

CQSE Software Intelligence Talk 2021-3

»Quality Control einführen«

Wir starten pünktlich um 10:33 Uhr

Agenda

Begrüßung

Quality Control: Technische Schulden im Griff haben

Dr. Christian Pfaller (CQSE GmbH)

Code Quality in der Krankenversicherungs-IT der VKB

Ekin Bapisch (Versicherungskammer Bayern)

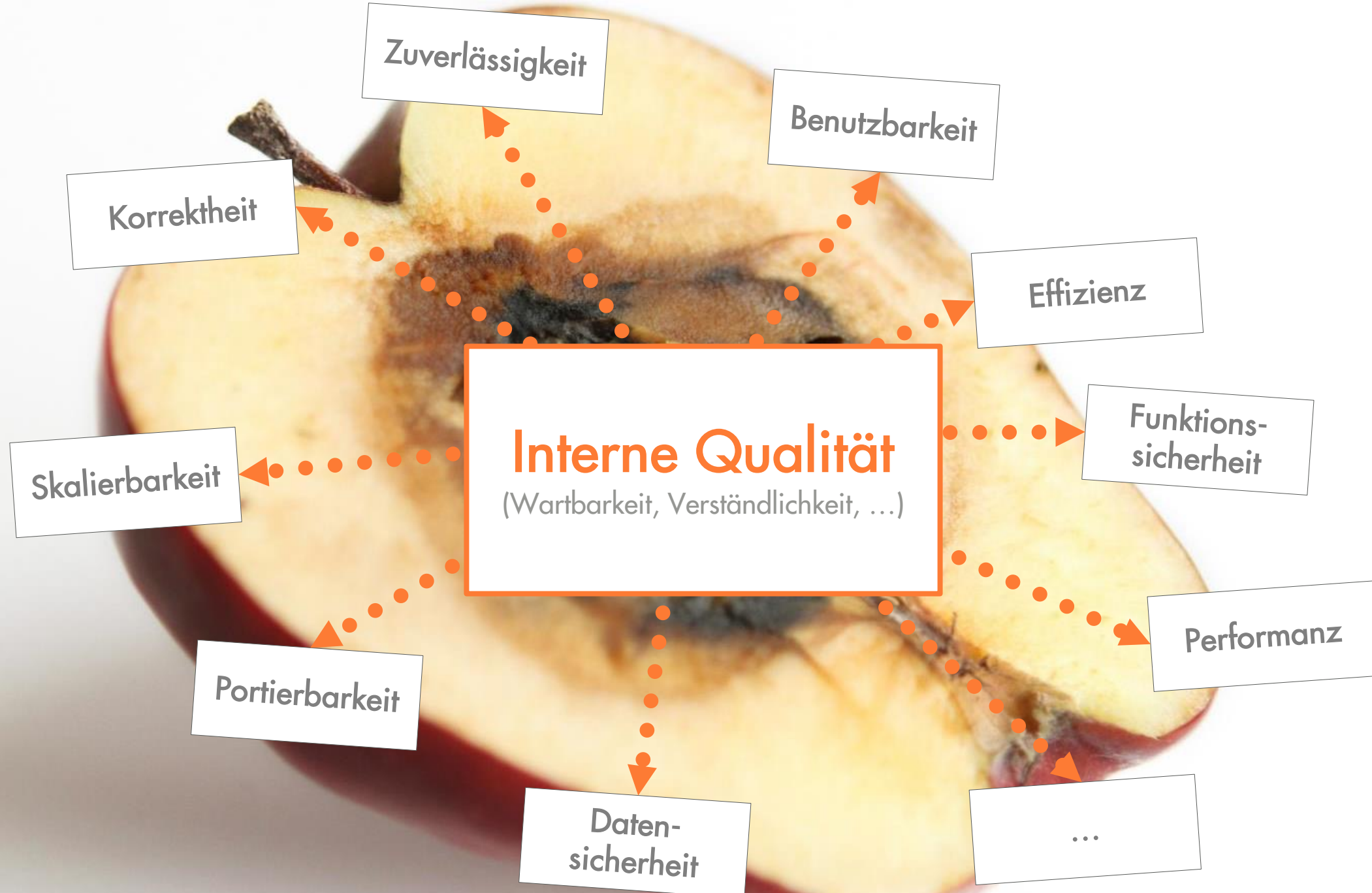
Quality Control: Technische Schulden im Griff haben



Dr. Christian Pfaller

Zukunftsfähigkeit





Zuverlässigkeit

Benutzbarkeit

Korrektheit

Effizienz

Interne Qualität
(Wartbarkeit, Verständlichkeit, ...)

Funktions-sicherheit

Skalierbarkeit

Performanz

Portierbarkeit

...

Daten-sicherheit

18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

```
2586:     }
2587:     if (arr.length == 2) {
2588:
2589:         // If the relative relationship is
2590:         // the relative from class of the type
2591:         for (int k = 0;
2592:             k < arr.length;
2593:             k++) {
2594:
2595:             Relationship r = new Relationship(
2596:                 type,
2597:                 arr[k],
2598:                 type,
2599:                 arr[k]);
2600:         }
2601:     } else {
2602:         logger.fatal(
2603:             "There can only be either "
2604:             + "one or two " + arr[0] + ".");
2605:     }
2606: }
2607: }
2608: }
2609: }
2610: } else {
2611:     logger.fatal("Invalid constant: "
2612:         + arr[0] + " " + arr[1]);
2613: }
2614: }
2615: }
2616: }
2617: } else {
2618:     logger.fatal("Invalid constant: " + arr[0] + " " + arr[1]);
2619: }
2620: }
2621: }
2622: return arr;
2623: }
```



```

/* Process the input string received prior to the
newline. */
do
{
    /* Pass the string to FreeRTOS+CLI. */
    xMoreDataToFollow = FreeRTOS_CLIProcessCommand( cInputString, cOutputString );

    /* Send the output generated by the command's
implementation. */
    sendto( xSocket, cOutputString, strlen( cOutputString ), 0, ( SOCKADDR * ) &xClient );

} while( xMoreDataToFollow != pdFALSE ); /* Until the command does not return more data.

/* All the strings generated by the command processing
have been sent. Clear the input string ready to receive
the next command. */
cInputIndex = 0;
memset( cInputString, 0x00, cmdMAX_INPUT_SIZE );

/* Transmit a spacer, just to make the command console
easier to read. */
sendto( xSocket, "\r\n", strlen( "\r\n" ), 0, ( SOCKADDR * ) &xClient );
}
else
{
    if( cInChar == '\r' )
    {
        /* Ignore the character. Newlines are used to
detect the end of the input string. */
    }
    else if( cInChar == '\b' )
    {
        /* Backspace was pressed. Erase the last character
in the string - if any. */
        if( cInputIndex > 0 )
        {
            cInputIndex--;
            cInputString[ cInputIndex ] = '\0';
        }
    }
}
else
{

```

```

/* Process the input string received prior to the
newline. */
do
{
    /* Pass the string to FreeRTOS+CLI. */
    xMoreDataToFollow = FreeRTOS_CLIProcessCommand( cInputString, cOutputString );

    /* Send the output generated by the command's
implementation. */
    sendto( xSocket, cOutputString, strlen( cOutputString ), 0, ( SOCKADDR * ) &xClient );

} while( xMoreDataToFollow != pdFALSE ); /* Until the command does not return more data.

/* All the strings generated by the command processing
have been sent. Clear the input string ready to receive
the next command. */
cInputIndex = 0;
memset( cInputString, 0x00, cmdMAX_INPUT_SIZE );

/* Transmit a spacer, just to make the command console
easier to read. */
sendto( xSocket, "\r\n", strlen( "\r\n" ), 0, ( SOCKADDR * ) &xClient );
}
else
{
    if( cInChar == '\r' )
    {
        /* Ignore the character. Newlines are used to
detect the end of the input string. */
    }
    else if( ( cInChar == '\b' ) || ( cInChar == cmdASCII_DEL ) )
    {
        /* Backspace was pressed. Erase the last character
in the string - if any. */
        if( cInputIndex > 0 )
        {
            cInputIndex--;
            cInputString[ cInputIndex ] = '\0';
        }
    }
}
else
{

```

```

/* Process the input string received prior to the
newline. */
do
{
    /* Pass the string to FreeRTOS+CLI. */
    xMoreDataToFollow = FreeRTOS_CLIProcessCommand( cInputString, cOutputString );

    /* Send the output generated by the command's
implementation. */
    sendto( xSocket, cOutputString, strlen( cOutputString ), 0, ( SOCKADDR * ) &xClient );

} while( xMoreDataToFollow != pdFALSE ); /* Until the command does not return more data.

/* All the strings generated by the command processing
have been sent. Clear the input string ready to receive
the next command. */
cInputIndex = 0;
memset( cInputString, 0x00, cmdMAX_INPUT_SIZE );

/* Transmit a spacer, just to make the command console
easier to read. */
sendto( xSocket, "\r\n", strlen( "\r\n" ), 0, ( SOCKADDR * ) &xClient );
}
else
{
    if( cInChar == '\r' )
    {
        /* Ignore the character. Newlines are used to
detect the end of the input string. */
    }
    else if( cInChar == '\b' )
    {
        /* Backspace was pressed. Erase the last character
in the string - if any. */
        if( cInputIndex > 0 )
        {
            cInputIndex--;
            cInputString[ cInputIndex ] = '\0';
        }
    }
}
else
{

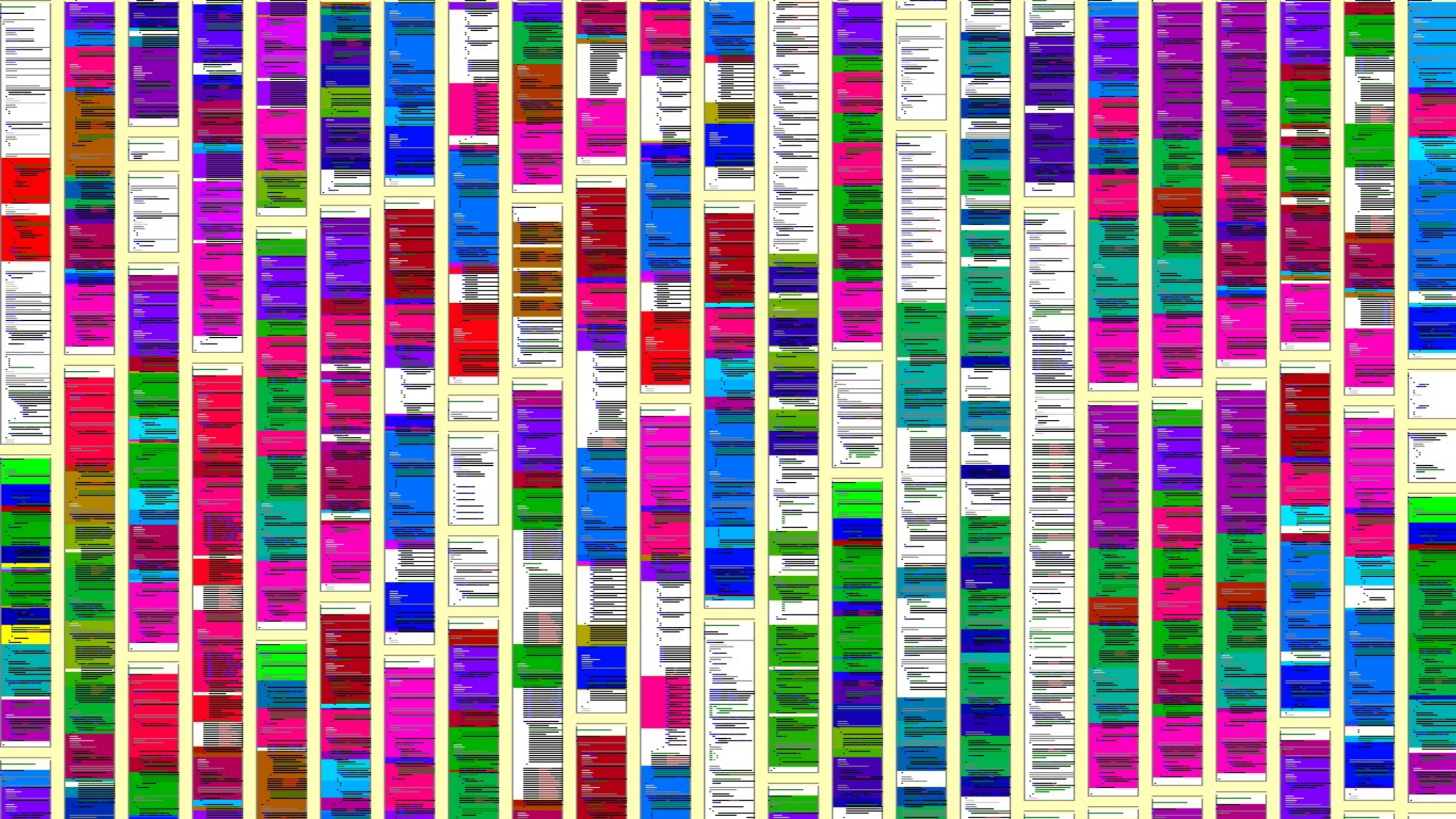
```



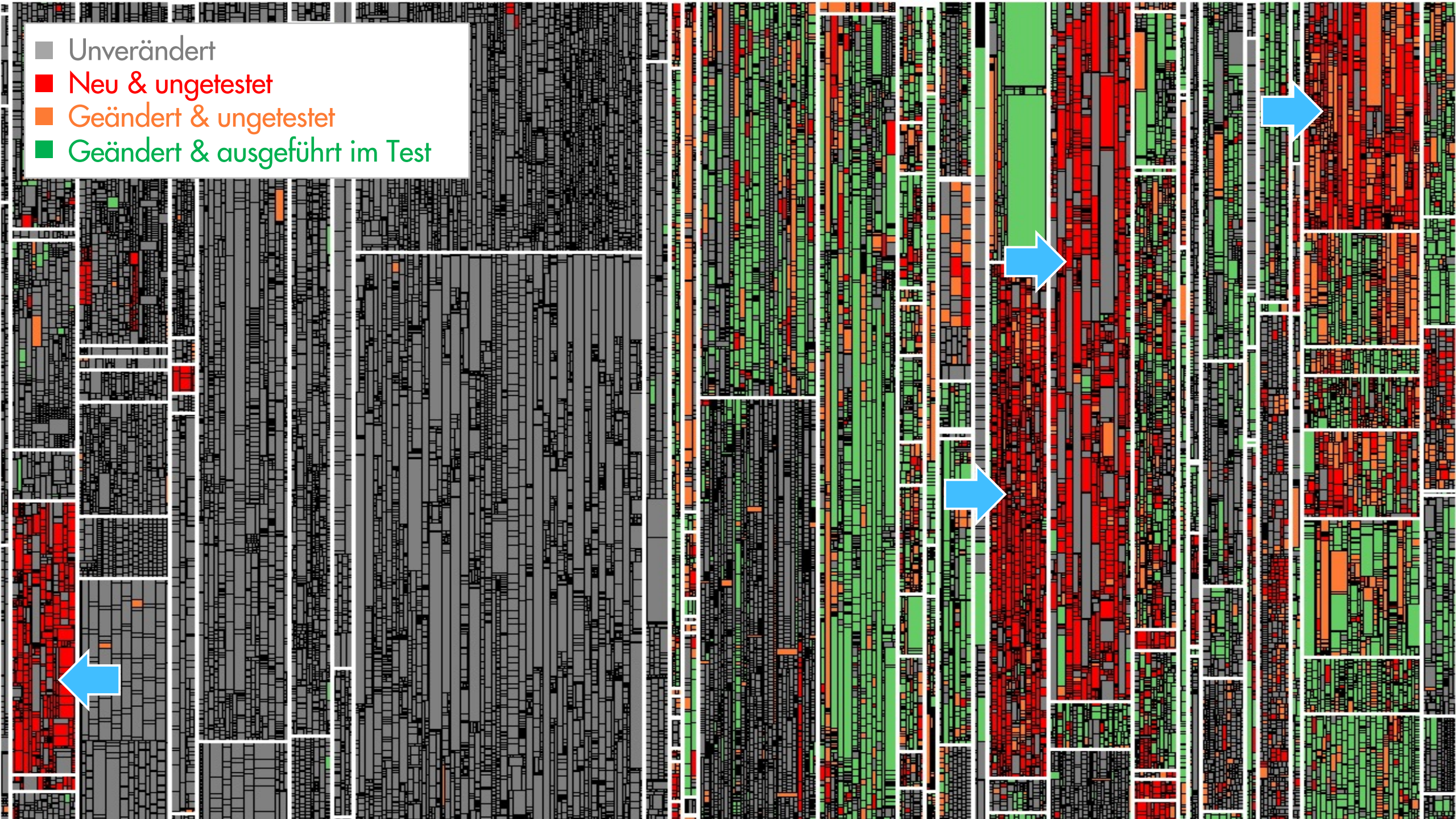
```

FreeRTOS+CLI: Hello World!
\r\n

```



- Unverändert
- Neu & ungetestet
- Geändert & ungetestet
- Geändert & ausgeführt im Test



Sicherheitslücke "goto fail": Was Apple-Nutzer jetzt tun müssen

```
if ((err = ReadyHash(&SSLHashSHA1, &hashCtx)) != 0)
    goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &clientRandom)) != 0)
    goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
    goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
    goto fail;
    goto fail;
    goto fail;
if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
    goto fail;
```

Fotos

```
err = sslRawVerify(ctx,
                   ctx->peerPubKey,
                   dataToSign,
```

```
/* plaintext */
```

SPIEGEL ONLINE

iPhones sind gefährdet, iPads und Mac-Rechner ebenfalls: Der Großteil der Apple-Geräte hat eine gravierende Sicherheitslücke. Für iPhone und iPad gibt es schon Updates, Mac-Nutzer müssen warten und höllisch aufpassen.

