

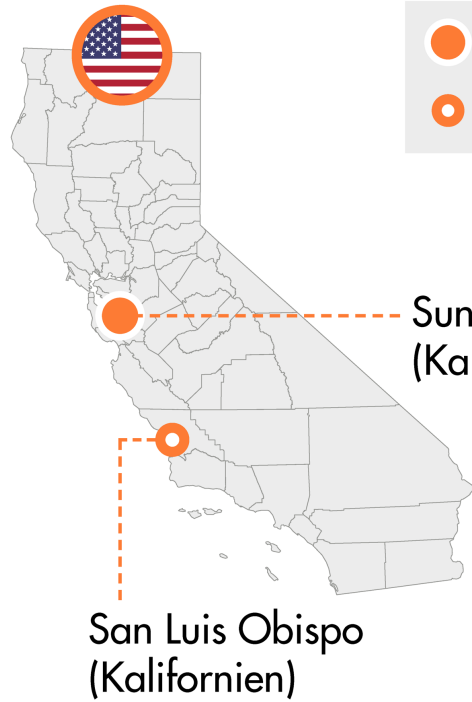
Continuous Quality Control

Qualität trotz immer kürzerer Releasezyklen

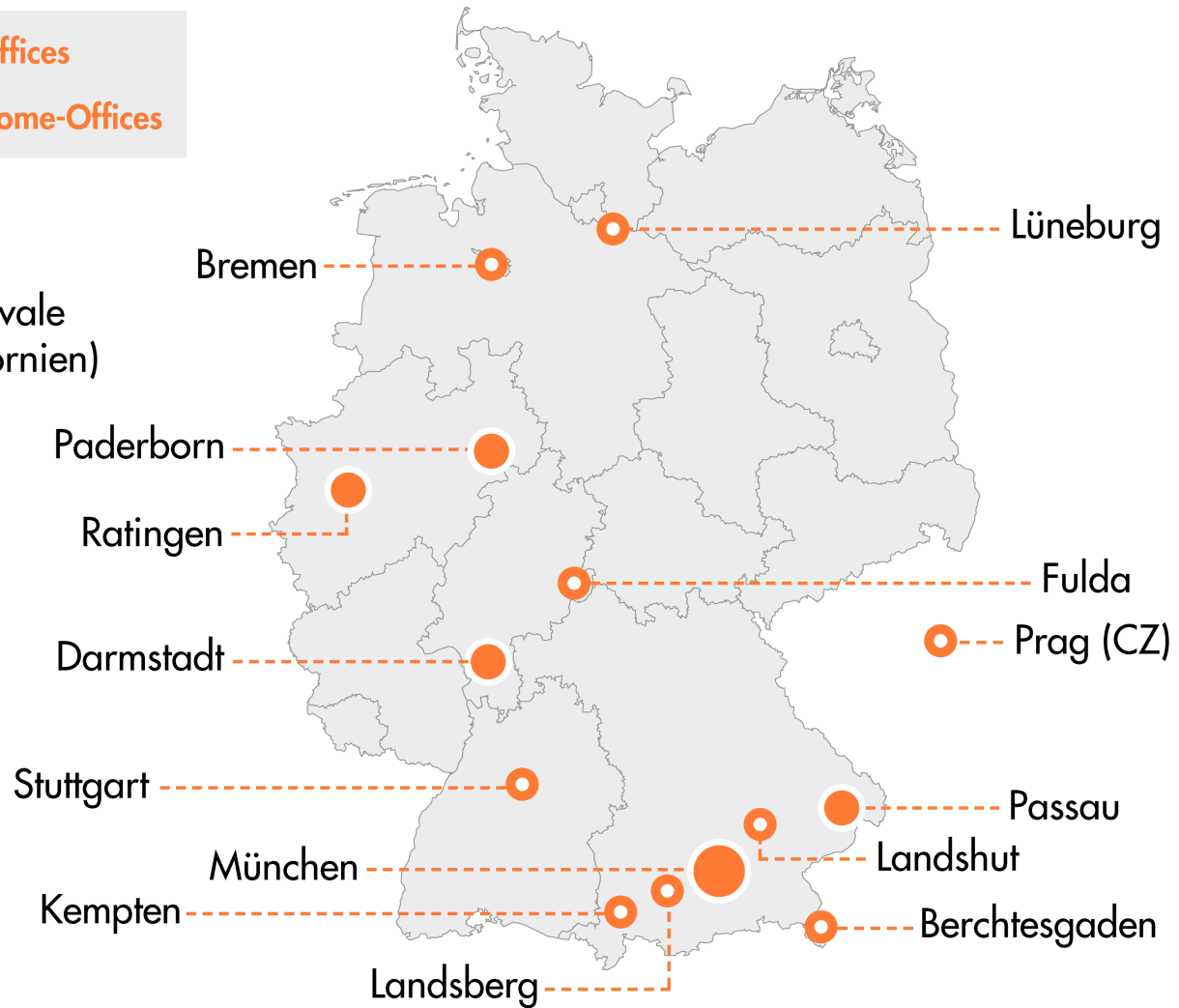
CQSE Workshop, 22. Juni 2023



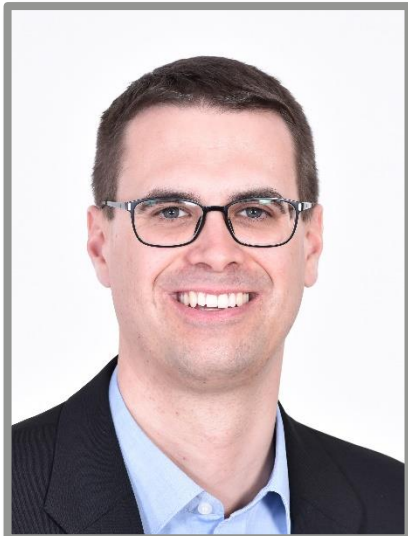
Dr. Tobias Röhm
Tobias Wiese



- Offices
- Home-Offices



Dr. Tobias Röhm



Tobias Wiese

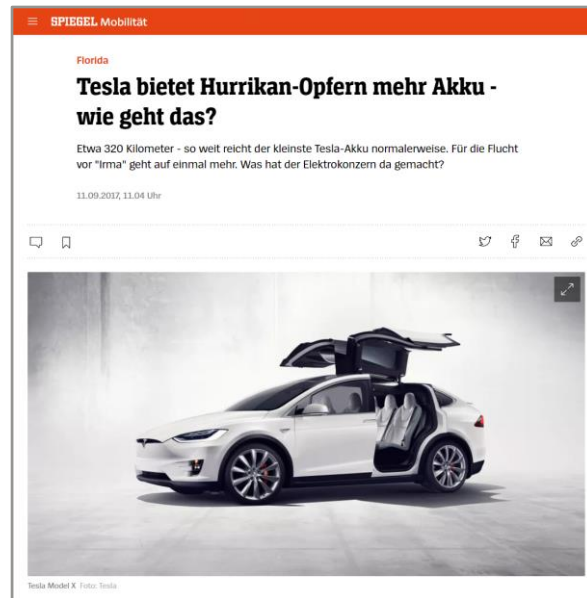


2015 - 2022



2022 - Heute

Kontext



Ereignisbasierte
Over-the-Air-Updates



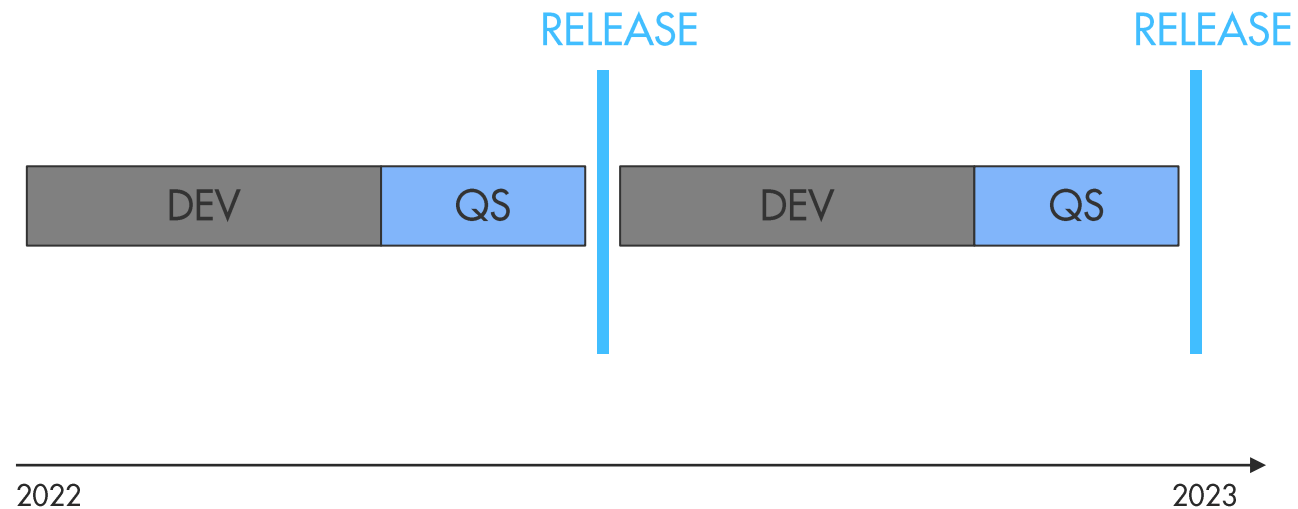
Viele, teilweise rückwirkende
Gesetzesänderungen



