

Wann führen Qualitäts- analysen zu guter Software? Erfahrungen aus 7 Jahren Praxiseinsatz

Dr. Elmar Juergens

Über Mich

Forschung

- Clone Detection
- Architekturanalyse

Beratung

- Mitgründer
- Qualitäts-Bewertung & Qualitäts-Controlling



```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        for (int i = 0; i < elements.length; i++) {
            ModelElement el = elements[i];
            res.append(showElementLink(el)).append(HTML.LINE_BREAK);
            found = true;
        }
        Index.getInstance().resetCurrentRenderer();
    }
    if (!found && nomsg != null && nomsg.length() > 0) {
        res.append(HTML.italics(nomsg));
    }
    return res.toString();
}
```

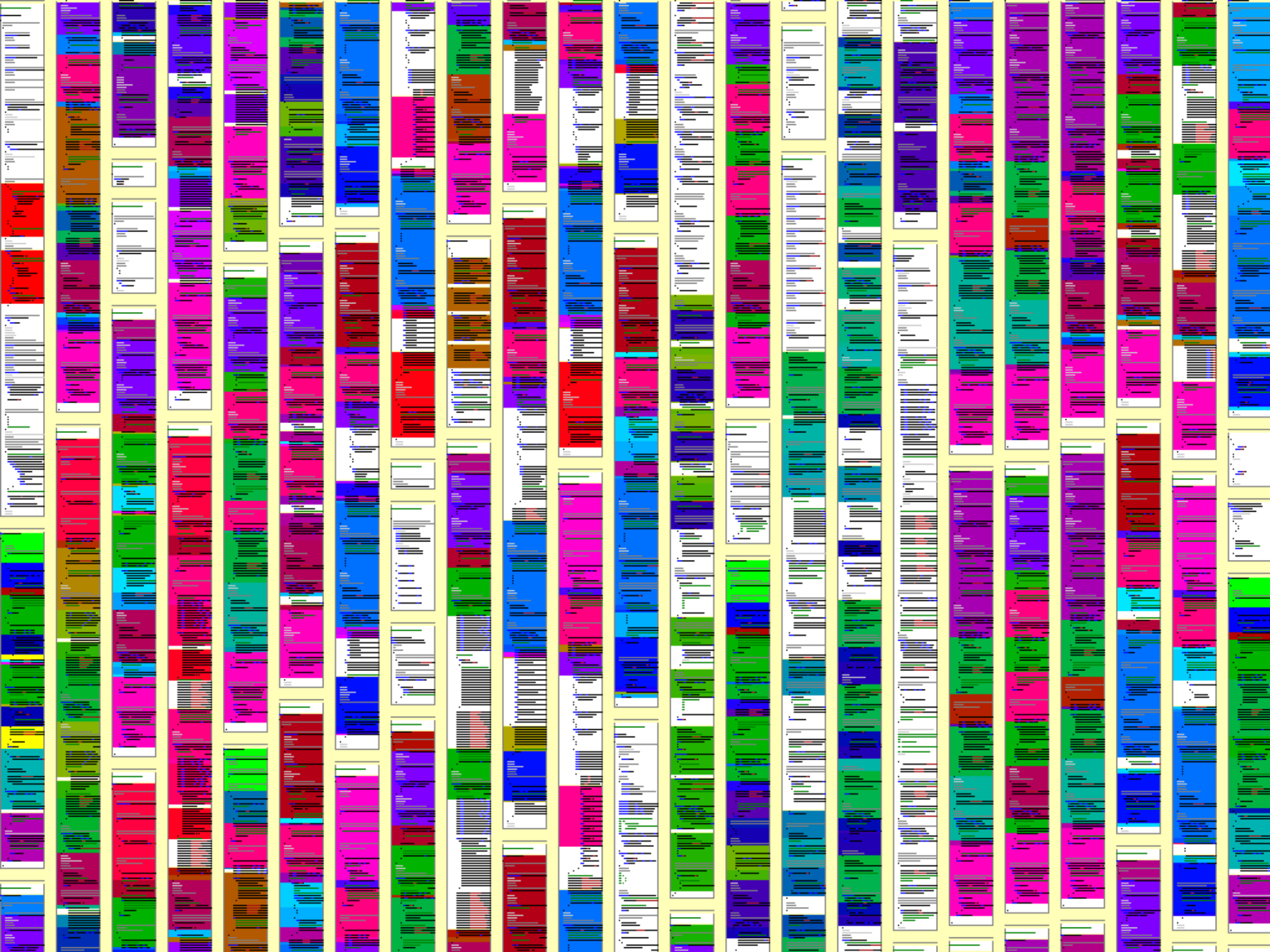
```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        for (int i = 0; i < elements.length; i++) {
            ModelElement el = elements[i];
            res.append(showElementLink(el)).append(HTML.LINE_BREAK);
            found = true;
        }
        Index.getInstance().resetCurrentRenderer();
    }
    if (!found && nomsg.length() > 0) {
        res.append(HTML.italics(nomsg));
    }
    return res.toString();
}
```

```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        for (int i = 0; i < elements.length; i++) {
            ModelElement el = elements[i];
            res.append(showElementLink(el)).append(HTML.LINE_BREAK);
            found = true;
        }
        Index.getInstance().resetCurrentRenderer();
    }
    if (!found && nomsg != null && nomsg.length() > 0) {
        res.append(HTML.italics(nomsg));
    }
    return res.toString();
}
```