Montag, 24.02.2014 - 11:53 Uhr

Drucken | Versenden | Merken

Nutzer von Apple-Geräten sind in großer Gefahr: In den Betriebssystemen des Konzerns für Tablets und Smartphones (iOS) sowie Mac-Rechner (OSX) klafft eine gravierende Sicherheitslücke.

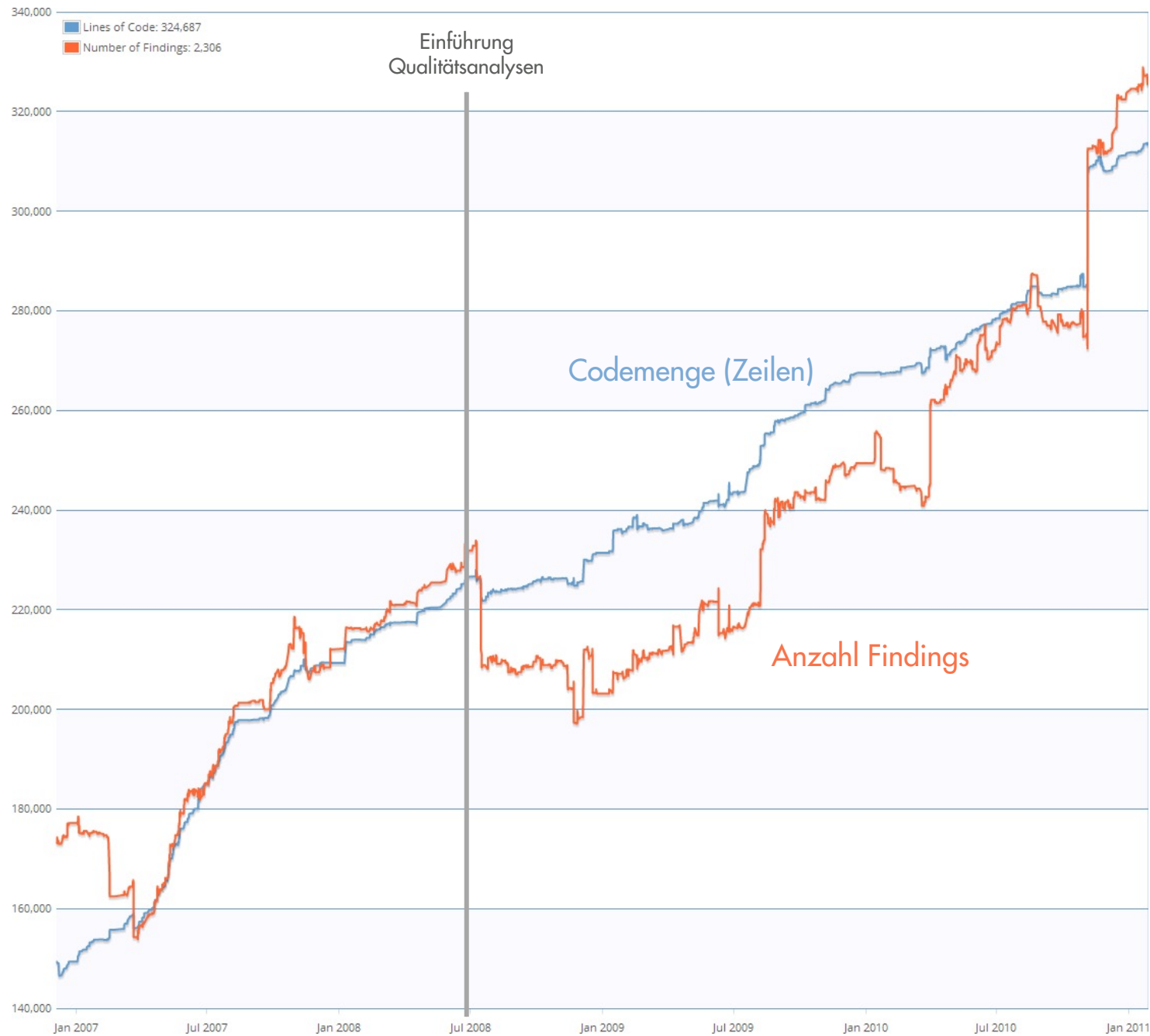
Hier die Antworten auf die wichtigsten Fragen zum Apple Lock:

» Uns fehlt aktuell leider aufgrund anderer Themen die Zeit uns mit Qualität zu beschäftigen. «

– Jeder 2. (oder mehr)

Wir brauchen ein
Werkzeug!

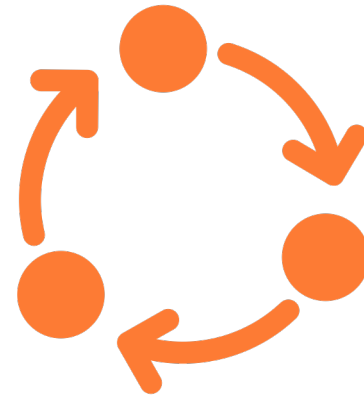




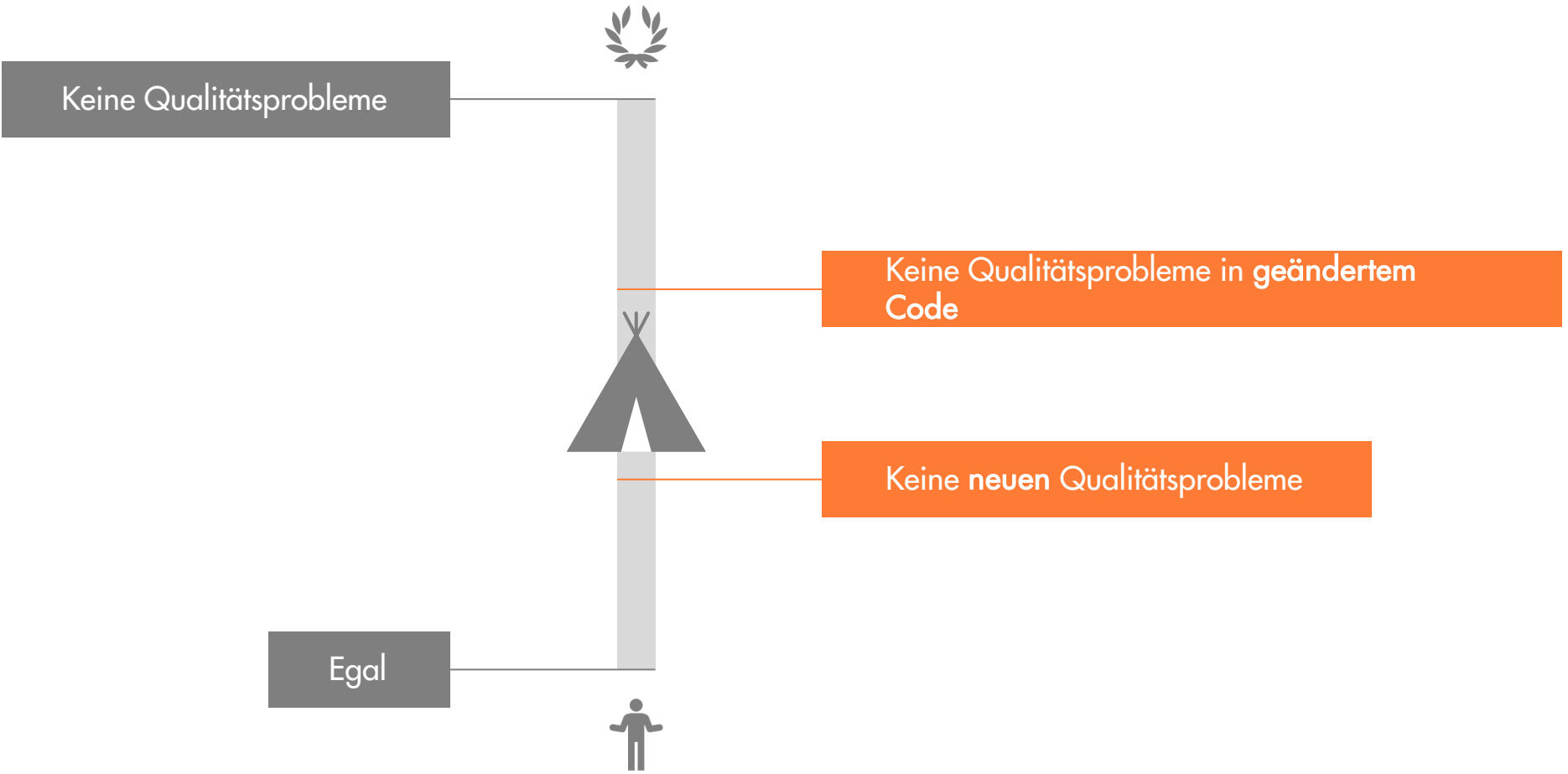
Euphorie am Anfang,
aber dann ...

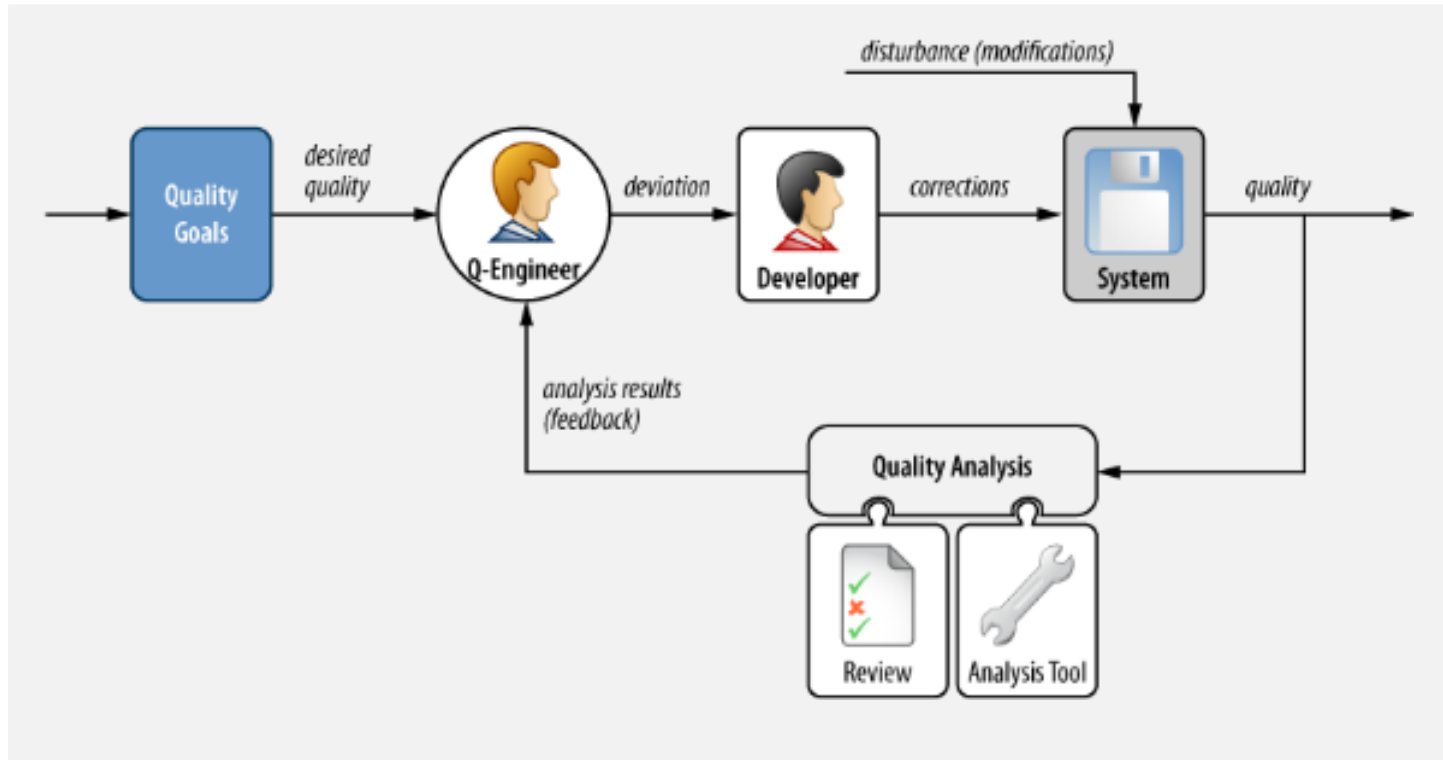


Ziel



Prozess





TUM
Technische Universität München



conQAT

CQSE

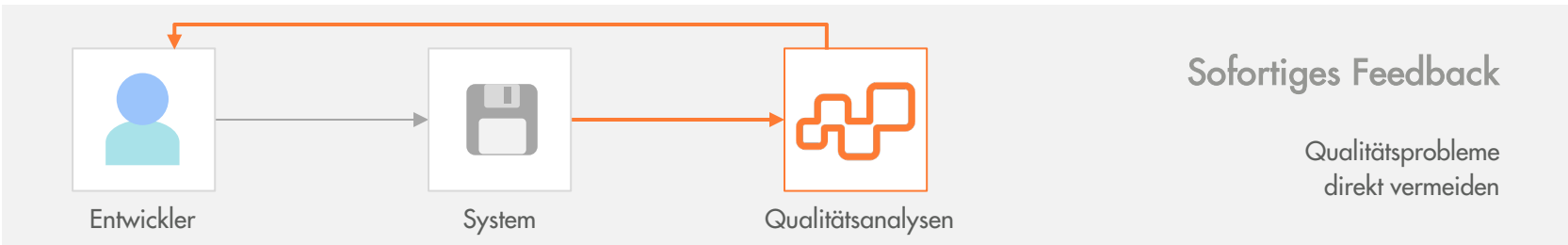
Teamscale



2009

2011

2014



Project

- jabref C:\data\git\jabref
 - .circleci
 - .github
 - .gradle
 - .idea
 - buildres
 - buildSrc
 - config
 - docs
 - gradle
 - javafx
 - lib
 - licenses
 - scripts
 - snap

```

91     private final StateManager stateManager;
92
93     @
94     public EntryEditor(BasePanel panel, EntryEditorPreferences preferences, FileUpdate
95         this.panel = panel;
96         this.databaseContext = panel.getBibDatabaseContext();
97         this.undoManager = panel.getUndoManager();
98         this.preferences = Objects.requireNonNull(preferences);
99         this.dialogService = dialogService;
100        this.taskExecutor = taskExecutor;
101        this.stateManager = stateManager;
102
103        fileLinker = new ExternalFilesEntryLinker(externalFileTypes, Globals.prefs.get
104
105
106
107
    
```