Zeitkritische Security-Patches

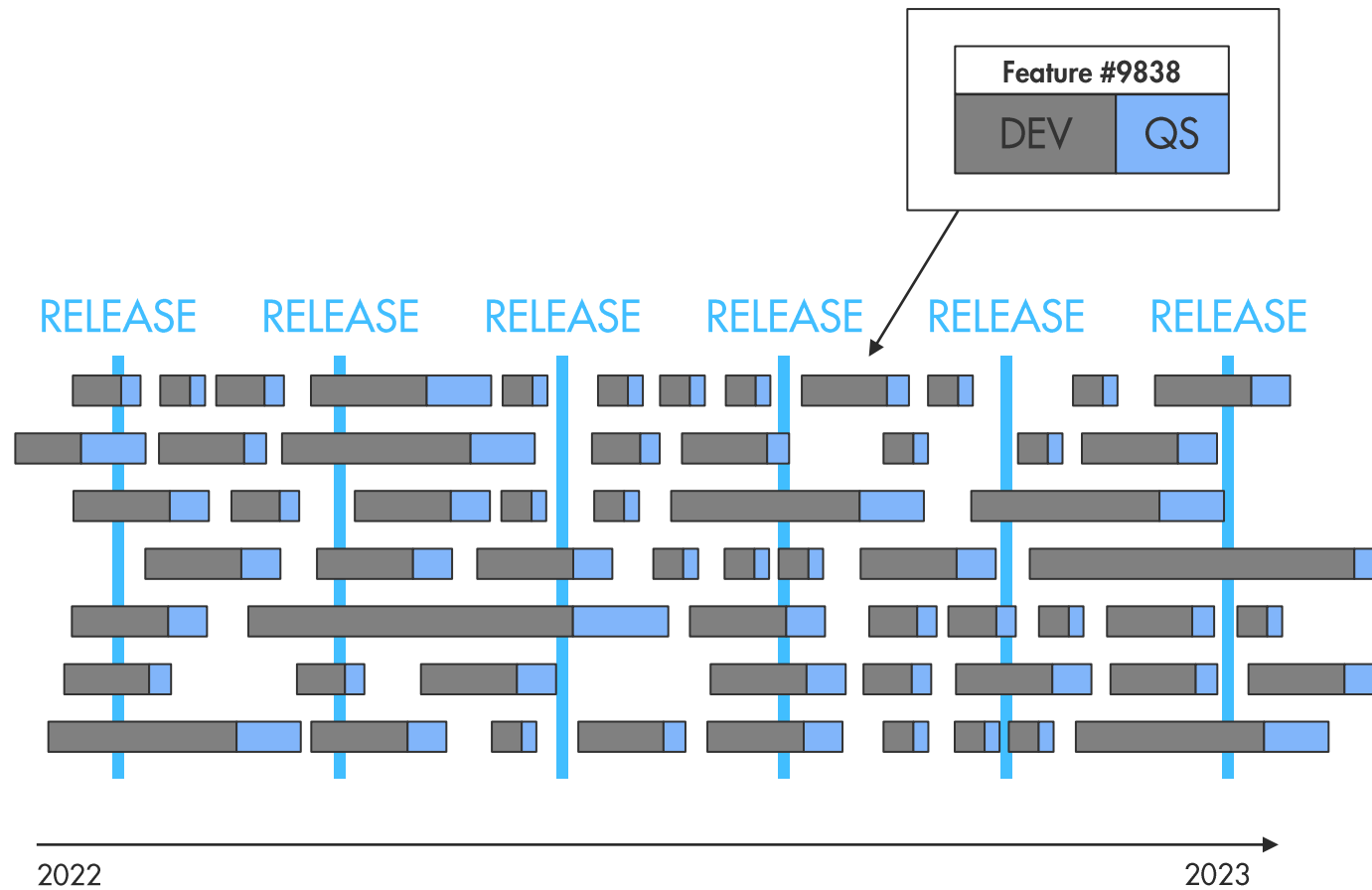
- Software-Updates müssen jederzeit möglich sein
- Software muss jederzeit ein releasefähiges Qualitätsniveau aufweisen

Problem: Traditionelle QS-Prozesse



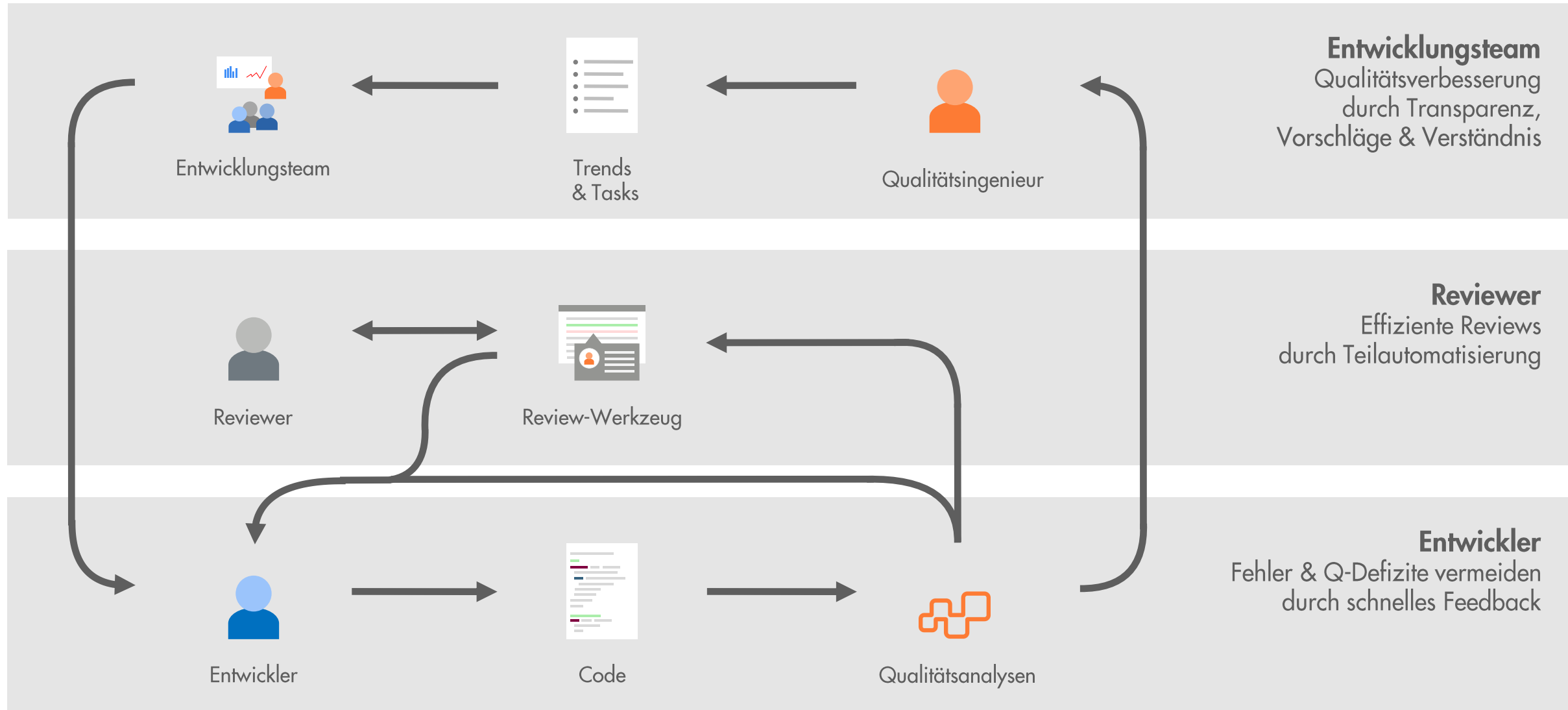
- Traditionelle QS-Prozesse sind (zu) langsam
- Software ist NICHT jederzeit auf einem releasefähiges Qualitätsniveau

Lösung: Continuous Quality Control (CQC)



