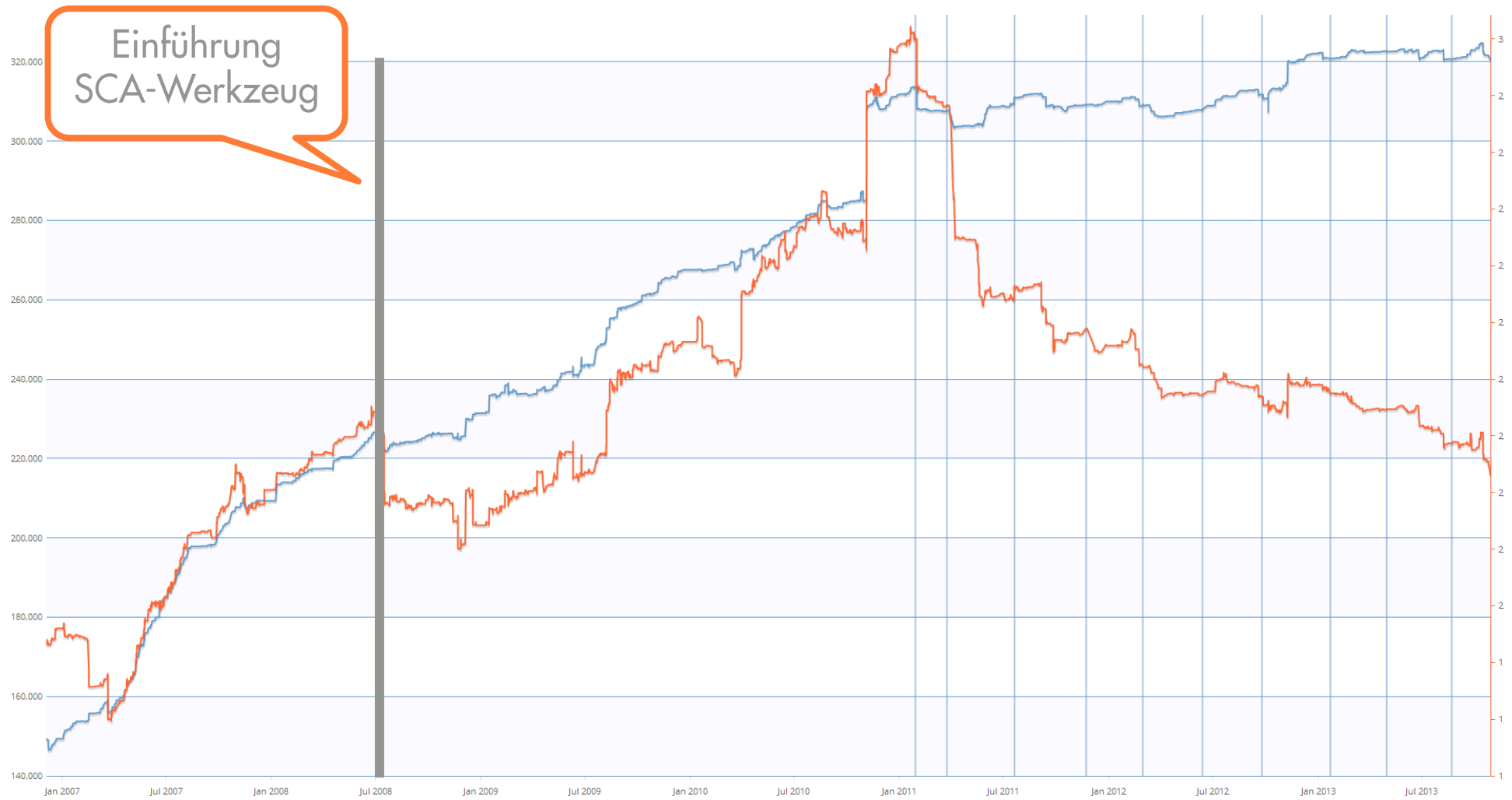
Tags: 1. Retro Security

→ Erfolgsfaktoren »Transparenz« und »Planbarkeit«



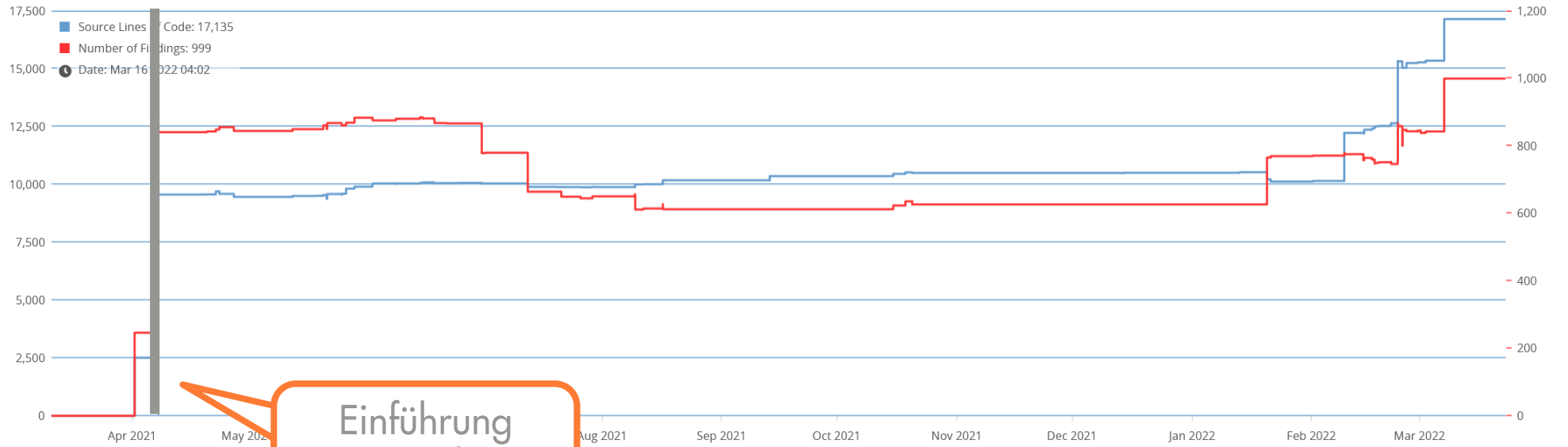


→ Kombination aus SCA-Werkzeug und QS-Prozess hat i.d.R. einen nachhaltigen Effekt



→ Kombination aus SCA-Werkzeug und QS-Prozess hat i.d.R. einen nachhaltigen Effekt