Studie

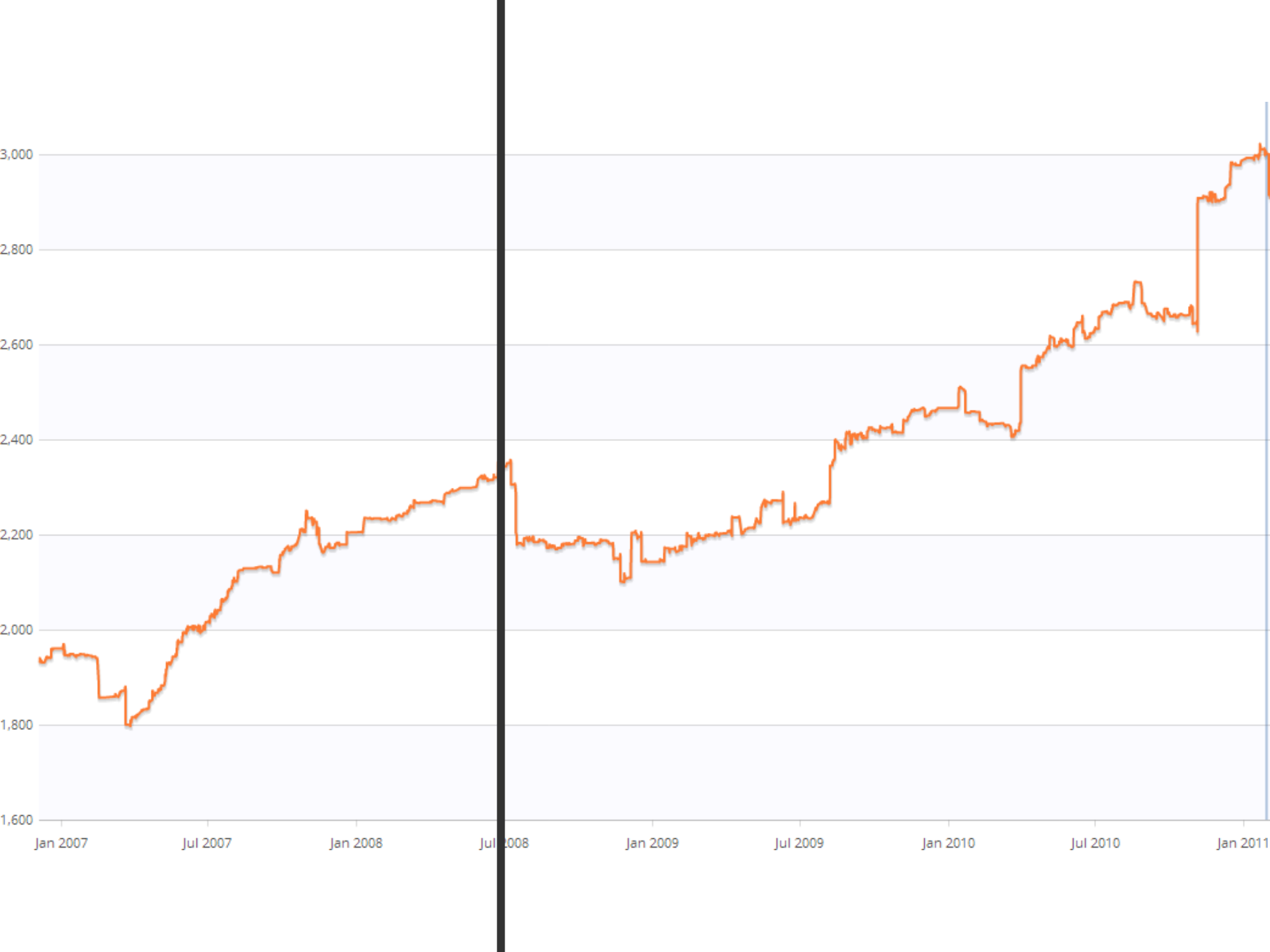
Munich Re 

- Über 100 Fehler in produktiver Software



- 52% aller ungewollten Unterschiede fehlerhaft

Juergens, Deissenboeck et al: *Do Code Clones Matter?* ICSE 2009