→ CQC hält Software jederzeit auf hohem Qualitätsniveau

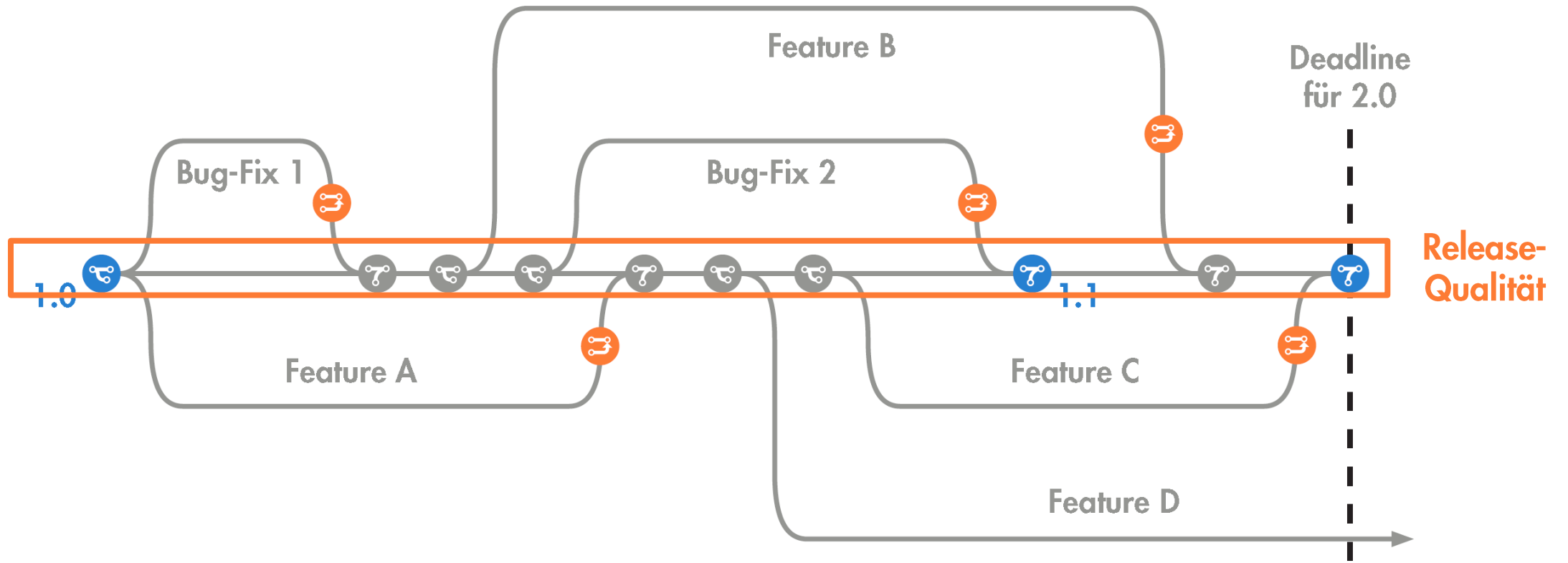
Continuous Quality Control



Voraussetzungen für Continuous Quality Control

Feature Branches in Git

Für SAP-Entwicklung lassen sich mit SAP-Transportwesen ähnliche Mechanismen bauen



→ Merge Request als »Qualitätstürchen«

Feedbackschleife 1:
Statische Codeanalyse hilft Entwicklern,
Fehler & Q-Defizite zu vermeiden

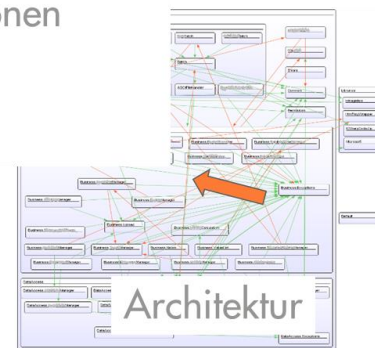
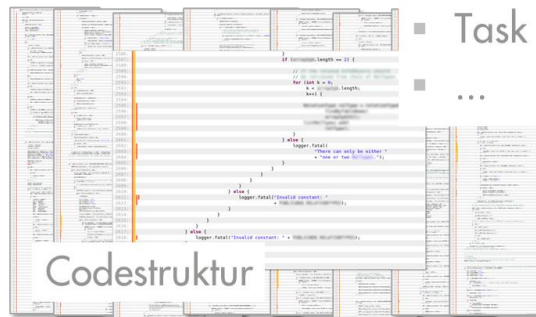
Feedbackschleife 1: Statische Codeanalyse hilft Entwicklern, Fehler & Q-Defizite zu vermeiden



Übersicht Statische Codeanalyse

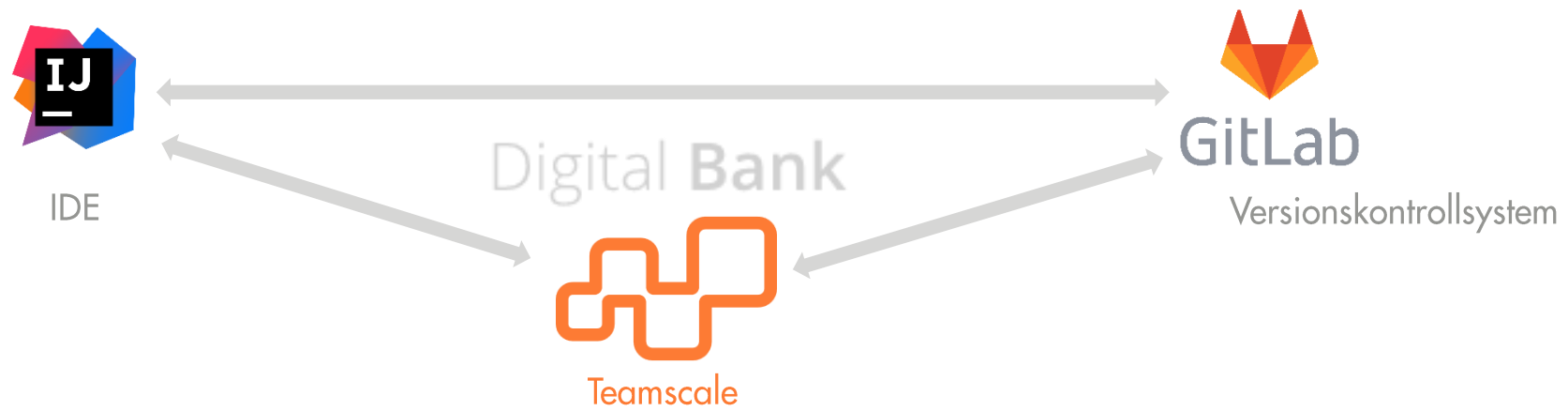


- Kommenturvollständigkeit
- Richtlinienverletzungen
- Auskommentierter Code
- Namenskonventionen
- Task Tags
- ...



→ Statische Codeanalyse schafft Transparenz bezüglich Code- & Architekturqualität

Demo: Statische Codeanalyse



IDE-Integration & Precommit-Analyse

```
47  /** Equator used for comparing elements. */
48  private final IEquator<? super T> equator;
49  // TODO generate a new finding for precommit analysis
50
51  /** Length of {@link #a}. */
52  private int n;
53
54  /** Length of {@link #b}. */
55  private int m;
56
57  /** The maximal possible difference between {@link #a} and {@link #b}. */
58  private final int max;
59
60  /**
61   * Maximal size of the delta produced. If the "real" delta would be larger, a
62   * truncated delta will be created.
63   */
64  private final int maxDeltaSize;
```

Findings	Category	Group	Lines
lib/org.conqat.lib.commons/src/oi			
Clone with 2 instances of length!	Code Dup...	Cloning	115-...
Clone with 2 instances of length!	Code Dup...	Cloning	395-...
Clone with 2 instances of length!	Code Dup...	Cloning	163-...
Clone with 2 instances of length!	Code Dup...	Cloning	85-92
TODO finding	Documen...	Task tags	49
Unrelated Member Comment	Documen...	Comment...	57
Unrelated Member Comment	Documen...	Comment...	311
Unrelated Member Comment	Documen...	Comment...	355
Unrelated Member Comment	Documen...	Comment...	350

Findings	Category	Group	Lines
lib/org.conqat.lib.commons/src			
TODO finding	Document...	Task tags	49

Pre-Commit Results for cqse-all @ CQSE: 1 finding added. No findings removed (3 minutes ago) 49:58 CRLF UTF-8 Tab* cr/20332_test_precommit

Precommit-Analyse ermöglicht Q-Feedback bereits vor VCS-Commit

Inkrementelle Analyse erlaubt schnelle Ergebnisse auch für komplexe Analysen

→ Feedback aus statischer Codeanalyse ist schnell, fokussiert und integriert nutzbar

Findings-Churn & Qualitätsziele für gewachsene Systeme



Share more code between dispatchSessionStorageEvents() and dispatchLocalStora...

Yesterday 17:34

by Chris Dumez as revision ffa7b160 in main (github)

Files: 1 changed

Findings



CachedResourceLoader::allCachedSVGImages() reparses resource URLs unnecessari...

Yesterday 17:26

by Ch

Files:

Findi

Unterscheidung zwischen »neuen Findings« und »Findings in geändertem Code« erlaubt schrittweise Q-Verbesserung von gewachsenen Codebasen mit der Pfadfinderregel.



[WinCairo][WK2] animations/par... is timing out https://bug...

Yesterday 13:20

by Fujii Hironori as revision 2d29f... in main (github)

Findings

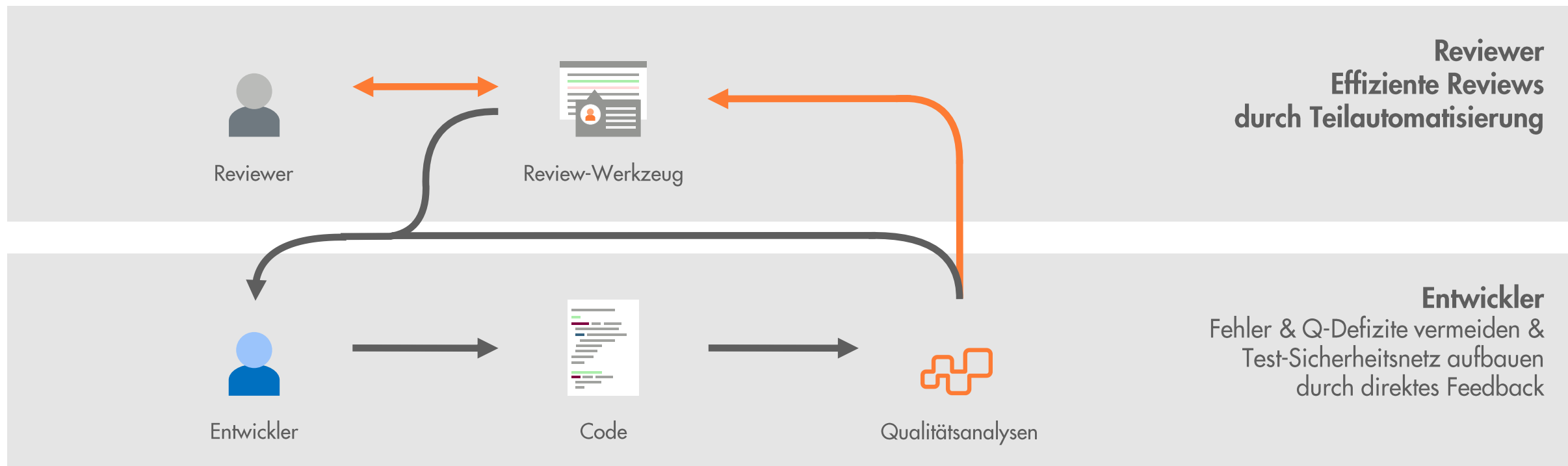


git-webkit Link to GitHub wiki https://bugs.webkit.org/show_bug.cgi?id=2370...