Violation of method length threshold (source lines of code) of 75: 84

EntryEditor > setupKeyBindings() > event -> {...}

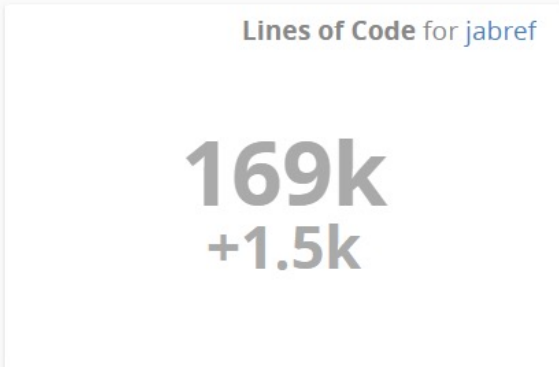
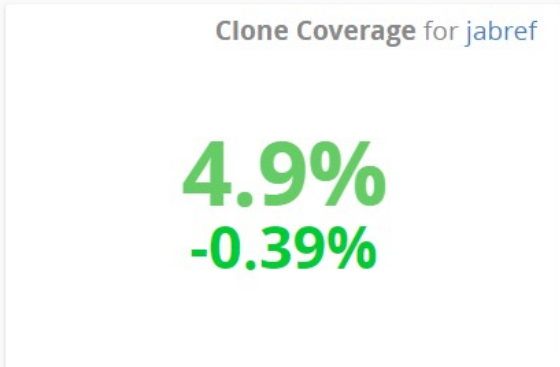
Findings

Findings	Lines	Category	Group
Private method 'navigateToPreviousEntry' is unused	256	Code Anomalies	Unused code
Statements in Conditional Expression are equal	229-231	Code Anomalies	Bad practice
Statements in Conditional Expression are equal	225-227	Code Anomalies	Bad practice
Violation of file size threshold (source lines of code) of 300: 301	0	Structure	File Size
Violation of method length threshold (source lines of code) of	199-235	Structure	Method Length
Violation of method length threshold (source lines of code) of	198-236	Structure	Method Length
Violation of method length threshold (source lines of code) of	135-189	Structure	Method Length
Violation of method length threshold (source lines of code) of	94-191	Structure	Method Length



Commit Message	Date
Merge remote-tracking branch 'upstream/master' into prefs_mvvm_external by Carl Christian Snethlage as revision 55bdf9d1 in master Files: 2 added, 1 deleted Findings: 42 (1), 1 (2), 1 (3)	Jun 14 2019 17:43
Merge pull request #5033 from calixtus/preferences_mvvm by Christoph as revision d811c175 in master Files: 8 added, 4 changed, 3 deleted Findings: 115 (1), 4 (2), 14 (3)	Jun 14 2019 17:31
Shutdown Unirest workers (#5054) by Stefan Kolb as revision 47e3cc56 in master Files: 1 changed Findings: 1 (1), 1 (2)	Jun 14 2019 09:42
Bibentrysource contextmenu (#5007) by Carl Christian Snethlage as revision eb42850f in master Files: 1 changed Findings: 1 (1)	Jun 10 2019 08:58
Add warning for empty BibTeX keys in entry editor (#5025) by ms111ds as revision 9cc4e656 in master Files: 2 changed Findings: 1 (2)	Jun 09 2019 08:07

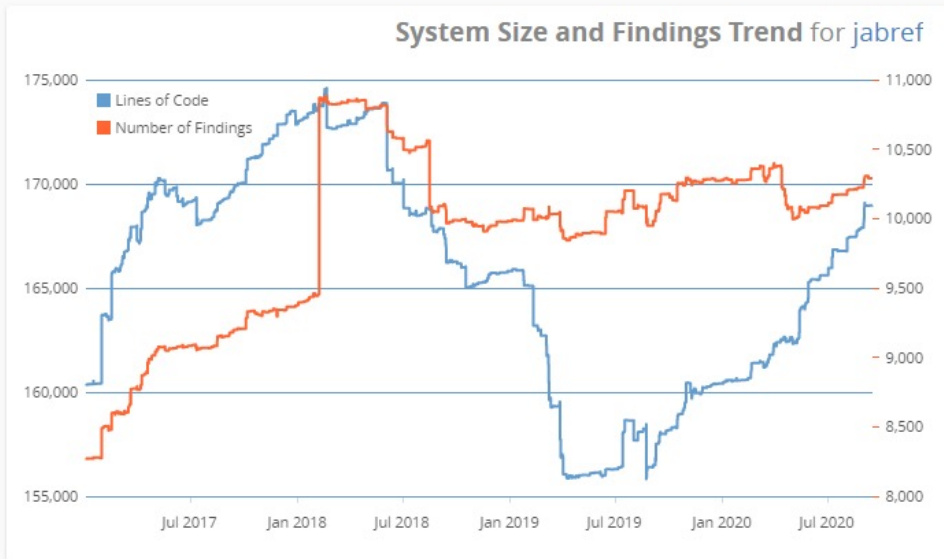
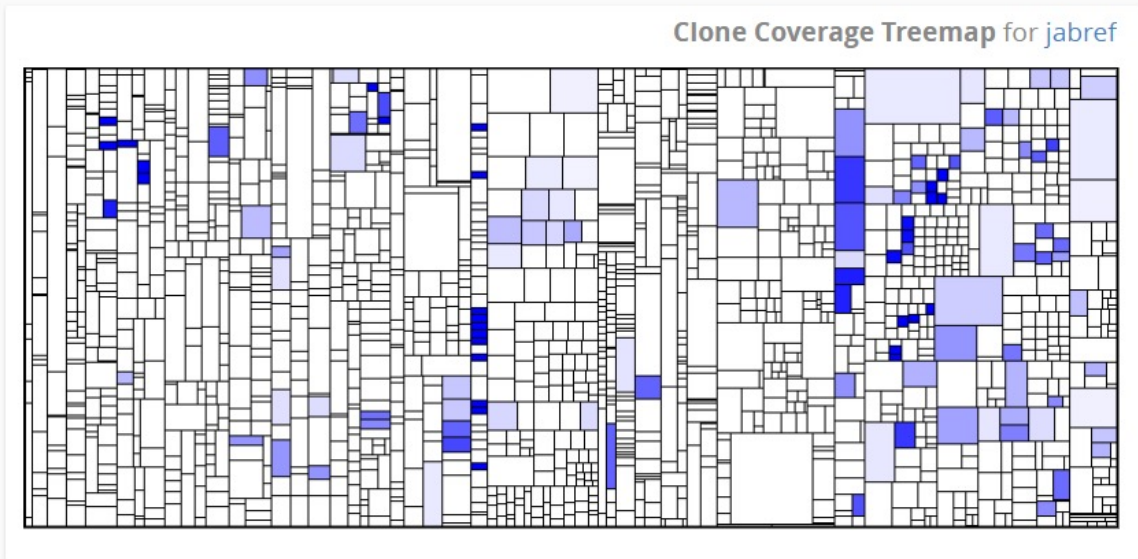
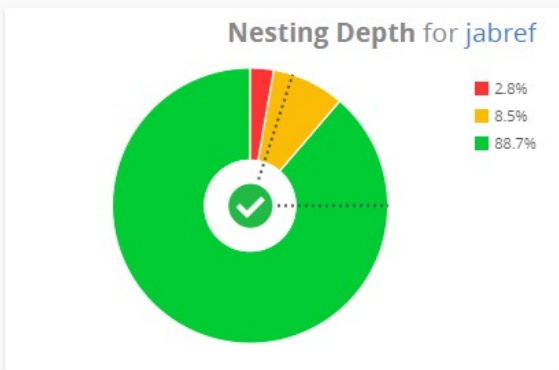
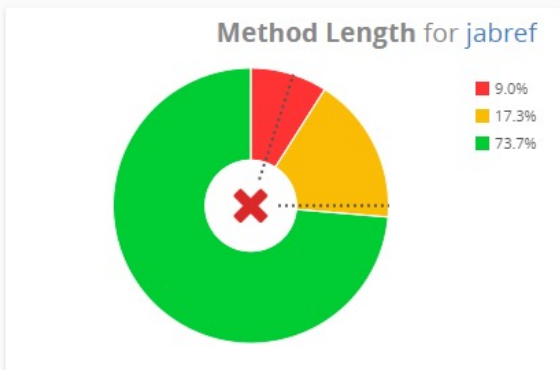
- Dashboard
- Activity
- Findings
- Metrics
- Test Gaps
- Quality Control
- Architecture
- Delta
- Project Configuration
- System
- Admin

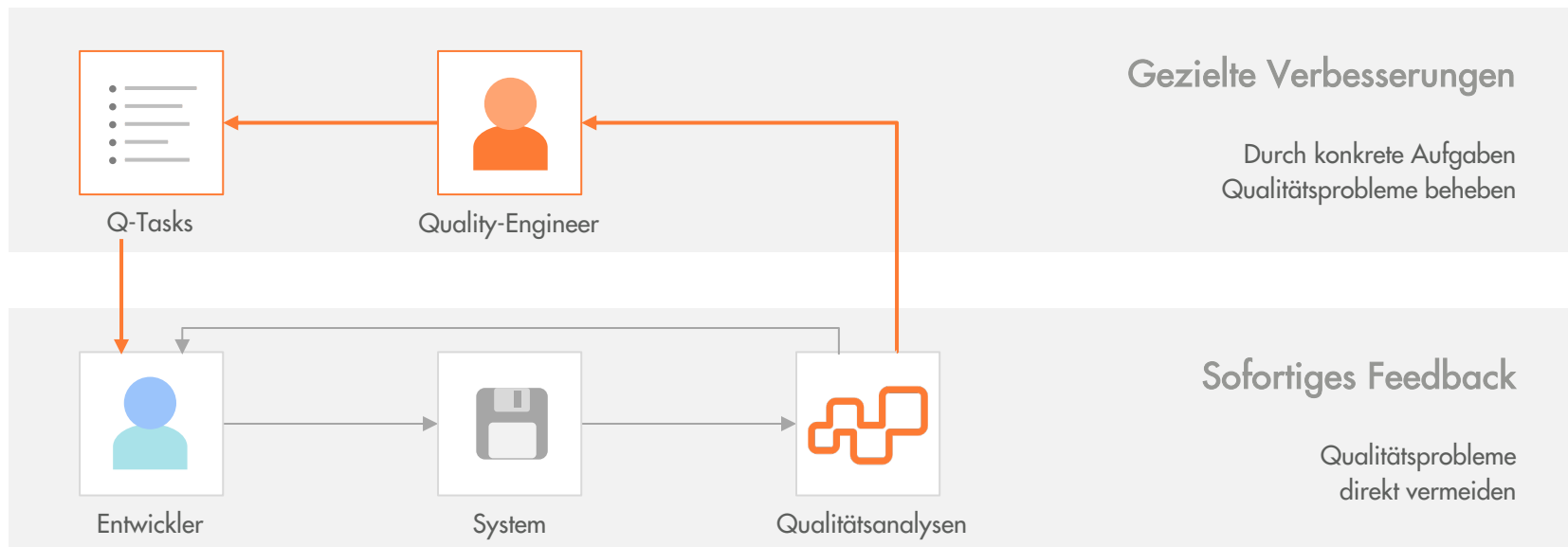


Overall Rating for jabref

Structure		
File Size Assessment	<div style="width: 100%;"><div style="width: 100%;"></div></div>	↗
Method Length Assessment	<div style="width: 100%;"><div style="width: 100%;"></div></div>	↗
Nesting Depth Assessment	<div style="width: 100%;"><div style="width: 100%;"></div></div>	↗
Cloning		
Clone Coverage	4.9%	↗
Findings		
Findings Density	60.9	↗
Architecture		
Architecture Violations	N/A	→
Tests		
Unit Test Coverage	0.0%	→

Trend since 30 Days ago







⚠ WARNING
BRIGHT LIGHT
- Do not stare into light beam
- Do not view at close range
Failure to do so will cause
permanent eye damage

Task 1 - Redundant code within PushToEmacsSettings.java and PushToVimSettings.java

created by  Default Administrator Today 18:39, last updated Today 18:41

Assignee

Status

Open

Resolution

None

Description

The two classes PushToEmacsSettings.java and PushToVimSettings.java have redundant code. Consider to introduce a base class (e.g. GenericSettings) and use the constructor of the new base class.

Tags

Duplicated Code

Report#1

Findings

2 open 0 resolved 0 blacklisted

- Clone with 2 instances of length 10 in src/main/java/org/jabref/gui/push/PushToEmacsSettings.java
- Clone with 2 instances of length 10 in src/main/java/org/jabref/gui/push/PushToVimSettings.java

Task 3 - Long Method in EntryEditor.java

created by  Default Administrator Today 18:53, last updated Today 18:57

Assignee

Status

Open

Resolution

None

Description

The overly long constructor in EntryEditor.java should be restructured, e.g. extract a parameterized method for the three cases depending on dragDropPreferencesType



Tags


Report#1

Method Length

Findings

2 open 0 resolved 0 blacklisted

- Violation of method length threshold (source lines of code) of 30: 50 in src/main/java/org/jabref/gui/entryeditor/EntryEditor.java:135-189 
- Violation of method length threshold (source lines of code) of 75: 84 in src/main/java/org/jabref/gui/entryeditor/EntryEditor.java:94-191 