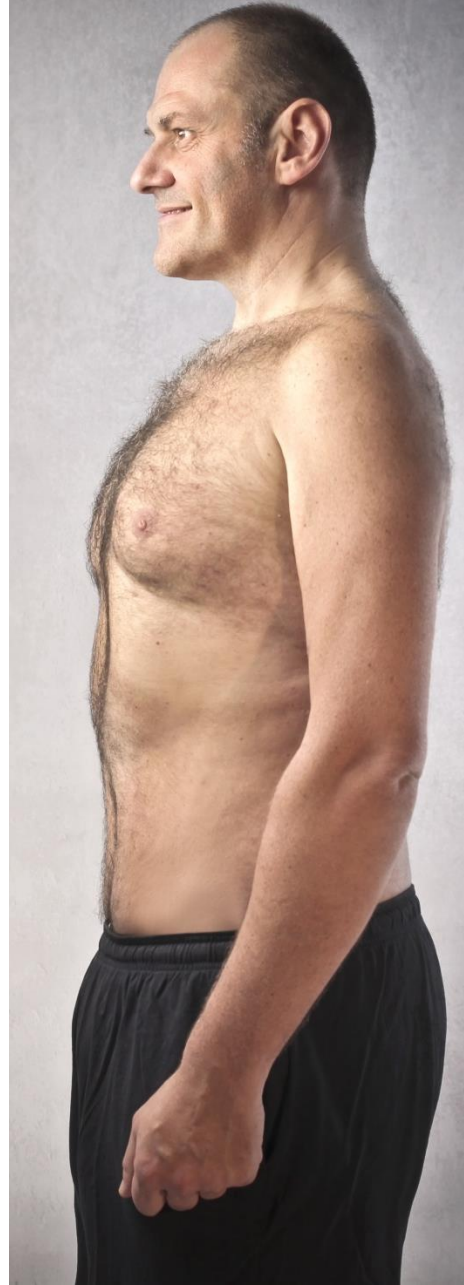




+



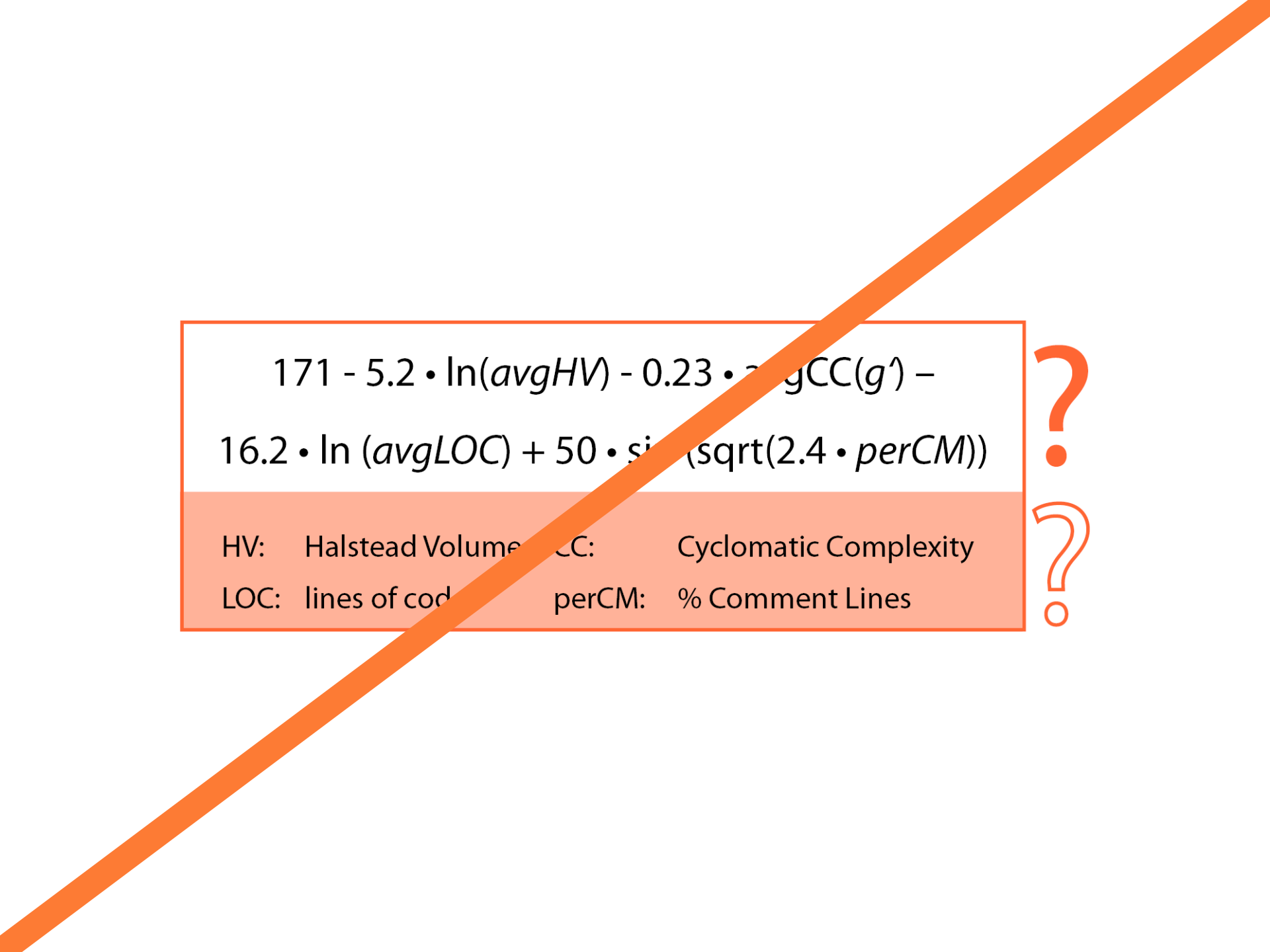
=







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