Yesterday 13:11

Feedbackschleife 2:
Statische Analyse und Test-Gap-Analyse
ermöglichen Reviewern effiziente Code Reviews

Feedbackschleife 2: Stat. Analyse und Test-Gap-Analyse ermöglichen Reviewern effiziente Code Reviews



Annotation von Findings aus statischer Analyse an Merge/Pull Requests

The screenshot displays a pull request interface for a repository named "Modify the code #7". The pull request is initiated by "asteckermeier" and aims to merge 2 commits into the "Sydro" branch from a branch named "merge_request_tga_demo_5".

Navigation tabs include "Code", "Pull requests 2", "Projects", "Security", and "Insights". Below the navigation, there are indicators for "Conversation 0", "Commits 2", "Checks 1", and "Files changed 1".

A finding is listed with the title "Fix method not being called" and ID "ac2cbc2". The finding is associated with the Teamscale (SWE) tool.

The "Added Findings" section states: "This pull request would introduce 1 new finding".

The "DETAILS" section shows the Teamscale logo and a finding summary: "Findings: 1 2". This summary is highlighted with an orange box. Below it, the finding text is: "WhiteBoard/WBProcessManager.m#L115 (AST) We should switch this to always log." with a link to "view in Teamscale".

The "ANNOTATIONS" section shows a warning icon and text: "Check warning on line 115 in WhiteBoard/WBProcessManager.m". Below this, the Teamscale logo and "teamscale-swe / teamscale-findings" are shown, followed by the file path "WhiteBoard/WBProcessManager.m#L115" and the finding text: "TODO (AST) We should switch this to always log." with a link to "https://teamscale.com/findings.html#details/sam-fac-plms-project-whiteboard?=merge_request_tga_d".

At the bottom, there is a link to "View more details on Teamscale (SWE)".

Annotation von Findings aus statischer Analyse an Merge/Pull Requests II

PR #4

[Open](#) nkxater wants to merge 22 commits into `master` from `firstBranch`

Conversation 0 | Commits 22 | Checks 1 | Files changed 3

Changes from all commits | File filter... | Jump to... | ⚙

▼ 4 ■ ■ ■ ■ jabref/JabRefGUI.java

```
@@ -35,7 +35,7 @@
35
36 public class JabRefGUI {
37
38 - private static final Logger LOGGER = LoggerFactory.getLogger(JabRefGUI.class);
38 + public static final Logger LOGGER = LoggerFactory.getLogger(JabRefGUI.class);
```

▼ 2 ■ ■ ■ ■ jabref/logic/bibtex/BibEntryWriter.java

```
@@ -163,7 +163,7 @@ private void writeField(BibEntry entry, Writer out, String name, int indentation
163 try {
164 out.write(fieldFormatter.format(field.get(), name));
165 out.write(',') + OS.NEWLINE);
166 - } catch (InvalidFieldValueException ex) {
166 + } catch (Exception ex) {
167 throw new IOException("Error in field '" + name + "': " + ex.getMessage(), ex);
168 }
```

169 }

⚠ Check warning on line 38 in jabref/JabRefGUI.java
Teamscale App / Teamscale Findings
jabref/JabRefGUI.java#L38
Interface comment missing

✖ Check failure on line 168 in jabref/logic/bibtex/BibEntryWriter.java
Teamscale App / Teamscale Findings
jabref/logic/bibtex/BibEntryWriter.java#L166-L168
Catch clause catches generic exception `Exception`



Test






Funktioniert Ticket-Implementierung?
Ging aus Versehen etwas kaputt?

Anwendung


Annotation von Testergebnissen an Merge/Pull Requests


Cr/20002 new report template


Edited 9 hours ago by build

 **Request to merge** `cr/20002_new_report_t...`  **into** `teamscale/v5.4.x` Open in Web IDE Check out branch 

The source branch is 31 commits behind the target branch

 **Pipeline #89447 failed for 040f1ce9 on cr/20002_new_report_t...** ✓ ✓ ✗ »

 Test summary contained 5 failed test results out of 4318 total tests Expand

 **Merge** Delete source branch Squash commits [?](#)

> **8 commits** and **1 merge commit** will be added to `teamscale/v5.4.x`. [Modify merge commit](#)

You can merge this merge request manually using the [command line](#)



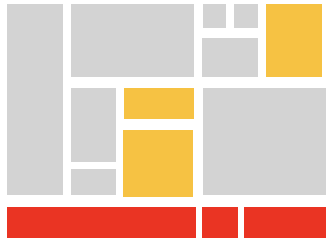
Test

Funktioniert Ticket-Implementierung?
Ging aus Versehen etwas kaputt?

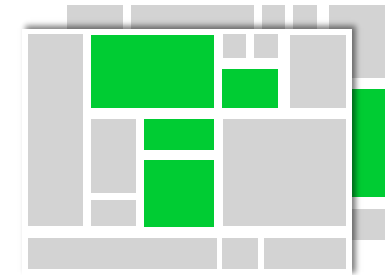
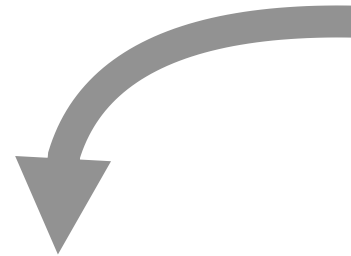
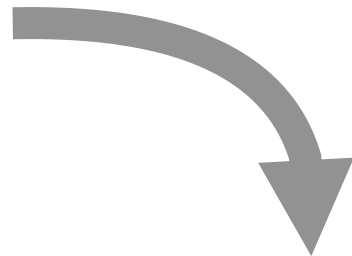
Wurden alle Codeänderungen getestet?

Anwendung


Test-Gap-Analyse

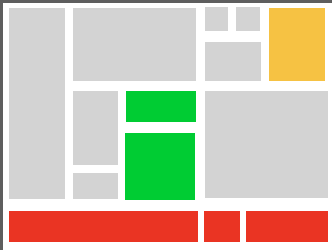


Codeänderungen



Testcoverage

 Teamscale



Test Gaps =
Ungetestete Codeänderungen

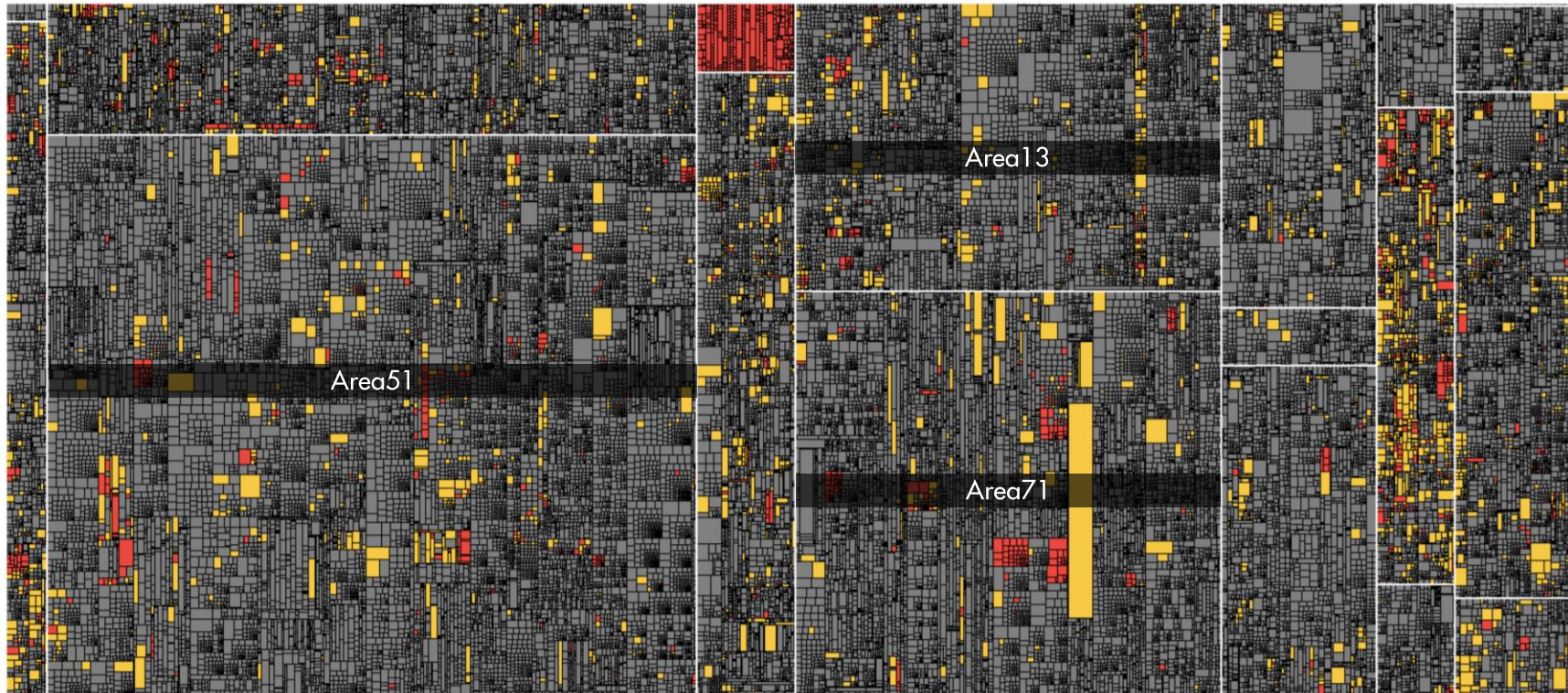
Test-Gap-Analyse identifiziert ungetestete Codeänderungen. Diese haben ein hohes Fehlerrisiko.