 Edit Task

Task 1 - Redundant code within PushToEmacsSettings.java and PushToVimSettings.java

created by  Default Administrator Today 18:39, last updated Today 18:41

Assignee

Status

Open

Resolution

None

Description

The two classes
Consider to in
the constructo

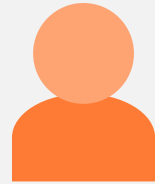
Tags

Duplicated Co

Findings

2 open 0

- Clone with 2 instances of length 10
- Clone with 2 instances of length 10



- relevante Findings
- konkrete Verbesserungsvorschläge

→ Unterstützung für Entwickler



Tags

Report#1

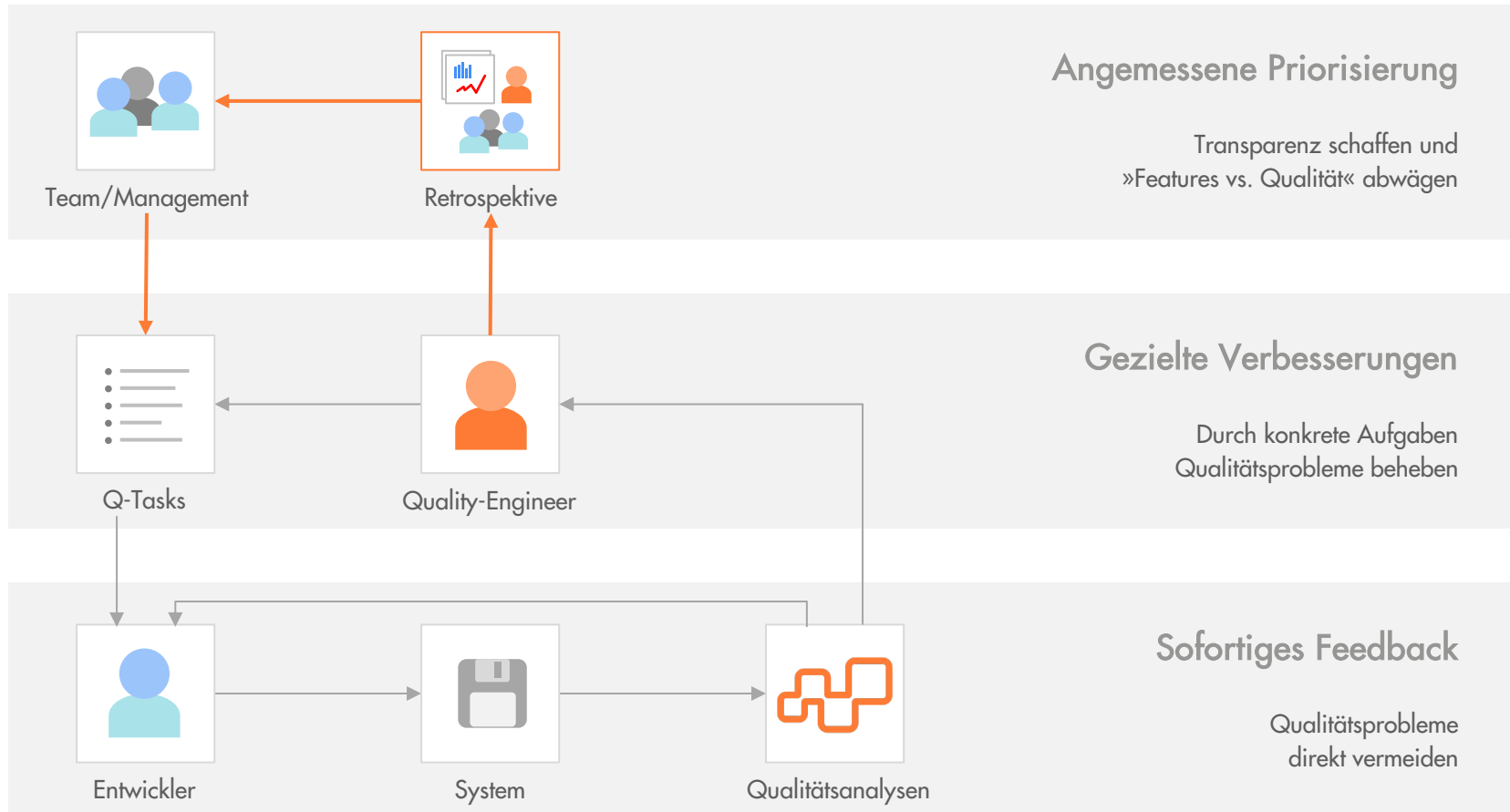
Method Length

Findings

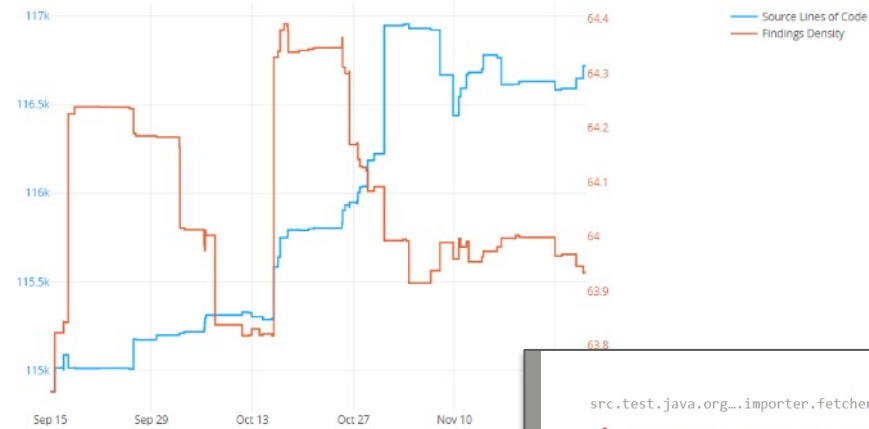
2 open 0 resolved 0 blacklisted

- Violation of method length threshold (source lines of code) of 30: 50 in src/main/java/org/jabref/gui/entryeditor/EntryEditor.java:135-189 
- Violation of method length threshold (source lines of code) of 75: 84 in src/main/java/org/jabref/gui/entryeditor/EntryEditor.java:94-191 

Edit Task



Size and Findings Trend



Überblick Qualitätsindikatoren 15.09.2019 – 28.11.2019

Architektur	✓	weiterhin optimal
Duplizierter Code	✓	weiterhin optimal
Potentielle Fehler	→	keine wesentliche Änderung
Codeanomalien	↗	verbessert
Methodenlänge	↘	verschlechtert
Schachtelungstiefe	↗	verbessert

src.test.java.org...importer.fetcher.AstrophysicsDataSystemTest.java

⚠ Task 13: contains() sollte für List nicht verwendet werden

```
135     }
136
137     @Test
138     public void testGetName() {
139         assertEquals("SAO/NASA Astrophysics Data System", fetcher.getName());
140     }
141
142     @Test
143     public void searchByQueryFindsEntry() throws Exception {
144         List<BibEntry> fetchedEntries = fetcher.performSearch("Diez slice theorem Lie");
145         assertTrue(fetchedEntries.contains(diezSliceTheoremEntry));
146     }
147
148     @Test
149     public void searchByEntryFindsEntry() throws Exception {
150         BibEntry searchEntry = new BibEntry();
151         searchEntry.setField(StandardField.TITLE, "slice theorem");
152         searchEntry.setField(StandardField.AUTHOR, "Diez");
153
154         List<BibEntry> fetchedEntries = fetcher.performSearch(searchEntry);
155         assertFalse(fetchedEntries.isEmpty());
156     }
157 }
```

FakeSystem 11/2019

Sep to Nov 2019
Project: Jabref

Überblick Qualitätsindikatoren 15.09.

Architektur

Duplizierter Code

Potentielle Fehler

Codeanomalien

Methodenlänge

Schachtelungstiefe



verbessert



- Qualität regelmäßig auf der Agenda
- Gemeinsame Diskussion
- Einheitliches Qualitätsverständnis
- Transparenz schaffen
- Handlungsbedarf identifizieren
- Feedback an Quality Engineer

➔ Planung und Priorisierung

— Source Lines of Code
— Findings Density

et werden

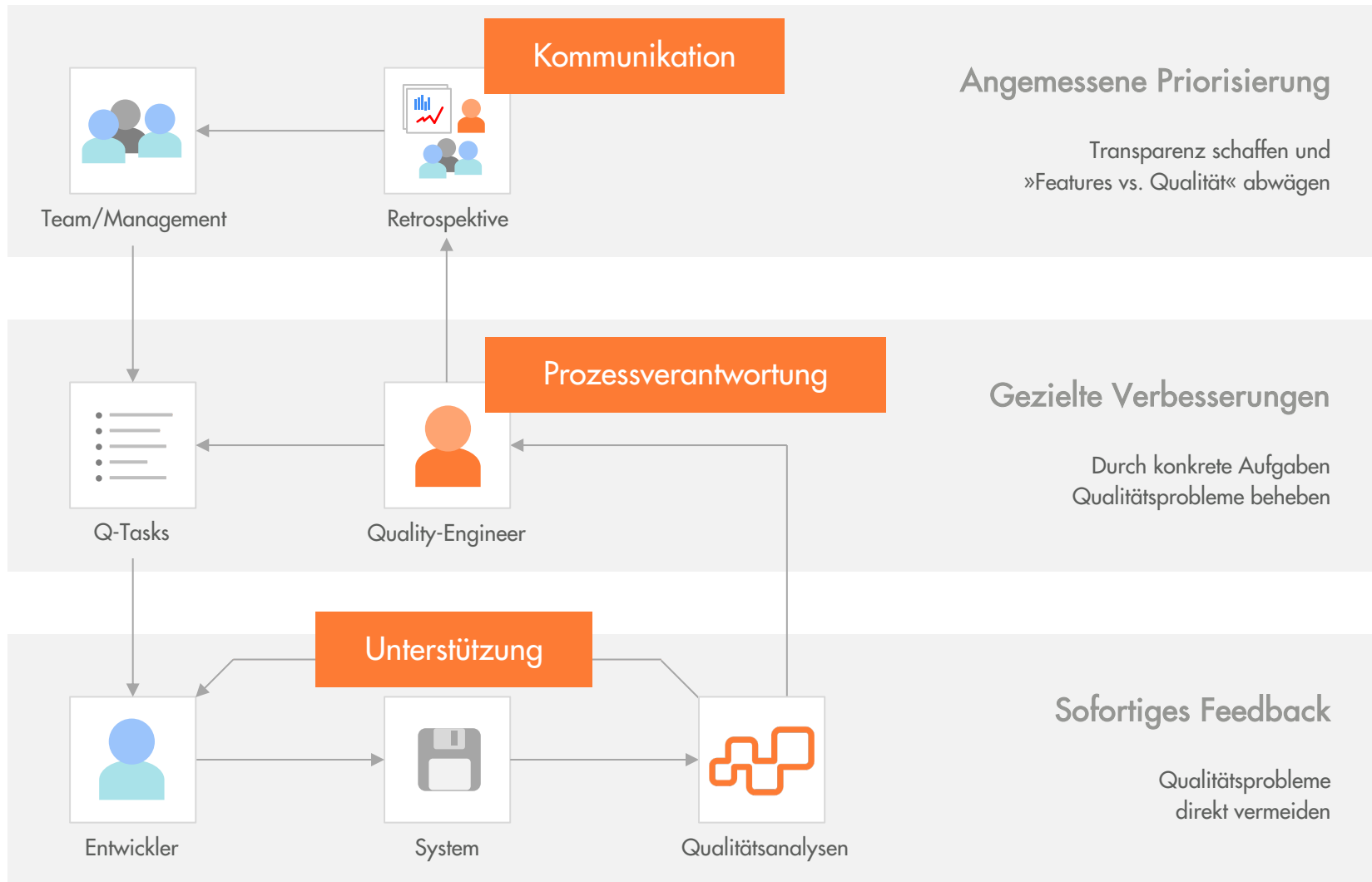
lie");

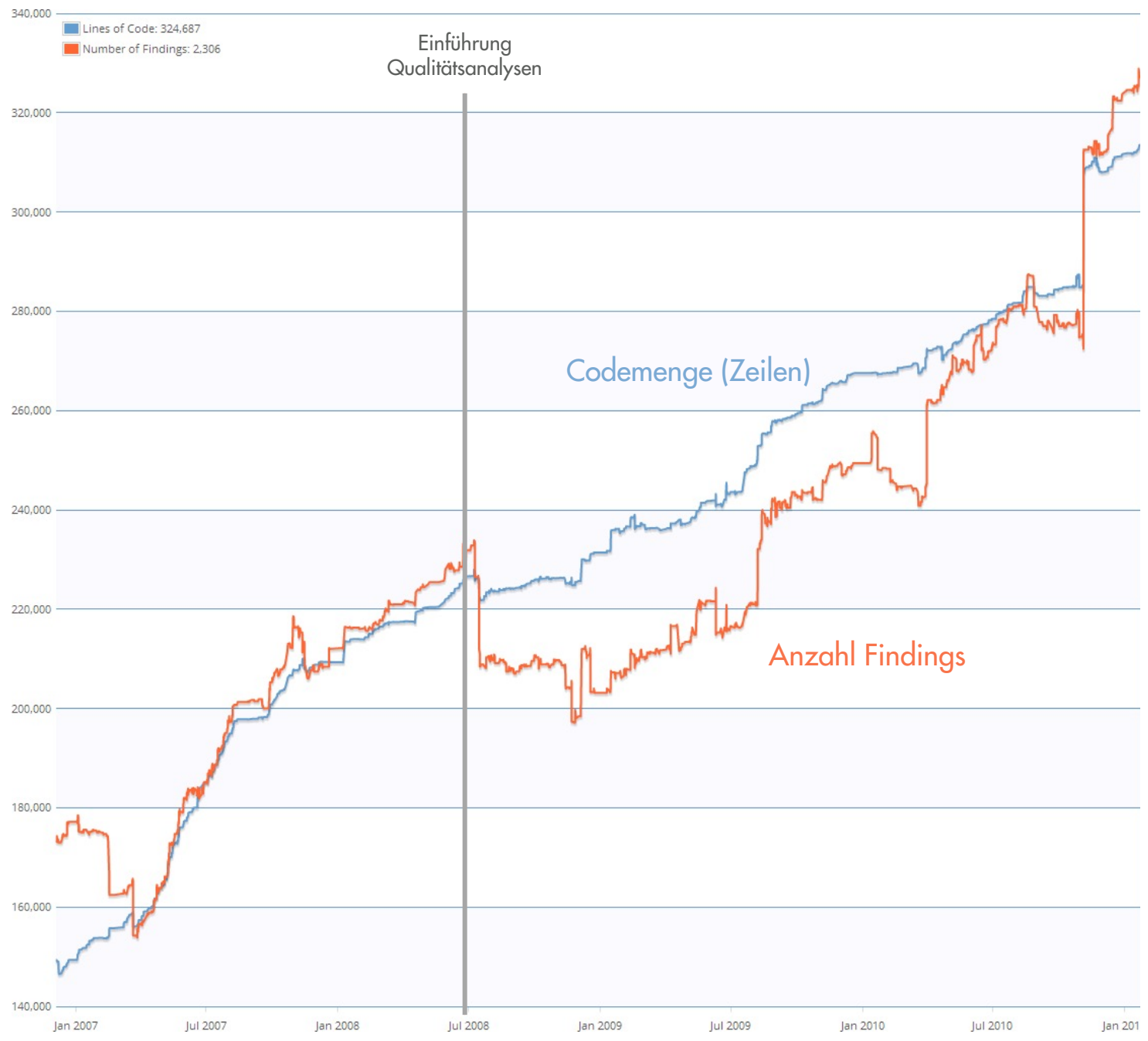
153

154

155

```
List<BibEntry> fetchedEntries = fetcher.performSearch(searchEntry);  
assertFalse(fetchedEntries.isEmpty());
```