Team mit
SCA-Werkzeug



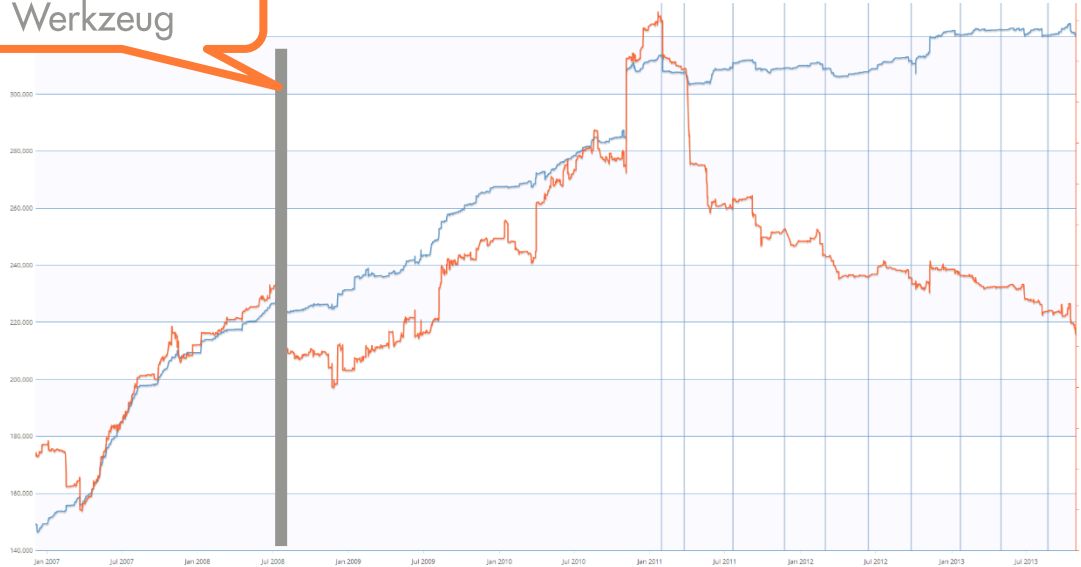
Einführung
SCA-Werkzeug

Team mit
SCA-Werkzeug
und Prozess

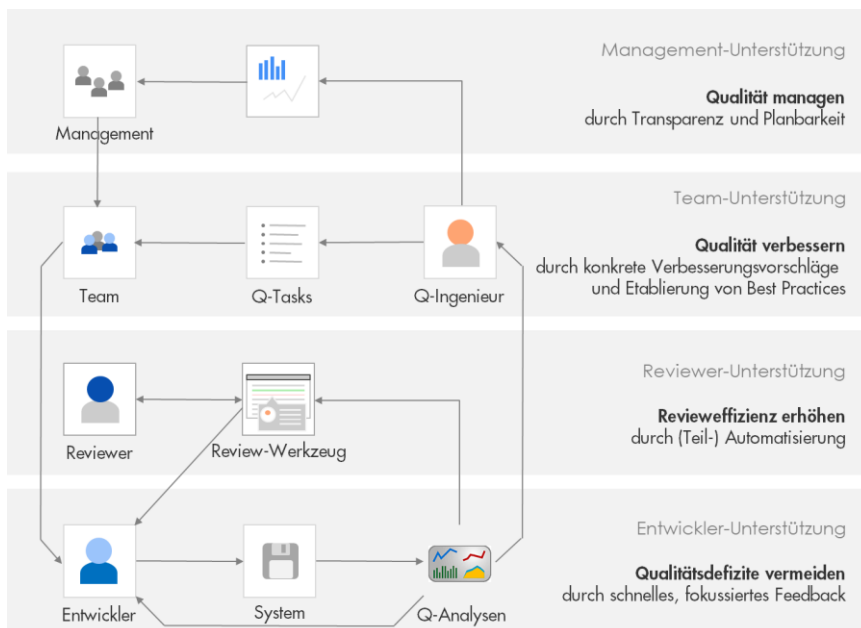


→ Kombination aus SCA-Werkzeug und QS-Prozess hat i.d.R. einen nachhaltigen Effekt

Einführung SCA-
Werkzeug



Die reine Einführung eines SCA-Werkzeugs hat häufig nur einen kurzfristigen Effekt. Kombination aus SCA-Werkzeug und QS-Prozess hat i.d.R. einen nachhaltigen Effekt.



QS-Prozess mit vier Feedbackschleifen zur Einbindung von Entwicklern und Management

Erfolgsfaktoren:

Realistisches Qualitätsziel

Keine Qualitätspolizei

Individuelle Analysekonfiguration

Schnelles, änderungsfokussiertes Feedback für Entwickler

Konkrete Verbesserungsvorschläge

Planbarkeit von Q-Verbesserungen

Transparenz über Q-Zustand und -Trends

Akzeptanz durch Entwickler und Management

»Der Weg einer nachhaltigen Verbesserung der Qualität geht nicht über ein Tool, sondern über den Prozess.«

C. Finkbeiner, Softwarearchitekt,
SEW-Eurodrive

Software Intelligence

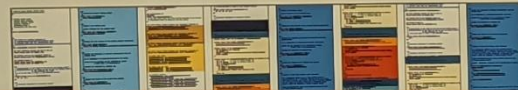
Welche Überraschungen lauern in Ihrer Software?

CQSE

Wo sind Probleme
in Ihrem Quelltext?

Welche Schwächen
hat Ihre Architektur?

Welche Änderungen
sind notwendig?



Diskussionsmöglichkeiten:

- Jetzt
- Am CQSE-Stand #31 (beim Buffet 😊)
- Via Mail: roehm@cqse.eu

Infos & Folien:

- http://cqse.eu/jfs_2022



Kontakt – Ich freue mich auf Diskussionen 😊



Dr. Tobias Röhm · roehm@cqse.eu · +49 1590 4330842

Danke an die CQSE-Kollegen, insbesondere Elmar Jürgens, für ihre Beiträge!