Did we test our changes? Assessing alignment between tests and development in practice. (Eder et al. 2013)

Codeänderungen

Churn

Jul 10 2022 - Aug 10 2022

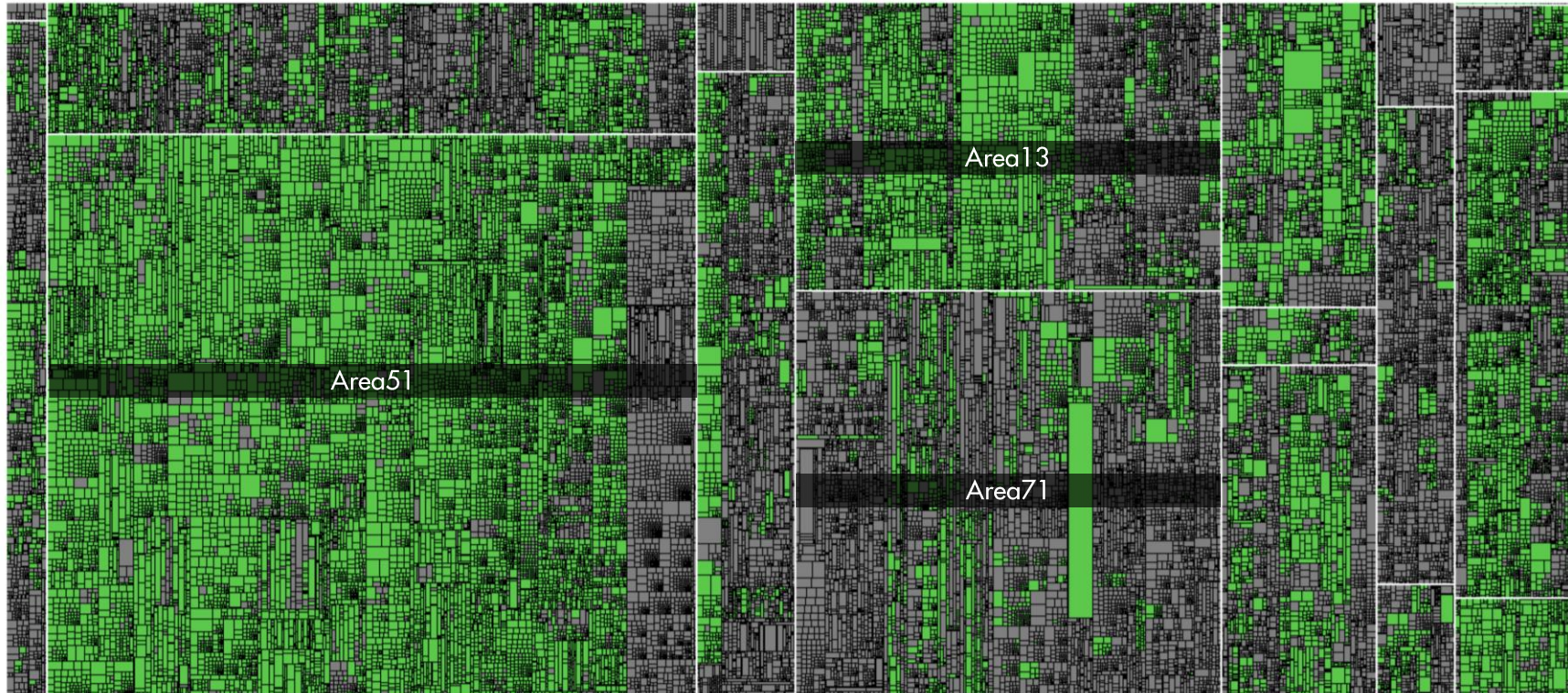


Methoden wurden...

- ... nicht geändert
- ... hinzugefügt
- ... geändert

Testcoverage

Jul 10 2022 - Aug 10 2022 | **Function Execution**
Execution Ratio: 43.79%

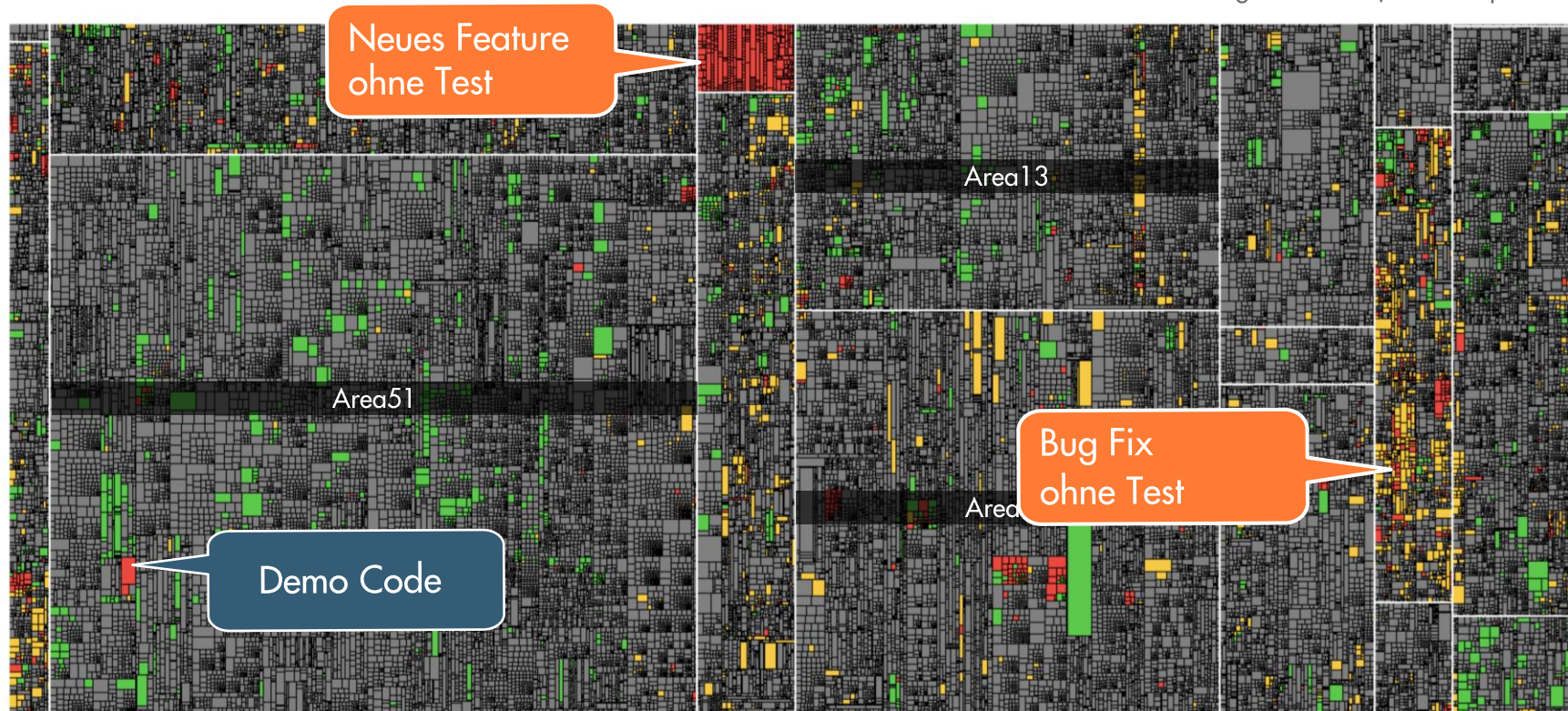


Methoden wurden...

- ... nicht getestet
- ... im Test ausgeführt

Test Gaps (= geänderter, ungetesteter Code)

Test Gaps
Jul 10 2022 - Aug 10 2022 | Test Gap: 63%



Methoden wurden...