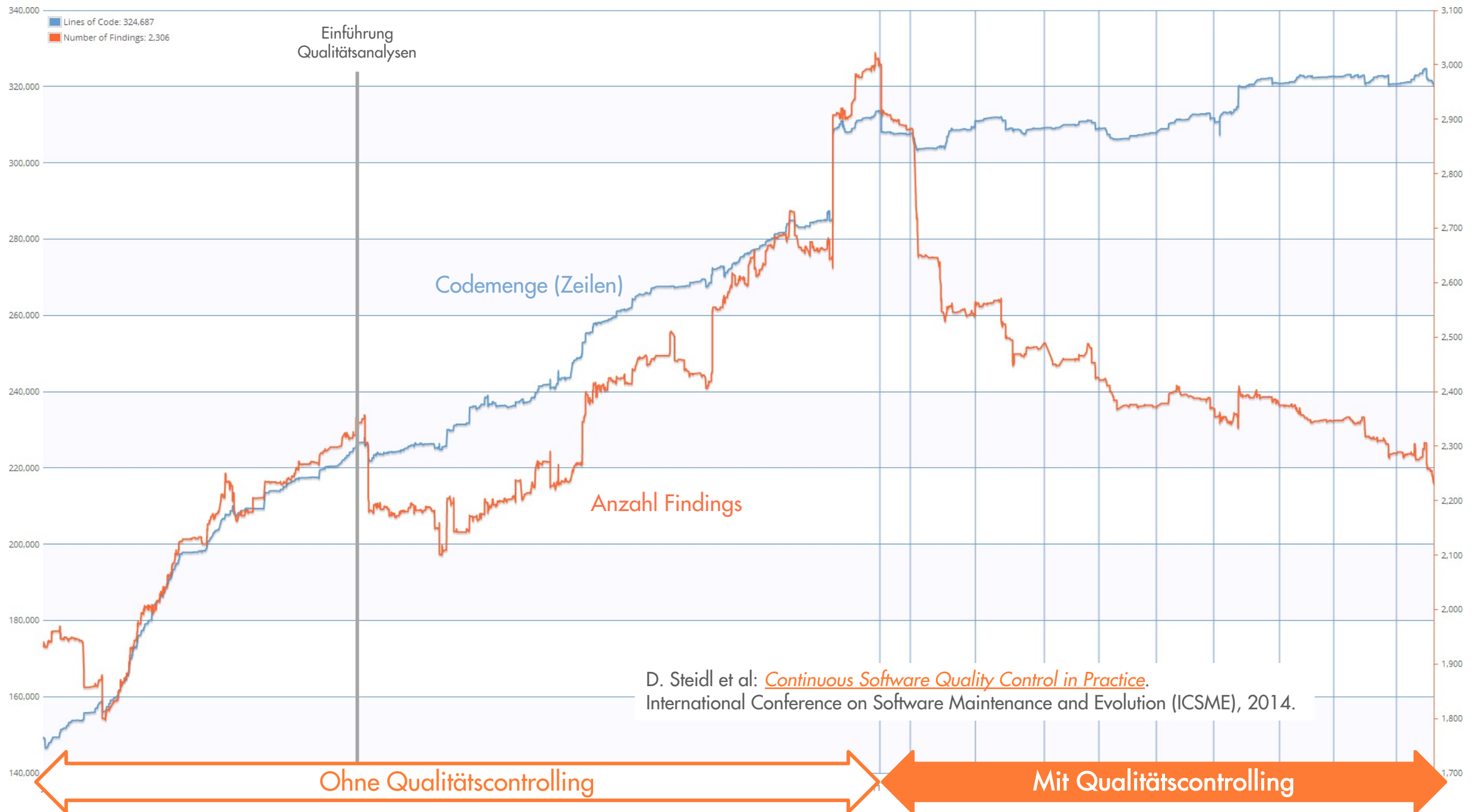


Einführung
Qualitätsanalysen

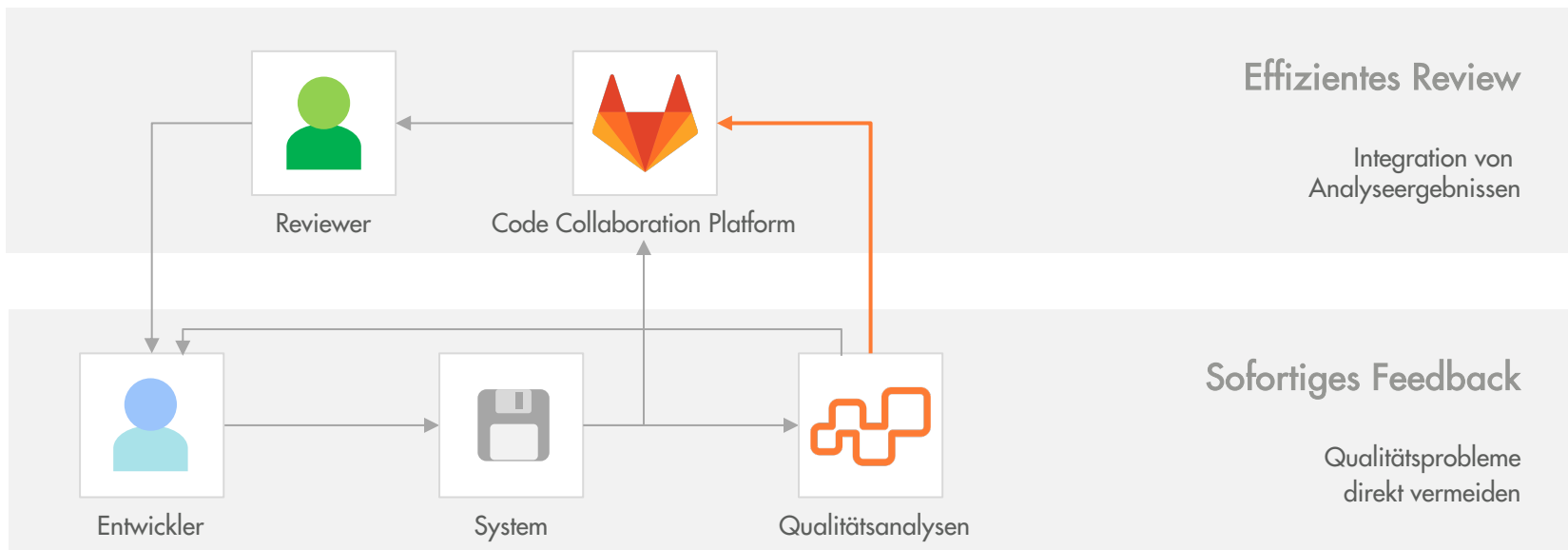
Codemenge (Zeilen)

Anzahl Findings

Lines of Code: 324,687
Number of Findings: 2,306



D. Steidl et al: *Continuous Software Quality Control in Practice*.
International Conference on Software Maintenance and Evolution (ICSME), 2014.





Team secondBranch → master

OPEN

Merge



Test PR



Overview Diff Commits



Show diff of



All changes in this pull request



Files



- src/main/java/org/jabref
 - cli
 - JabRefMain.java
 - logic
 - exporter
 - BibtexDatabaseWriter.java
 - TypedBibEntry.java**
 - Globals.java

src / main / java / org / jabref / logic / **TypedBibEntry.java** MODIFIED

Blame



```

13 13
14 14 /**
15 15  * Wrapper around a {@link BibEntry} offering methods for {@link BibDatabaseMode} dependend results
16 16  */
17 17 public class TypedBibEntry {
18 +
19 +     // TODO
    
```



Teamscale

⚠ TODO (view in Teamscale)

Reply · Edit · Delete · Create task · 3 days ago

```

18 20
19 21     private final BibEntry entry;
20 22     private final Optional<BibDatabase> database;
21 23     private final BibDatabaseMode mode;
24 +
25 +     public static volatile TestCoverageAnnotator testCoverageAnnotator;
    
```



Teamscale

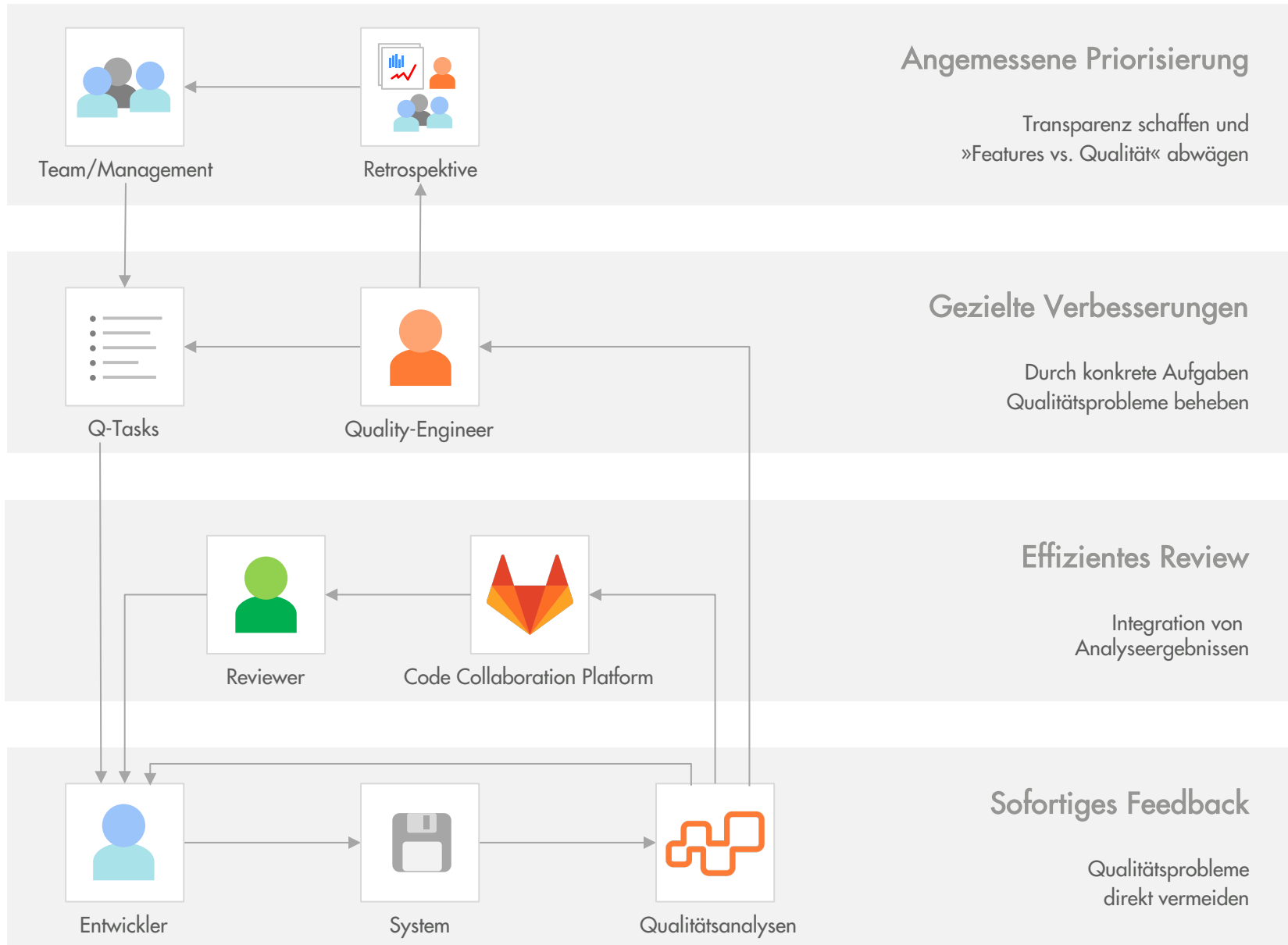
⚠ Interface comment missing (view in Teamscale)

⚠ Non-constant public attribute `testCoverageAnnotator` (view in Teamscale)

⚠ Static variable `testCoverageAnnotator` is not initialized (view in Teamscale)

Reply · Edit · Delete · Create task · 3 days ago

22 26



Code Quality in der Krankenversicherungs-IT der VKB

CQSE Software Intelligence Talk – 29. September 2021



-
- ▶ 1 Vorstellung

 - ▶ 2 Ausgangslage

 - ▶ 3 Initiative

 - ▶ 4 Awareness

 - ▶ 5 Ziele

 - ▶ 6 Prozesse

 - ▶ 7 Operationalisierung

 - ▶ 8 Fazit
-



Erkin Bapisch

erkin.bapisch@vkb.de

BLBV AG

Verschiedene Managementpositionen bei
verschiedenen Versicherungen unter anderem
bei der ERV, ERGO, Auxilia

Die Union Krankenversicherung – UKV und ihre Schwestergesellschaft Bayerische Beamtenkrankenkasse sind Teil des Konzerns Versicherungskammer, dem bundesweit größten öffentlichen Versicherer und siebtgrößtem Erstversicherer in Deutschland. Seine beiden Krankenversicherer UKV und Bayerische Beamtenkasse sind der Kranken- und Pflegeversicherer der Sparkassen-Finanzgruppe.

Mit über drei Millionen versicherten Personen bilden sie die drittgrößte private Krankenversicherungsgruppe in Deutschland.

-
- ▶ 1 Vorstellung

 - ▶ 2 Ausgangslage

 - ▶ 3 Initiative

 - ▶ 4 Awareness

 - ▶ 5 Ziele

 - ▶ 6 Prozesse

 - ▶ 7 Operationalisierung

 - ▶ 8 Fazit
-

Ausgangsposition

- Mainframe Cobol Anwendungslandschaft
- Große Module, eng Bindung zu anderen SW Komponenten
- 2-3 Releases im Jahr
- Kopfmonopole, hohes Durchschnittsalter
- Manuelle Testverfahren
- Hierarchische Organisation, Silos
- Wasserfallprojekte

Ziel

- Moderne Anwendungslandschaft
- Besser wartbare SW Module
- Viele Releases, CI/CD
- Schnelles Onboarding von SW Entwickler und Reduzierung der Risiken durch Kopfmonopole
- Automatisierte Testverfahren
- Serviceorganisation
- Agile Teams

-
- ▶ 1 Vorstellung

 - ▶ 2 Ausgangslage

 - ▶ 3 Initiative

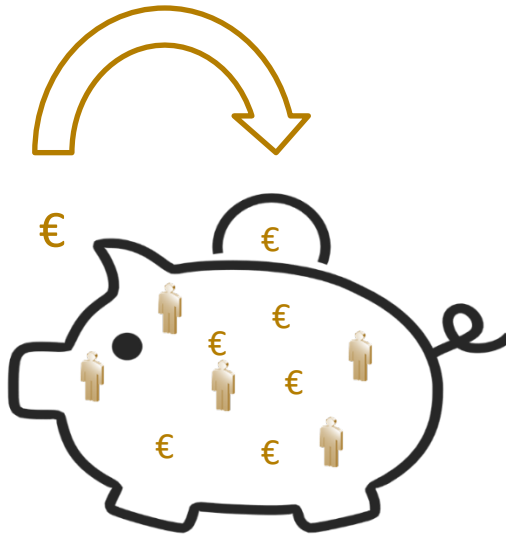
 - ▶ 4 Awareness

 - ▶ 5 Ziele

 - ▶ 6 Prozesse

 - ▶ 7 Operationalisierung

 - ▶ 8 Fazit
-



Top H7 Programm

- Diverse Themen zur Effizienzgewinnung
- Auswahl und Steuerung durch ein Gremium
- Invest für zu erwartende Effizienzgewinne wird bereitgestellt

Thema Code Quality Kranken

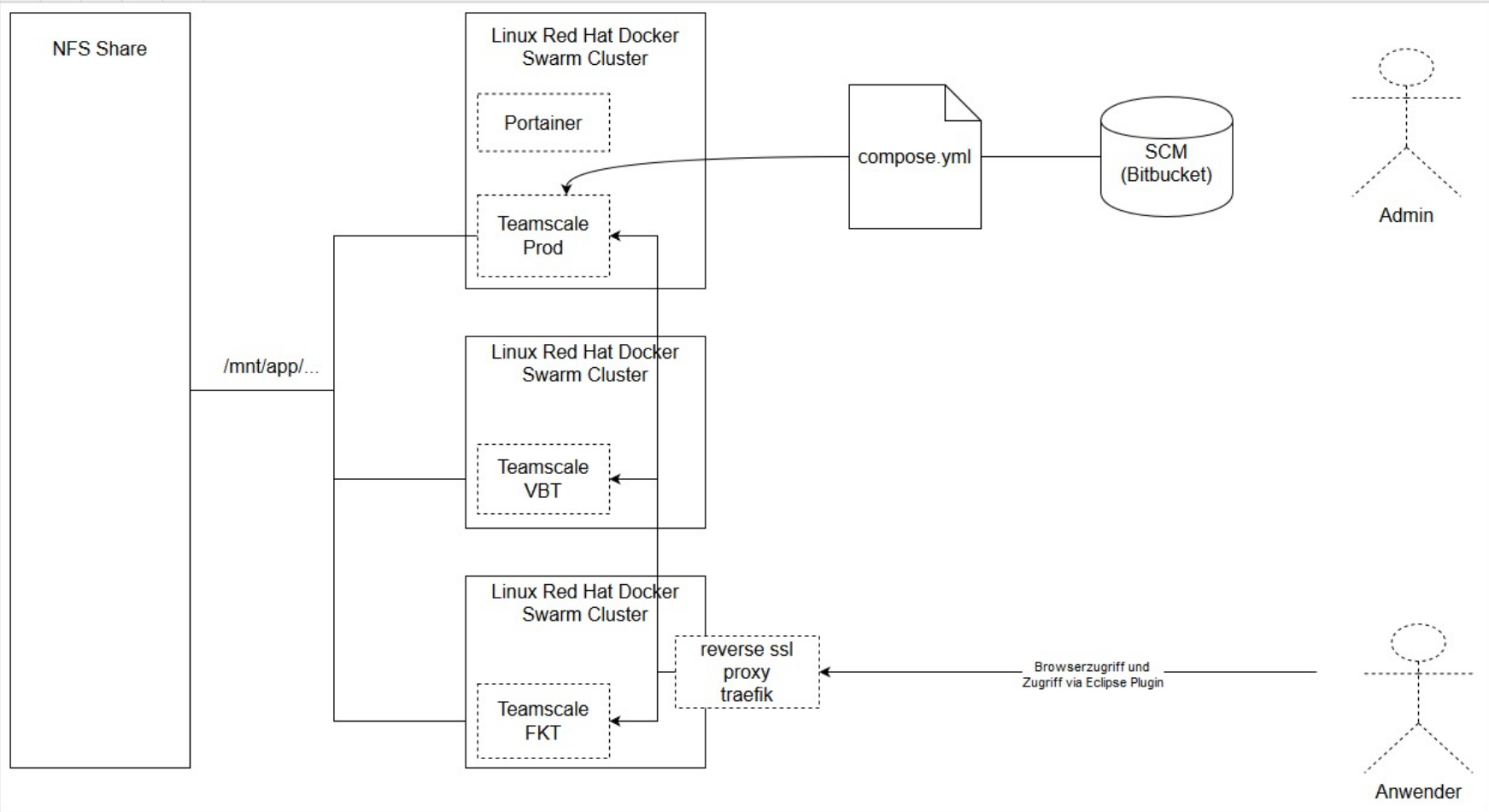
- MAK* Einsparung durch Reduzierung von Fehlerquellen und Verbesserung der Wartbarkeit erwartet
- Schätzung: ca. 3% der operativ eingesetzten Tage werden freigespielt
- Erstinvest durch TOP H7 Programm. Folgeinvest in Linie
- Auf Nachweis der Einsparung wird verzichtet
Auf MAK Einsparung wird nicht verzichtet 😊