- ... nicht geändert
- ... hinzugefügt & nicht getestet
- ... geändert & nicht getestet
- ... geändert & getestet

Annotation von Test Gaps an Merge/Pull Requests

Code Pull requests 2 Projects Security Insights

Modify the code #7

Open asteckermeier wants to merge 2 commits into Sydro from merge_request_tga_demo_5

Conversation 0 Commits 2 Checks 1 Files changed 1

Fix method not being called ac2cbc2

Teamscale (SWE)

teamscale-findings Resolve

Teamscale (SWE) / teamscale-findings
Started 4m 8s ago

Added Findings

This pull request would introduce 1 new finding

DETAILS

Teamscale Findings: 1 2 2 Test Gaps (67%)

WhiteBoard/WBProcessManager.m#L115: TODO (AST) We should switch this to always log. (view in Teamscale)

ANNOTATIONS

Check warning on line 115 in WhiteBoard/WBProcessManager.m

teamscale-swe / teamscale-findings

WhiteBoard/WBProcessManager.m#L115

TODO (AST) We should switch this to always log.
https://teamscale.c...com/findings.html#details/sam-fac-plms-project-whiteboard?merge_request_tga_d

View more details on Teamscale (SWE)

7 Modify the code

Request to merge merge_request_tga_demo_5 into Sydro OPEN NEGATIVE

This change contains 2 commits and introduces 1 new finding

Test Gaps

May 21 2022 12:22-Now | Test Gap: 67% Coverage sources SyWBjeasAT

WhiteBoard/WBProcessManager.m

Method log_demo_details

State Method was added after the baseline and was not tested after its latest modification

Findings 1 2 2

New findings

Message	Location	Finding Group
TODO More implementation needed	FrontBoard/FBProcessManager.m:136	Documentation > Task Tags > Comments/Task ...

Demo: Test-Gap-Analyse

Done Issue TS-15286 - Allow upload of backups to S3 storage

Creator: Andreas Sewe (on May 23 2018 16:00) Last update: May 29 2018 15:54
 Parent: Issue TS-14733

Assignee: Andreas Sewe

project	Type	Priority	Resolution	Fix Version	Component
TS	Feature	Normal	Green	Teamscale 4.3	Backend
Affected Version	Customer	Customer Issue	Epic Name	Freshdesk URL	

Merge Request	PDash Task	QA-Contact
https://git. . . eu/cqse/teamscale/merge_requests/2963/		kinnen

Description

This is part of TS-14733. Initial goal for 4.3 (as the feature freeze is soon) is to support special `{{s3}}` URI as *Path to backup.zip*.

Credentials can be stored either in the URI or (if possible) under *Admin > Settings > System Accounts*.

⌵ Affected files (15)

⚡ Test Gap Treemap

May 28 2018 13:40–now | Test Gap: 14.89% Coverage sources: **All**

- Test-Gap-Analyse schafft Transparenz über Test Gaps (= geänderten, ungetesteten Code)
- Test-Gap-Analyse unterstützt beim effizienten Aufbau von Test-Sicherheitsnetz

Code Review-Kommentar in GitLab

Discussion 1 Commits 2 Pipelines 2 Changes 1 Show all activity 1/1 thread resolved

Andreas Sewe @sewe started a thread on an old version of the diff 4 months ago
Resolved by Lars Heinemann 4 months ago [Toggle thread](#)

engine/org.conqat.engine.index/src/org/conqat/engine/index/repository/RepositoryContentUpdaterBase.java

```
293 293         List<BasicTokenElementInfo> tokenElements = tokenElementIndex.getValues(paths);
294 294         for (int i = 0; i < paths.size(); i++) {
295 295             String path = paths.get(i);
296 -             RepositoryChangeInfo changeInfo = changeSet.getChangeInfo(path);
297 -             ERepositoryChangeType changeType = patchChangeType(changeInfo,
298 -             tokenElements.get(i));
299 -             if (changeType != null) {
300 -                 result.add(new RepositoryChangeEntry(path,
301 -                 requiredNode.getRevision().getRevision(), changeType,
302 -                 commit, changeInfo.getOriginPath(),
303 -                 changeInfo.getOriginCommit()));
304 -             }
305 +             BasicTokenElementInfo element = tokenElements.get(i);
306 +             addChangeEntry(requiredNode, result, changeSet, path, element);
```

Andreas Sewe @sewe · 4 months ago Maintainer

I would re-order the message signature. ATM, it's hard to tell which parameter the change entry is added to. If you have `result` first, as in `result.add(...)` things become clearer.

Alternatively, let the extracted method return `Optional<RepositoryChangeEntry>` and do the `add` here.

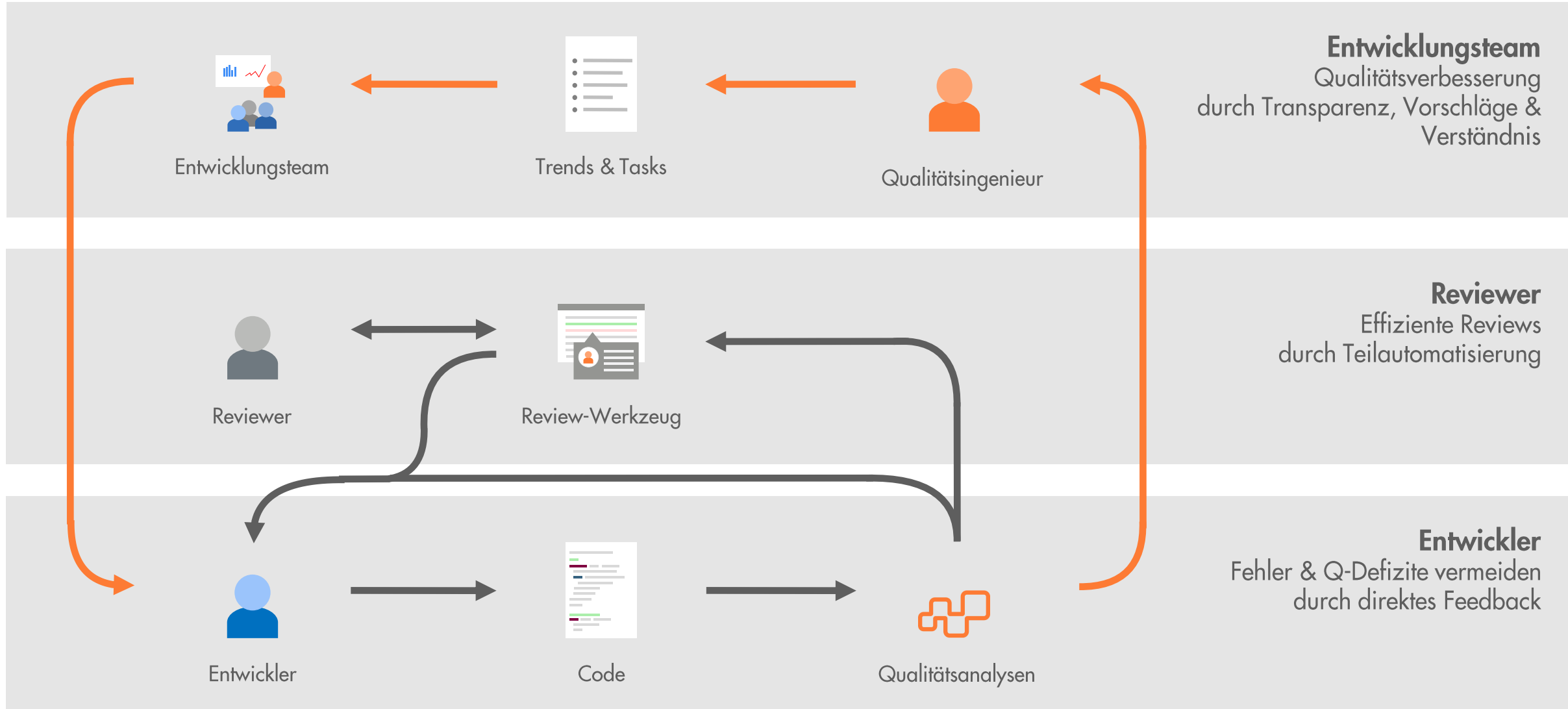
```
computeChangeEntry(requiredNode, changeSet, path, element).ifPresent(result::add);
```

Lars Heinemann @heinemann changed this line in version 2 of the diff 4 months ago

→ Hohe Eingangsqualität erlaubt effiziente, menschliche Code Reviews

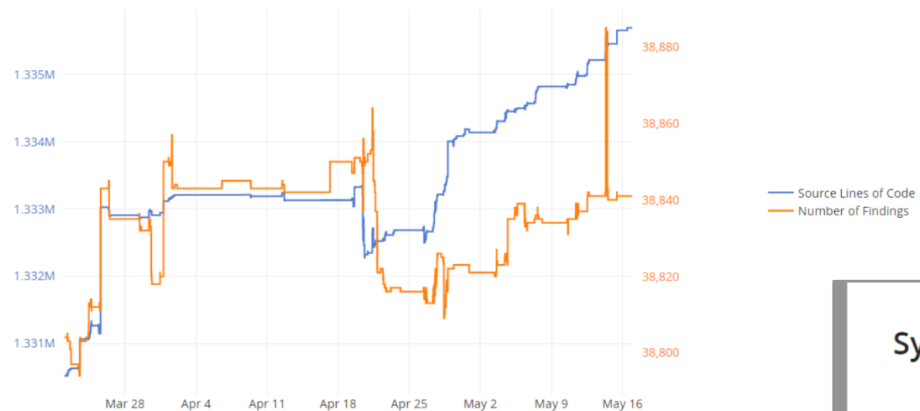
Feedbackschleife 3:
Q-Retrospektiven unterstützen Entwicklungsteams
bei Qualitätssicherung & -verbesserung

Feedbackschleife 3: Q-Retrospektiven unterstützen Entwicklungsteams bei Qualitätssicherung & -verbesserung



Demo: Q-Retrospektiven

Trend 22.März - 17. Mai



Ca. +5000 Zeilen (SLOC) Code, mehr neue Findings. Sprung am 14.05.: Code in DG_APLOUT_TO_SCE kopiert, welcher wieder auskommentiert wurde.

Findings-Delta
Findings ❌ 213 ✅ 176

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	→		

→ Trendanalyse schafft Transparenz für Team und Management

Task 34 - Redundanz zwischen UI5_TRANSLATION_TOOL und TRANSLATION_TOOL

created by Sep 16 2021 15:22, last updated Sep 17 2021 17:19

Assignee

Status

Resolution

Description
Der Code im neuen Paket UI5_TRANSLATION_TOOL ist oftmals redundant zum Code in TRANSLATION_TOOL. Da beide Pakete wohl ähnliche Logik implementieren, sollte die Redundanz hier reduziert werden, z. B. durch verwenden von Basisklassen.

Tags 3. Retro Redundanz

[Edit Task](#)

^ Findings

0 open 17 resolved 0 blacklisted

- ✓ Clone with 2 instances of length 18 in TRANSLATION_TOOL/CLAS/ EL_TRANSLATION.abep:979
- ✓ Clone with 2 instances of length 17 in UI5_TRANSLATION_TOOL/CLAS/ EL_UI5_TRANSLATION.abep:1208

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

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

Assignee

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.

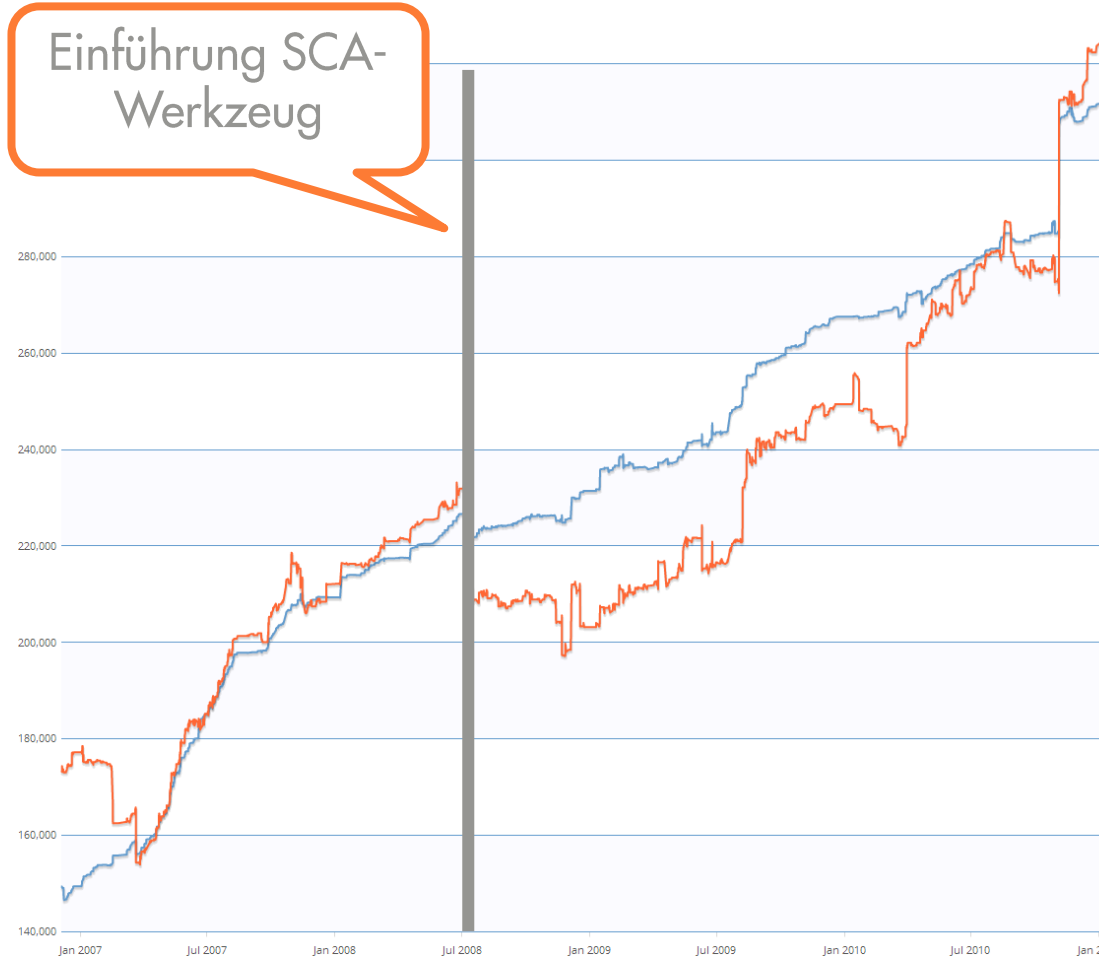
Tags 1. Retro Security

[Edit Task](#)

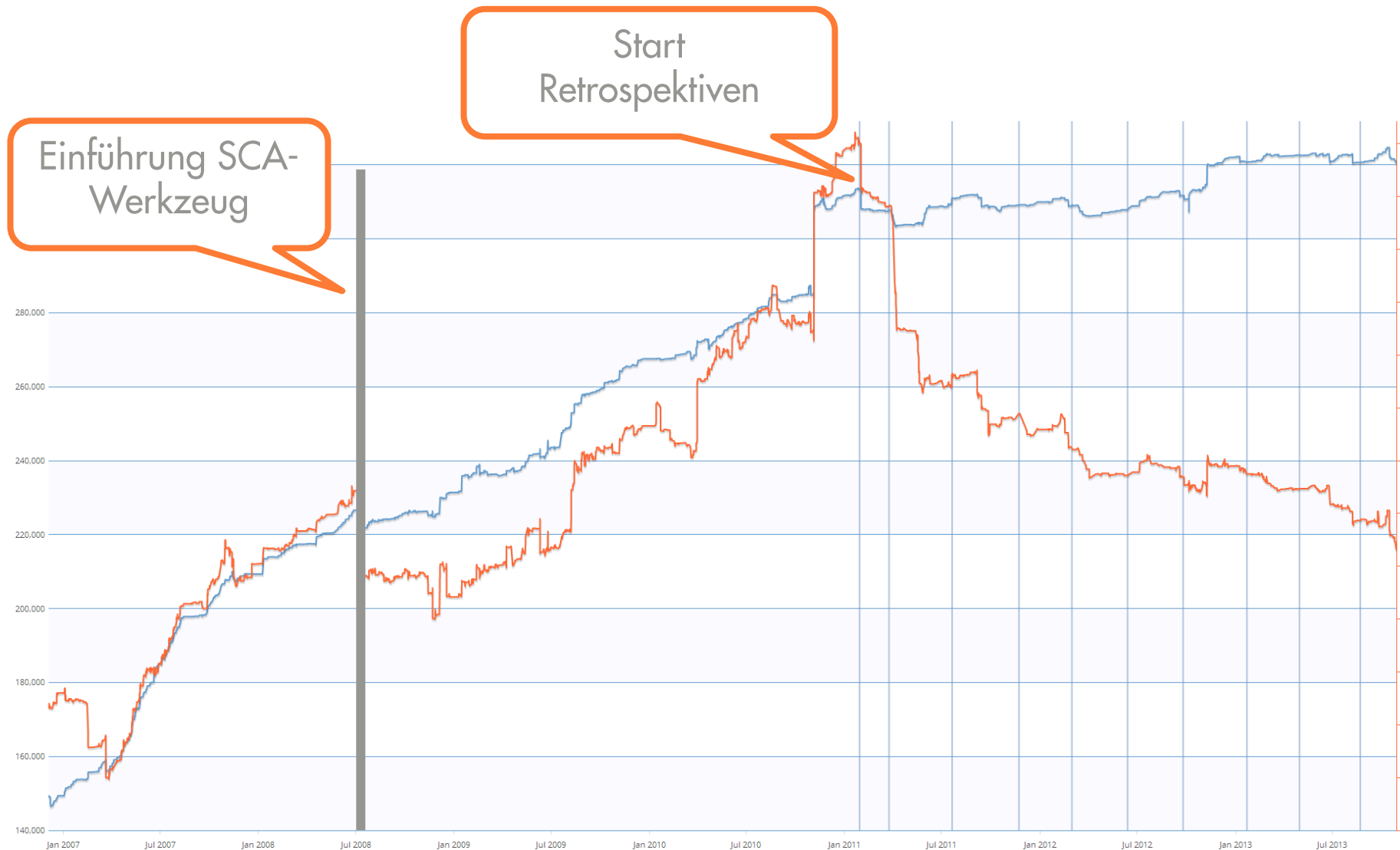
→ Verbesserungsvorschläge ermöglichen Einplanung und konkrete Verbesserung

→ Diskussionen schaffen gemeinsames Qualitätsverständnis und Best Practices

Was brought Continuous Quality Control?