Aufgaben:

- Erstellung einer Softwarelandkarte der relevanten Systeme und logischen und technischen Zusammenhänge
- Definition des Code Quality Prozesses (Rollen, Regeln, Schwellwerte, Qualitätsziele)
- Einrichtung der Analyseverfahren und Werkzeuge sowie Integration in den Entwicklungsprozess und die Entwicklungsinfrastruktur
- Definition und Verifikation von Kennzahlen zur Bewertung der Prozessergebnisse

Benötigte Qualifikationen:

- Nachgewiesene Erfahrung und Kompetenz in der Durchführung von Architekturreviews in komplexen Systemen
- Erprobtes Modell zur Steuerung der handwerklichen Qualität von Softwaresystemen
- Nachgewiesene Erfahrung in der Entwicklung und Betreuung komplexer Softwaresysteme
- IT-Hintergrund und Erfahrung in den eingesetzten Technologien
- Ausgeprägte Kommunikationsfähigkeiten und Durchsetzungsstärke
- Erfahrungen mit verteilten Entwicklerteams



-
- ▶ 1 Vorstellung

 - ▶ 2 Ausgangslage

 - ▶ 3 Initiative

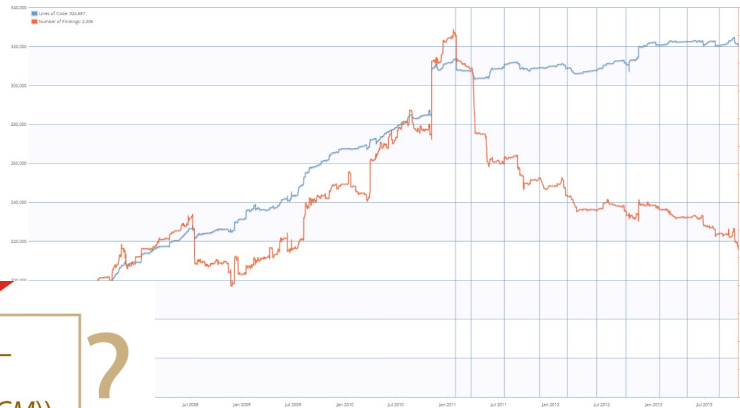
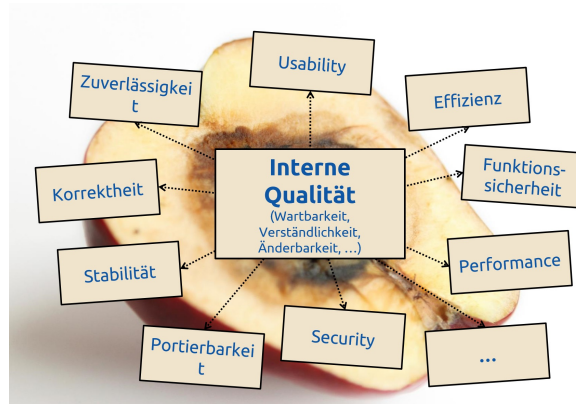
 - ▶ 4 Awareness

 - ▶ 5 Ziele

 - ▶ 6 Prozesse

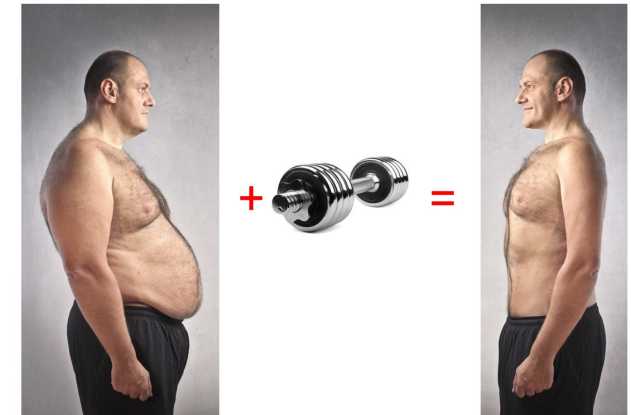
 - ▶ 7 Operationalisierung

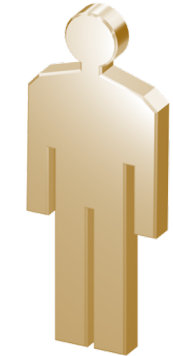
 - ▶ 8 Fazit
-



$$171 - 5.2 \cdot \ln(\text{avgHV}) - 0.23 \cdot \text{avgCC}(g) - 16.2 \cdot \ln(\text{avgLOC}) + 50 \cdot \sin(\sqrt{2.4 \cdot \text{perCM}})$$

HV: Halstead Volume CC: Cyclomatic Complexity
LOC: lines of code perCM: % Comment Lines





Entwickler



Tester



Steuernde Rollen
PO/SM/PM/RTE



IT-Management



Fachfunktionen

-
- ▶ 1 Vorstellung

 - ▶ 2 Ausgangslage

 - ▶ 3 Initiative

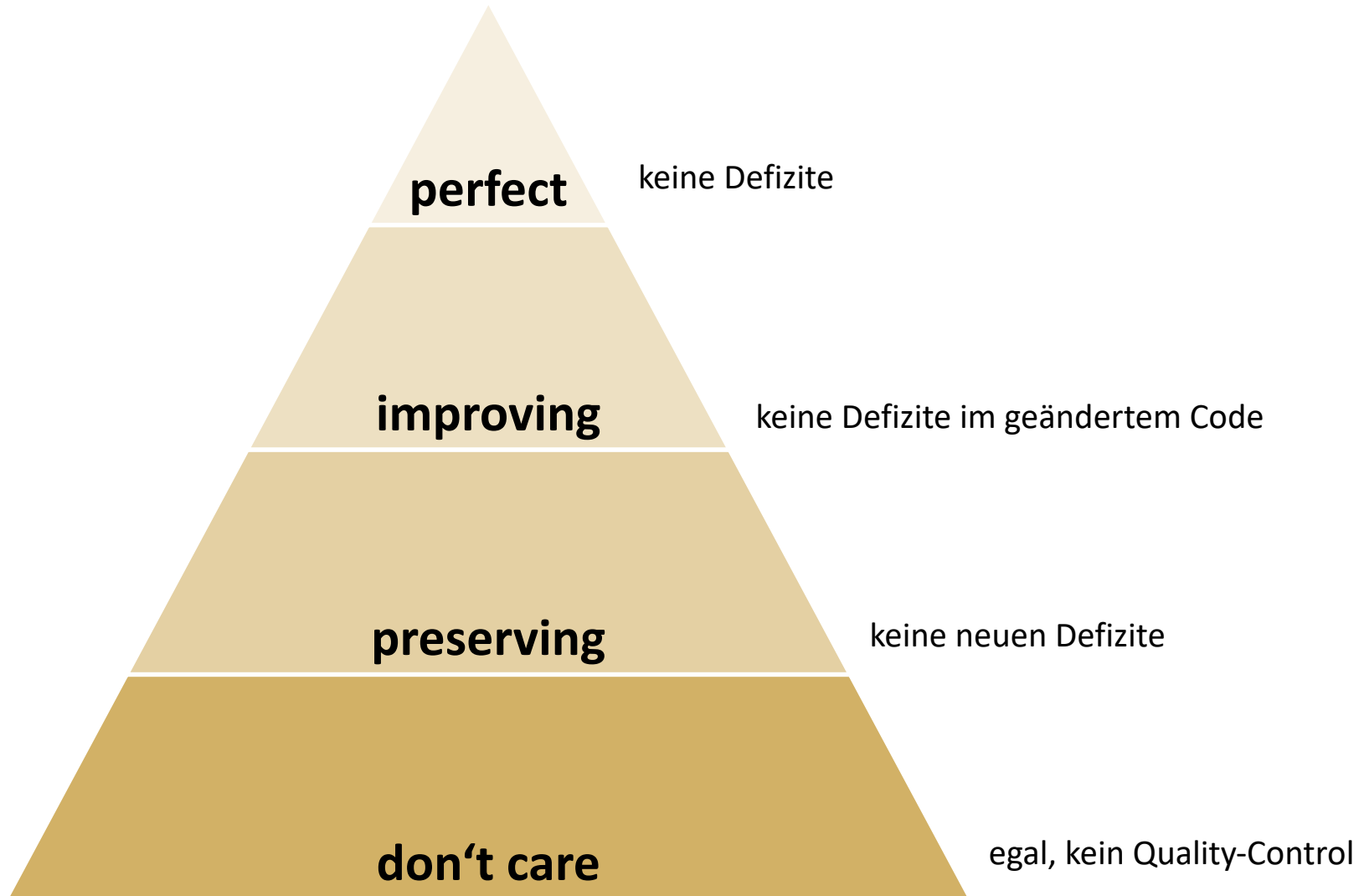
 - ▶ 4 Awareness

 - ▶ 5 Ziele

 - ▶ 6 Prozesse

 - ▶ 7 Operationalisierung

 - ▶ 8 Fazit
-



Beispiel für Qualitätsziele

von 2018/2019

CRP Neu (Java, Angular)

- Qualitätsziel „**perfect**“
- Keine Defizite

CRP Neu (KGxxxx)

- Qualitätsziel „**improving**“ in neuen Prozeduren und „**preserving**“ in übernommenen Code
- keine Defizite im geänderten Code (anhängig von Codeanteilen alt und neu)

BFS (KGxxxx)

- Qualitätsziel „**don't care**“

ORKAS Schnittstellen (Java)

- Qualitätsziel „**improving**“
- keine Defizite im geänderten Code

ORKAS Oberflächen (Java)

- Qualitätsziel „**improving**“ oder „**preserving**“
- keine Defizite im geänderten Code

wird im Team entschieden

ORKAS Rest (Java, C)

- Qualitätsziel „**preserving**“
- Keine neuen Defizite

-
- ▶ 1 Vorstellung

 - ▶ 2 Ausgangslage

 - ▶ 3 Initiative

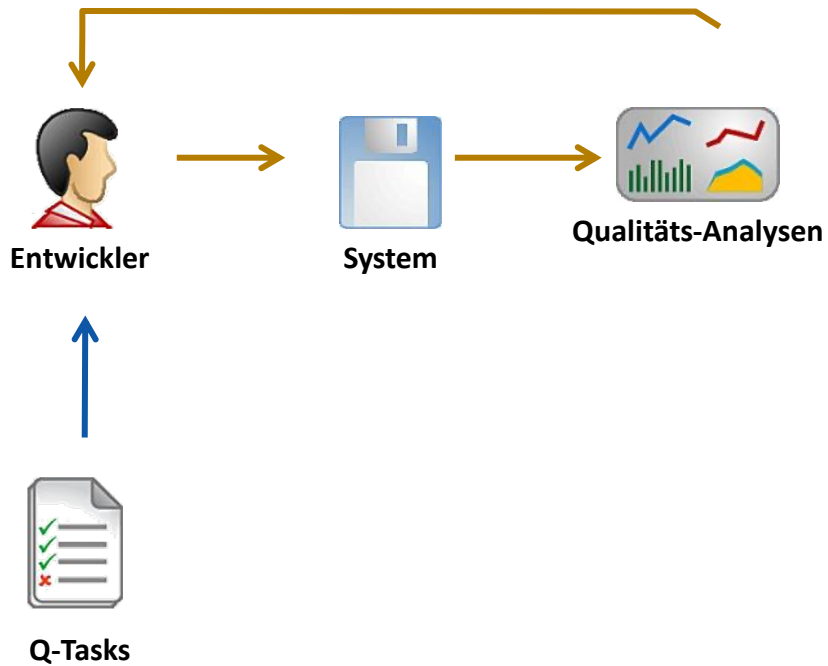
 - ▶ 4 Awareness

 - ▶ 5 Ziele

 - ▶ 6 Prozesse

 - ▶ 7 Operationalisierung

 - ▶ 8 Fazit
-

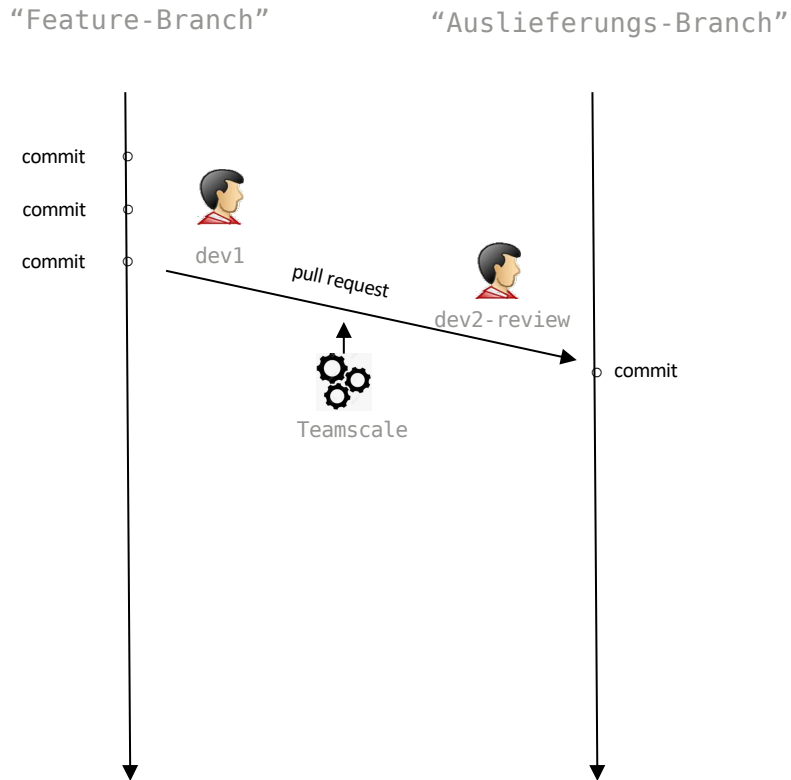


Unmittelbares Feedback aus Teamscale

Direkte Reaktion auf Qualitätsdefizite durch Entwickler

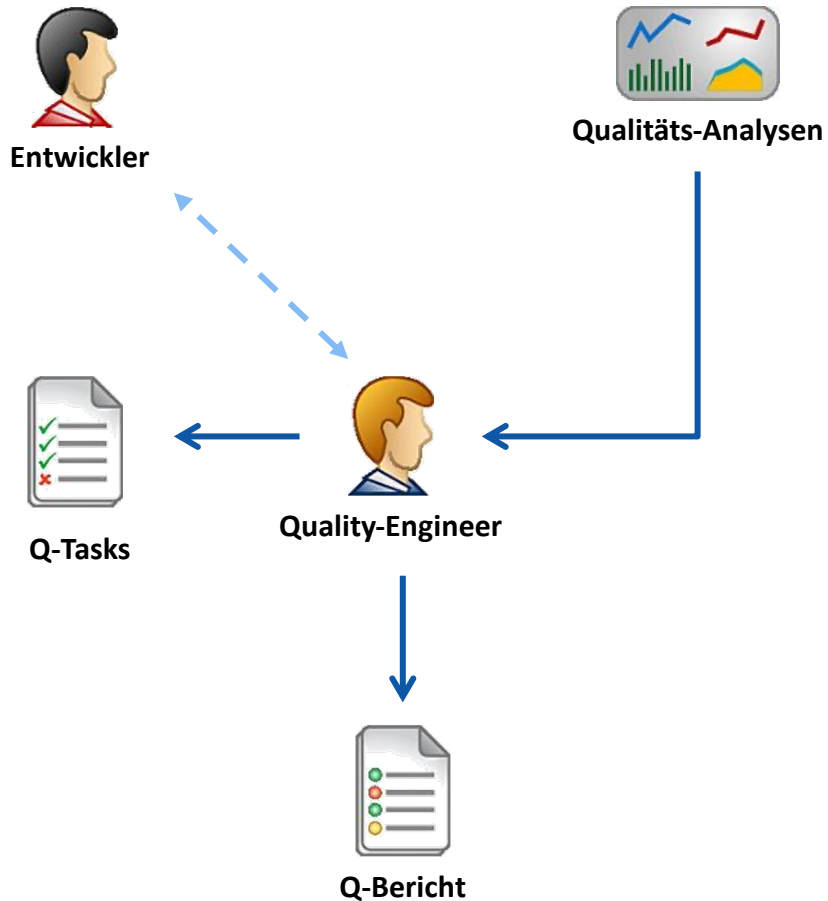
Qualitäts-Tasks aus Retrospektive

Umsetzung kleiner Tasks mit laufender Entwicklung



Feedback aus Teamscale bei Pull Requests

Unterstützung Reviewer zu Qualitätsdefiziten

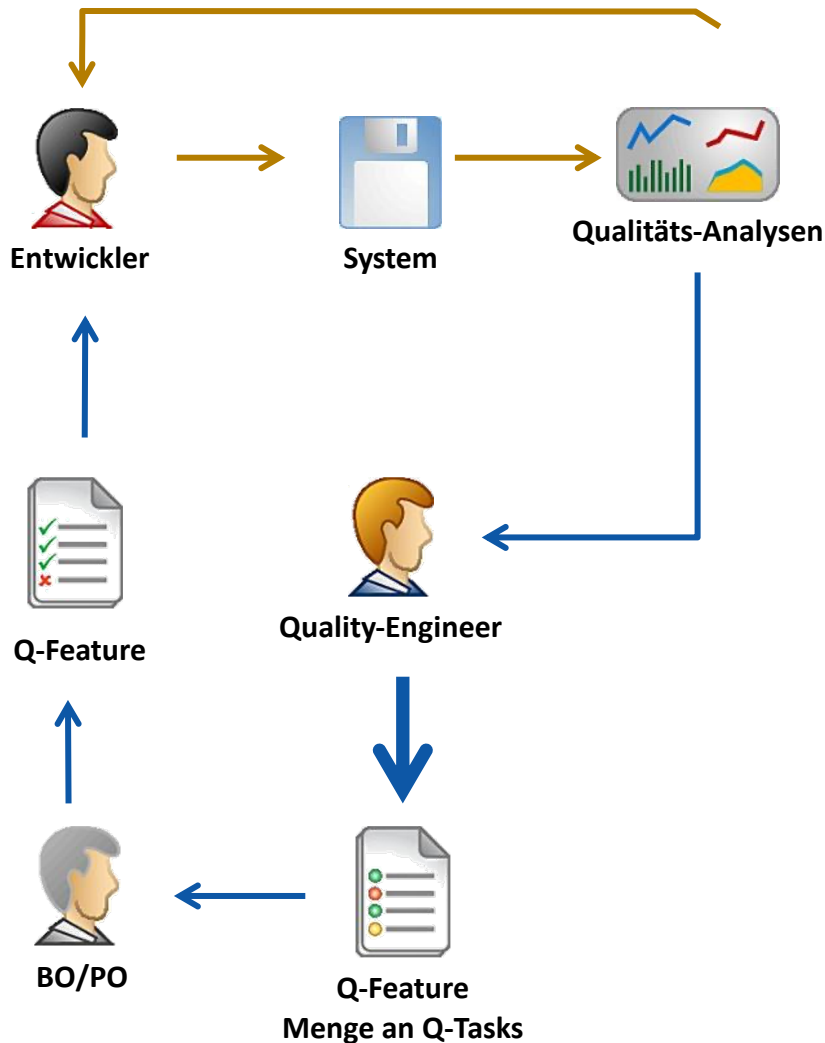


Regelmäßige Code-Retrospektive durch den Quality-Engineer (CQSE)

Identifikation von Qualitäts-Tasks

Tasks werden mit Entwicklern diskutiert

Knapper Qualitätsbericht zu Status und Trend



Qualitäts-Tasks aus Retrospektive

Erstellung Liste mit Q-Tasks durch Quality Engineer (CQSE)

Paketierung von Qualitäts-Features

Sammlung von Q-Tasks in einem Feature durch Team

Priorisierung von Qualitäts-Features

Entscheidungen und Priorisierung durch BO

Planung von Qualitäts-Features

Durch PO und Team im Sprint

Unmittelbares Feedback aus Teamscale

Direkte Reaktion auf Qualitätsdefizite durch Entwickler bei Umsetzung



Qualitäts-Analysen



Quality-Engineer

Erhöhte Transparenz

Qualitätsbericht aus Retrospektiven auch an BO/PO/SM/PM/RTE



Q-Reporting



Entwickler
BO/PO/SM
PM/RTE

-
- ▶ 1 Vorstellung

 - ▶ 2 Ausgangslage

 - ▶ 3 Initiative

 - ▶ 4 Awareness

 - ▶ 5 Ziele

 - ▶ 6 Prozesse

 - ▶ 7 **Operationalisierung**

 - ▶ 8 Fazit
-

Code Quality Prozess derzeit im Einsatz in folgenden Teams

- Kranken Leistung Team 1
- Kranken Leistung Team 3
- Kranken Leistung Team 6
- Kranken Betrieb Team 4
- Kranken Betrieb Team 5
- Projektteam NeoKranken

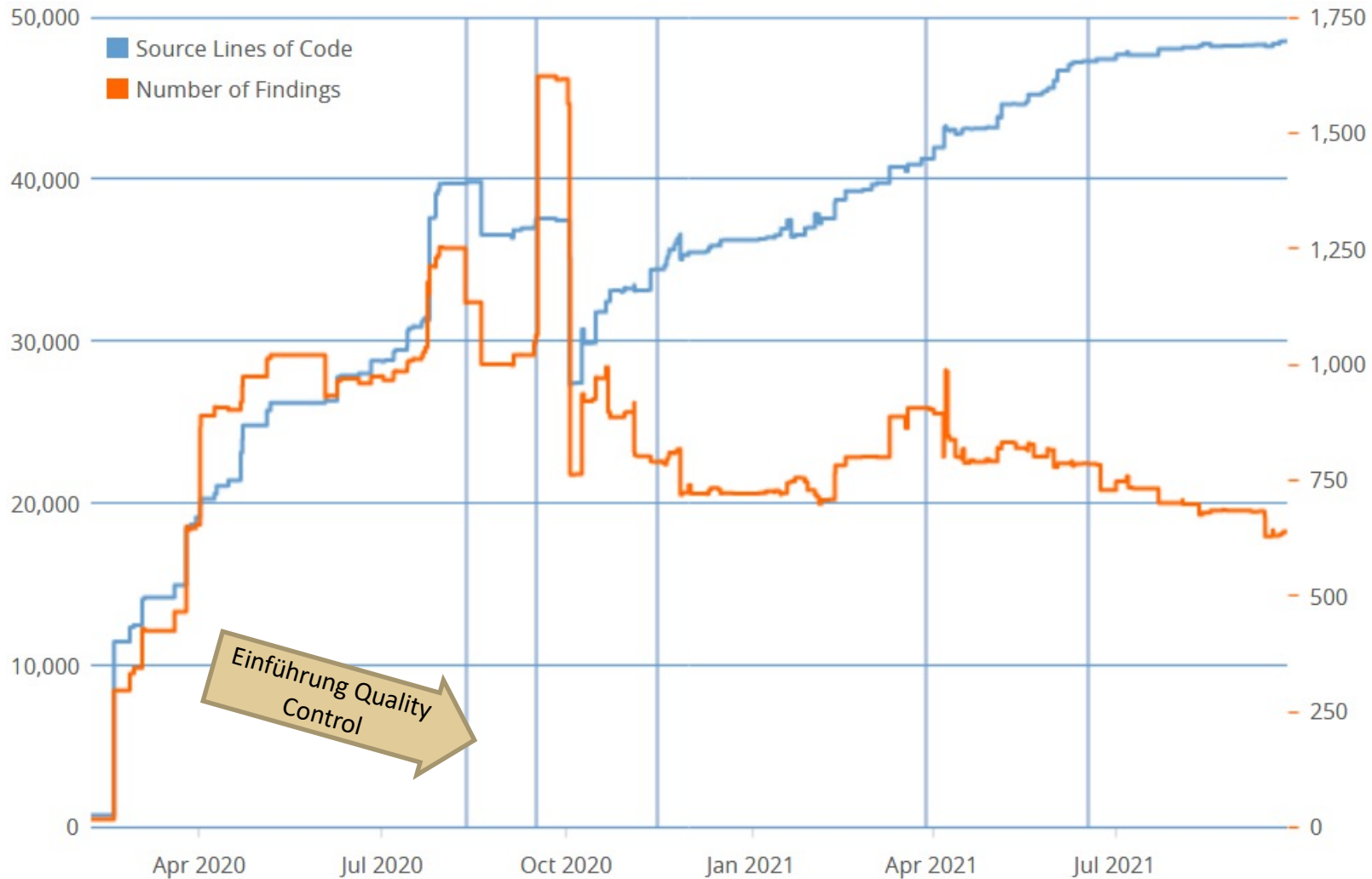
Code Quality Engineers von CQSE i.d.R. immer als Paar.

Administration Teamscale durch Systemteam Kranken mit dediziertem CQSE Support

- Zeitraum: 16.09. – 15.11.2020
- Teamscale Projekt abc
- Änderungen auf Branch master
- Git Repositories:

Repository	Include Pattern
abc/xyz-service	** .java, ** .js, ** .ts, ** .kt
abc/pruefen-framework	
abc/pruefen-komponente1	
abc/pruefen-komponente2	

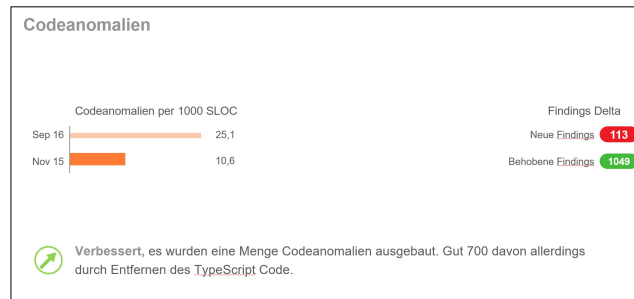
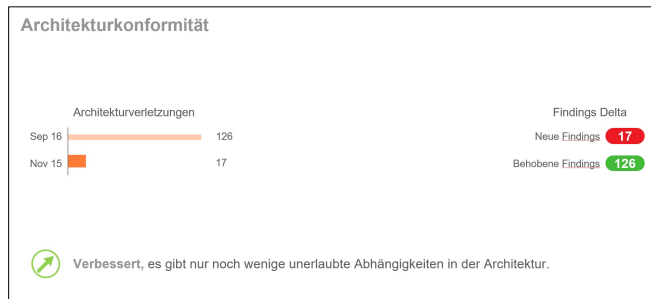
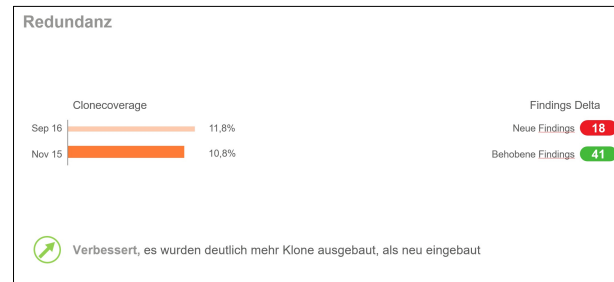
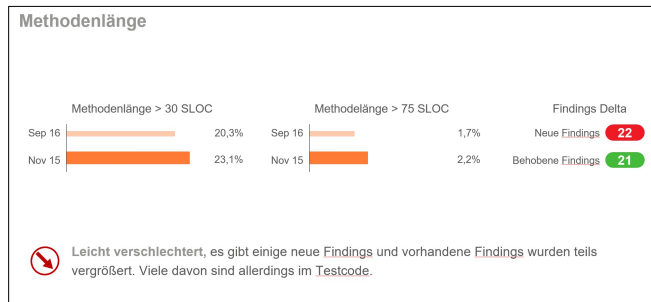
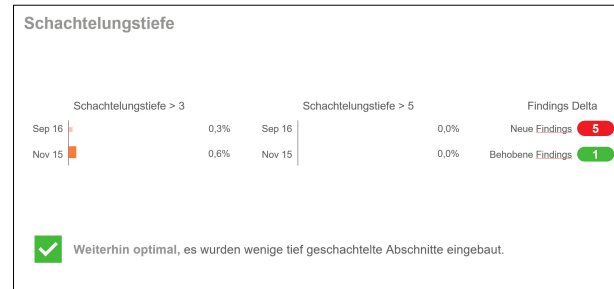
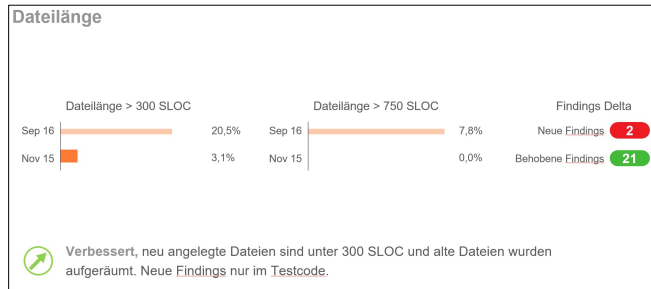
Findings Trend



Beispiel Überblick Qualitätsindikatoren

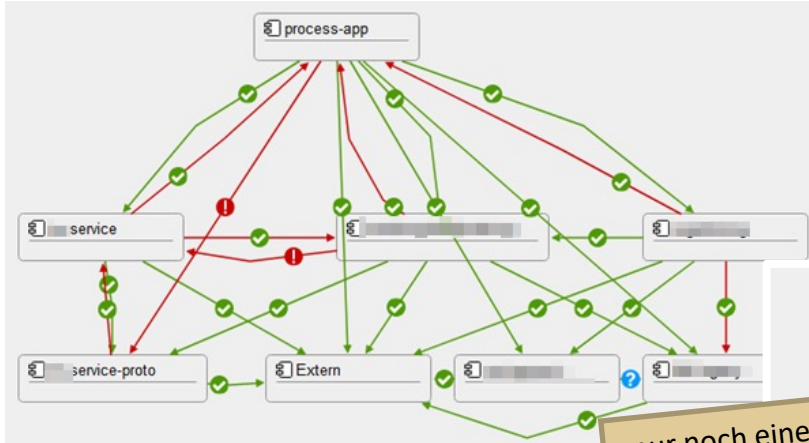
Dateilänge		verbessert
Methodenlänge		leicht verschlechtert
Schachtelungstiefe		weiterhin optimal
Redundanz		verbessert
Codeanomalien		verbessert
Architekturkonformität		verbessert