→ SCA-Werkzeug hat i.d.R. nur einen kurzfristigen Effekt



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

Nutzen der Klonanalyse für Munich Re

Findings	
<input checked="" type="checkbox"/> All	6793
<input checked="" type="checkbox"/> Architecture	1
<input checked="" type="checkbox"/> Architecture Conformance	1
<input checked="" type="checkbox"/> Code Anomalies	1344
<input checked="" type="checkbox"/> Bad practice	971
<input checked="" type="checkbox"/> Correctness	2
<input checked="" type="checkbox"/> Exception Handling	62
<input checked="" type="checkbox"/> General checks (built-in)	120
<input checked="" type="checkbox"/> Null pointer dereference	13
<input checked="" type="checkbox"/> Performance	36
<input checked="" type="checkbox"/> Unused code	93
<input checked="" type="checkbox"/> Unused variable or parameter	47
<input checked="" type="checkbox"/> Code Duplication	988
<input checked="" type="checkbox"/> Cloning	101
<input checked="" type="checkbox"/> Redundant Literals	887
<input checked="" type="checkbox"/> Documentation	3378
<input checked="" type="checkbox"/> Comment completeness	3236
<input checked="" type="checkbox"/> Task tags	142
<input checked="" type="checkbox"/> Formatting	6
<input checked="" type="checkbox"/> Code formatting	6
<input checked="" type="checkbox"/> Naming	110
<input checked="" type="checkbox"/> Java naming conventions	110
<input checked="" type="checkbox"/> Structure	966
<input checked="" type="checkbox"/> File Size	38
<input checked="" type="checkbox"/> Method Length	278
<input checked="" type="checkbox"/> Nesting Depth	650

$$500 \frac{PT}{Jahr}^*$$

Munich Re spart durch Einsatz von Clone Detection jährlich ca. 500 PT Aufwand für Fehlerbehebung

$$\text{Ersparnis Aufwand} = 6\%^*$$

Munich Re spart durch Einsatz von Clone Detection jährlich 6% Aufwand durch vermiedene Redundanz ein.

→ Continuous Quality Control hat i.d.R. großen Nutzen

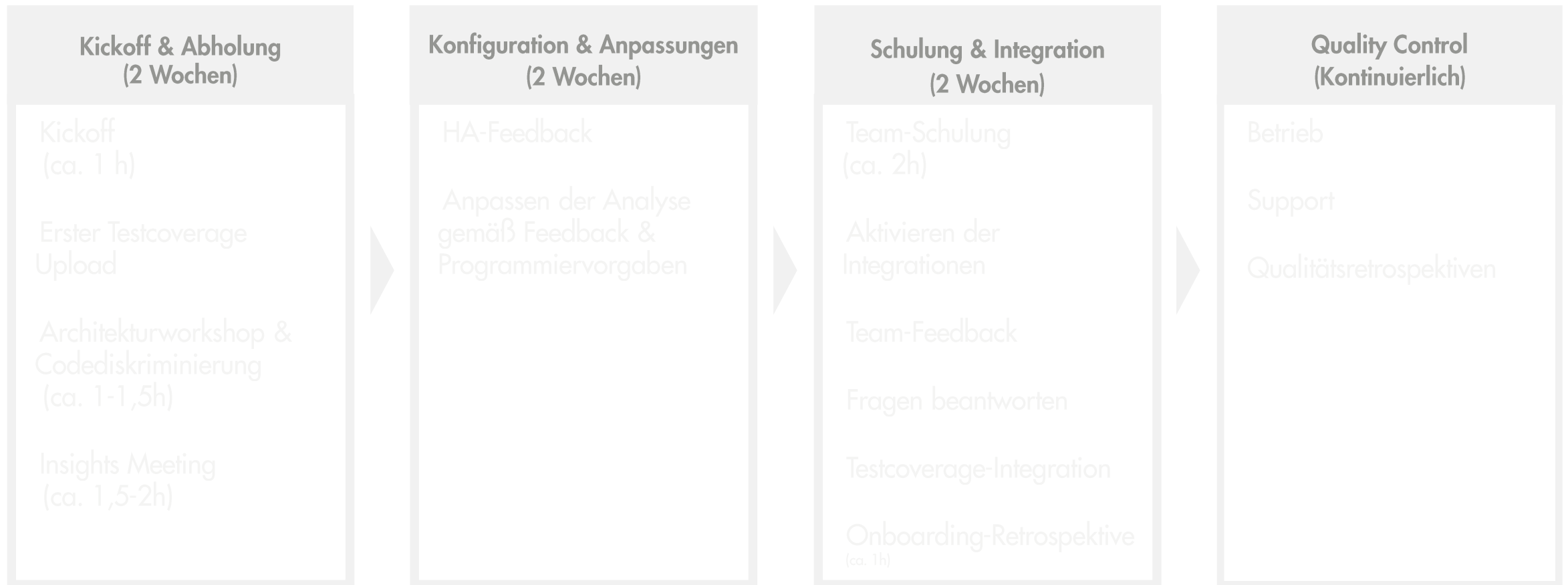
* Übertragbarkeit muss geprüft werden; für ausführliche, konservative Herleitung siehe [CQSE-Workshop „Copy, Paste und dann ...?“](#)

Wie führt man Continuous Quality Control ein?

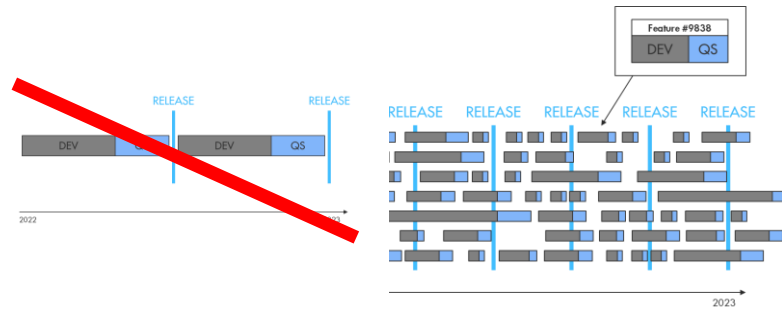


→ Technische und soziale Faktoren berücksichtigen

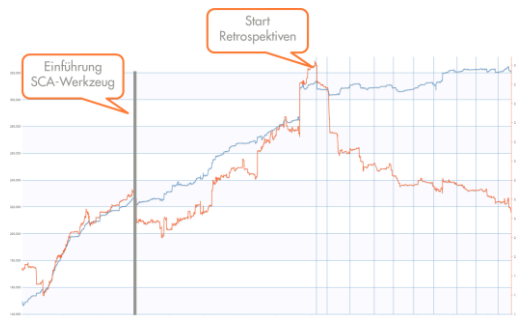
Teamscale-Onboarding-Prozess (Übersicht)



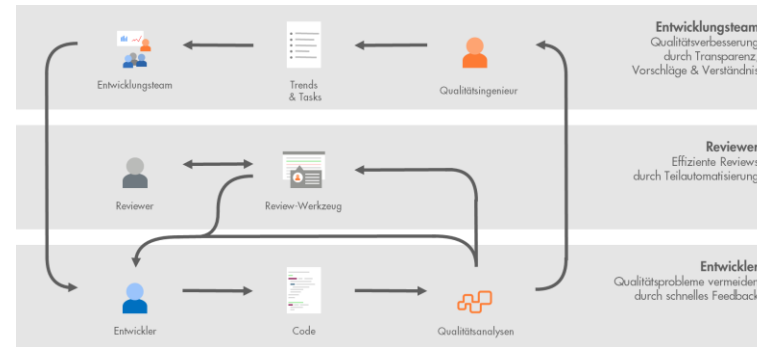
Zusammenfassung



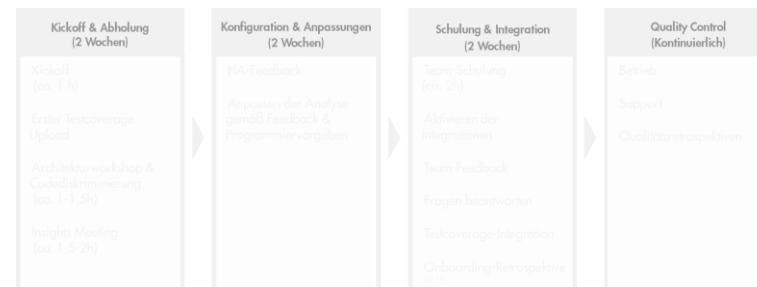
→ Releasefähigkeit erfordert jederzeitige Releasequalität und kontinuierliche QS



→ CQC hat i.d.R. großen Nutzen (auch bei gewachsenen Anwendungen)



→ CQC ermöglicht dies durch Feedbackschleifen für Entwickler, Reviewer und Entwicklungsteam



→ Einführung von CQC erfordert planvolles Vorgehen

Kontakt



Dr. Tobias Röhm · roehm@cqse.eu · [Termine](#)

Tobias Wiese · wiese@cqse.eu · [Termine](#)

CQSE GmbH
Centa-Hafenbrädl-Straße 59
81249 München

Vielen Dank für Ihre Teilnahme!

Präzise wie ein Uhrwerk

Mit 80.000 Tests sorgt die ATOSS Software AG für reibungsloses Digital Workforce Management

online & kostenlos

Mittwoch, 20. September 2023
10:30 bis 12:00 Uhr

Dr. Bernd Vogel
ATOSS Software AG

Raphael Nömmer
CQSE GmbH



CQSE Software Intelligence Talk

Anmeldung: cqse.eu/si-talk-23-9-qc236

Test-Gap-Analyse

Ungetestete Änderungen im Quelltext aufdecken

online & kostenlos

Mittwoch, 04. Oktober 2023

10:30 bis 12:00 Uhr



Fabian Streitl



Thomas Pettinger

CQSE Workshop

Anmeldung: cqse.eu/tga-23-10-qc236

Copy, Paste und dann ...?

Erfahrung aus 15 Jahren Clone-Detection und -Management

online & kostenlos

Mittwoch, 18. Oktober 2023

10:30 bis 12:00 Uhr



Dr. Elmar Jürgens



Dr. Michael Reif

CQSE Workshop

Anmeldung: cqse.eu/clone-23-10-qc236