Details zu Qualitätsindikatoren

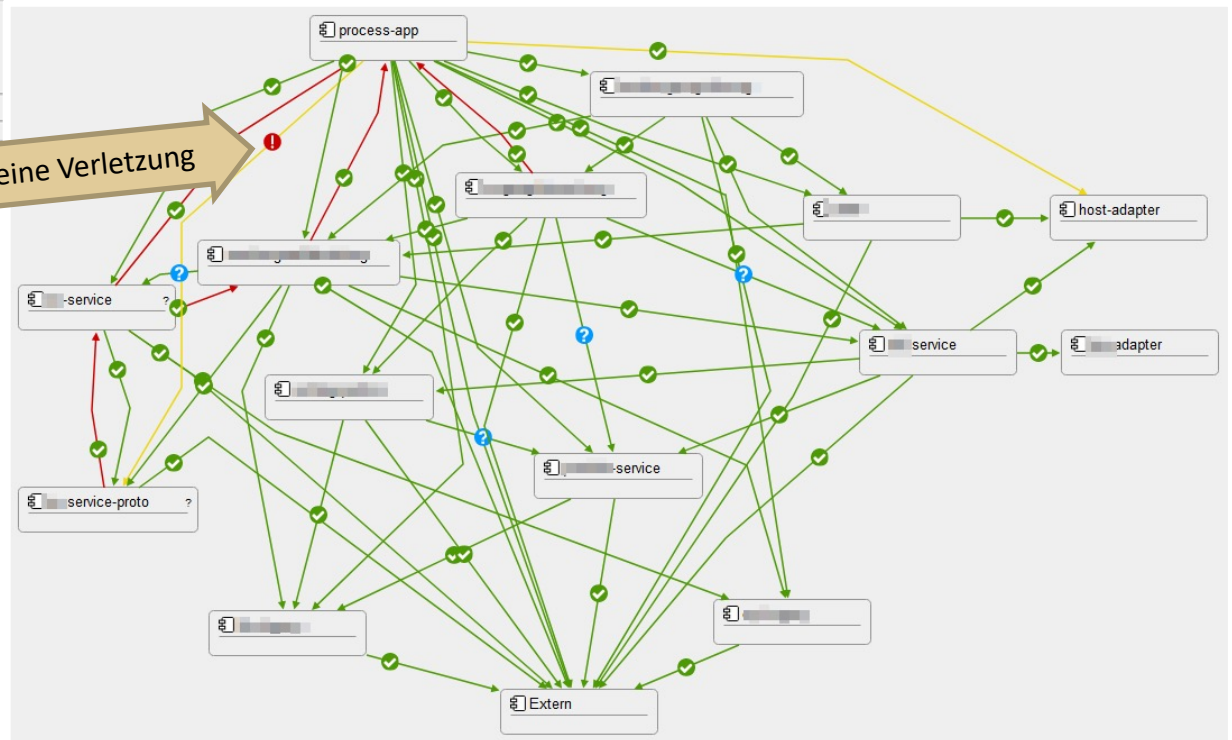


Architekturkonformität - abc

Architektur-Modell bei 1. Retrospektive



Architektur-Modell heute



Screenshot Q-Tasks

32 tasks selected

ID	Subject	Status	Author	Assignee	Created	Last updated	by	Resolution	Tags	Findings summary
2	Formatierung sollte einheitlich sein	Open	e019161	None	Nov 16 2020 17:48	Nov 16 2020 17:48	None	None	3. Retro, Formatierung	0 0 0 0 0
7	Redundanz in Mappingstrategies*.java	Open	e019161	None	Nov 17 2020 09:47	Nov 17 2020 16:14	None	None	Long Running, Redundanz, 1. Retro	0 0 1 0 0
15	Redundante Typdefinition	Open	szverlov	None	Mar 25 2021 15:56	Jun 21 2021 09:23	None	None	Redundanz, 2. Retro	9 0 0 0 0
16	Zwei identische Klassen Versicherungsnummer.java	Open	szverlov	None	Mar 25 2021 15:59	Mar 29 2021 10:19	None	None	2. Retro, Redundanz	2 0 0 0 0
19	"Magic Numbers" in 112 Klassen	Open	szverlov	None	Mar 25 2021 16:10	Jun 21 2021 08:51	None	None	2. Retro, Formatierung, Diskussion	1 2 2 54 0
20	Der Name Klassenproperties sollte nicht den Klassennamen erhalten	Open	szverlov	None	Mar 25 2021 16:14	Mar 29 2021 10:29	None	None	2. Retro, Codeanomalien	13 0 0 0 0
22	Exception Stacktrace geht verloren	Open	szverlov	None	Mar 25 2021 16:43	Mar 29 2021 10:37	None	None	2. Retro, Formatierung	1 2 0 0 0
23	Benennung der Variablen	Open	szverlov	None	Mar 25 2021 16:47	Mar 29 2021 10:39	None	None	2. Retro, Formatierung	5 1 0 0 0
24	Nicht adressierte TODOS in 19 Dateien	Open	szverlov	None	Mar 25 2021 16:54	Mar 29 2021 10:42	None	None	2. Retro, Task Tags	3 15 0 0 0
25	Nicht adressierte FIXMEs	Open	szverlov	None	Mar 25 2021 16:56	Mar 29 2021 10:45	None	None	2. Retro, Task Tags	4 0 0 0 0
26	Redundanz zwischen *Beleg-Klassen	Open	Christian Pfaller	None	Jun 21 2021 09:32	Jun 23 2021 11:01	None	None	3. Retro, Redundanz	6 1 0 0 0
27	Redundanz zwischen BelegMappingstrategie11.java und BelegMappingstrategieWilsfittel.java	Open	Christian Pfaller	None	Jun 21 2021 09:44	Jun 21 2021 09:44	None	None	3. Retro, Redundanz	2 0 0 0 0
28	Werfen von RuntimeException	Open	Christian Pfaller	None	Jun 21 2021 16:17	Jun 23 2021 11:02	None	None	3. Retro, Codeanomalien	0 7 0 0 0
29	Fangen von generischen Exceptions	Open	Christian Pfaller	None	Jun 21 2021 16:42	Jun 23 2021 11:03	None	None	3. Retro, Codeanomalien	7 1 0 2 0
30	Templärer Operator in SbiSoopIdexpter.java	Open	Christian Pfaller	None	Jun 21 2021 17:04	Jun 23 2021 11:03	None	None	3. Retro, Codeanomalien	0 1 0 0 0
31	Auskommentierter Code in VorgangstrennungService.java und SbiSoopIdexpter.java	Open	Christian Pfaller	None	Jun 21 2021 17:11	Jun 22 2021 15:25	None	None	3. Retro, Codeanomalien	0 3 0 0 0
32	Leere Blöcke	Open	Christian Pfaller	None	Jun 21 2021 17:16	Jun 23 2021 11:04	None	None	3. Retro, Codeanomalien	2 1 0 0 0
33	Deklaration von events sollte näher an Verwendung in VorgangsschlusselungService.java sein	Open	Christian Pfaller	None	Jun 21 2021 17:20	Jun 23 2021 11:04	None	None	3. Retro, Codeanomalien	1 0 0 0 0
34	Leere Datei tabBelegRepository.java	Open	Christian Pfaller	None	Jun 21 2021 17:21	Jan 01 1970 01:00	None	None	3. Retro, Codeanomalien	1 0 0 0 0
35	*-Imports in BelegMappingSourceHeaderOrg.java	Open	Christian Pfaller	None	Jun 21 2021 17:35	Jun 23 2021 11:05	None	None	3. Retro, Codeanomalien	2 0 0 0 0

-
- ▶ 1 Vorstellung

 - ▶ 2 Ausgangslage

 - ▶ 3 Initiative

 - ▶ 4 Awareness

 - ▶ 5 Ziele

 - ▶ 6 Prozesse

 - ▶ 7 Operationalisierung

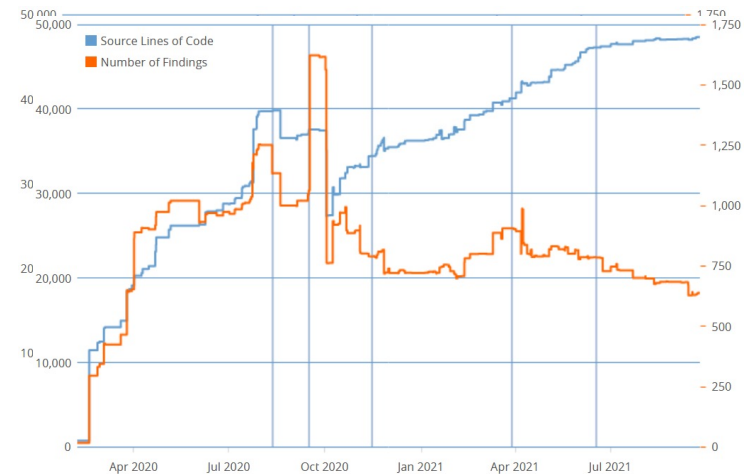
 - ▶ 8 Fazit
-

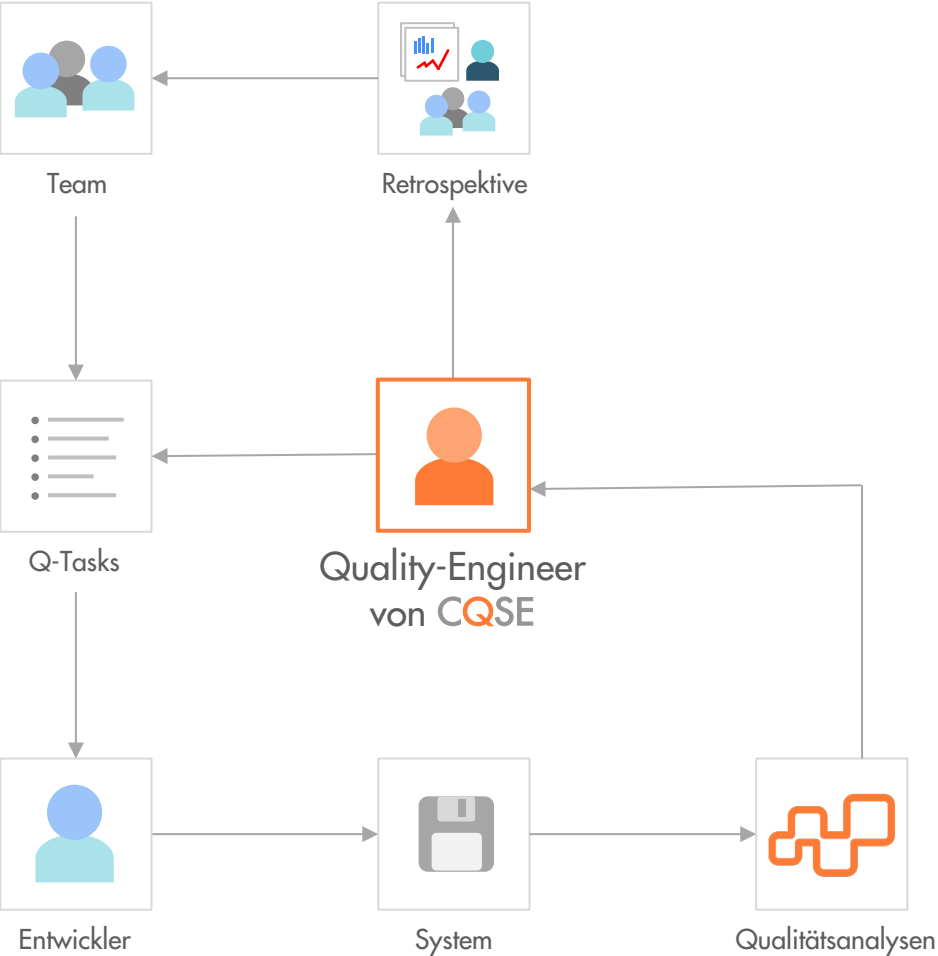
Fazit (nach fast 3 Jahren):

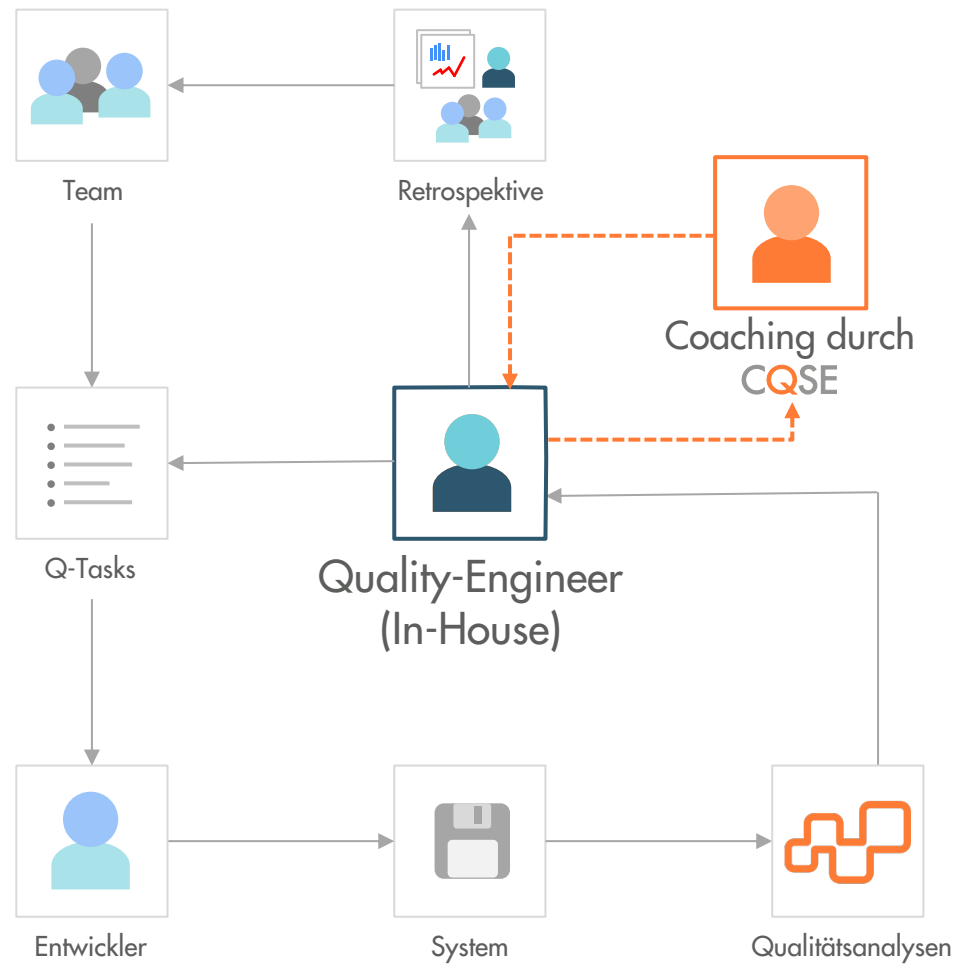
- Aller Anfang ist schwer
- Awareness, Awareness, Awareness ...
- Es braucht einen „Digital Evangelist“ für das Thema mit einem Mandat
- Business Case Nachbetrachtungen sind extrem schwer
- Positive Effekte in Findings zu LoC und Architekturverletzungen über die Zeit feststellbar.

Erfolgsfaktoren:

- Agile Teams
- Doppelspitze Intern/Extern
- Budget









Kontinuierliches Feedback



Analyseergebnisse interpretieren



Qualität kommunizieren

Vielen Dank für Ihre Teilnahme!

Feedback zum
Software Intelligence Talk 2021-3
»Quality Control einführen«



cqse.eu/si-talks/3/feedback

Anmeldung zum
CQSE Workshop
»Qualität unter der Lupe« - 27. Okt. 2021



cqse.eu/audit-workshop-2021/si