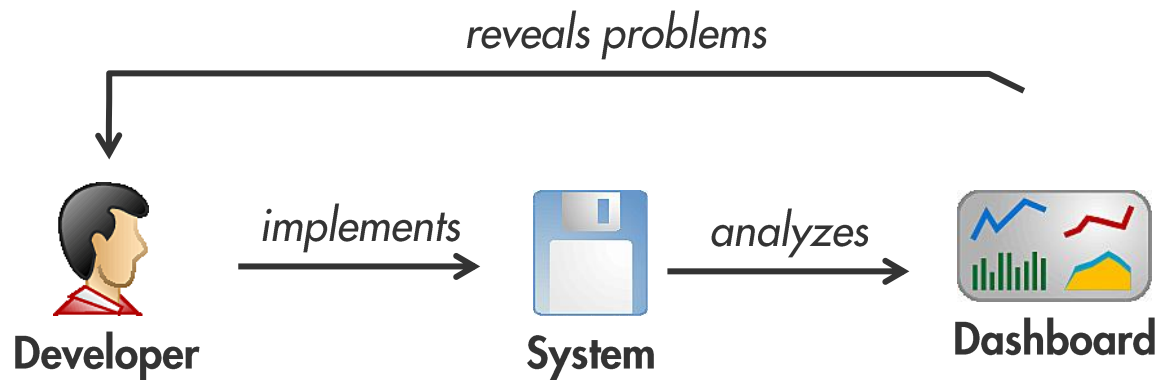

$$171 - 5.2 \cdot \ln(\text{avgHV}) - 0.23 \cdot \ln(\text{avgCC}(g)) - \\ 16.2 \cdot \ln(\text{avgLOC}) + 50 \cdot \sin(\text{sqrt}(2.4 \cdot \text{perCM}))$$

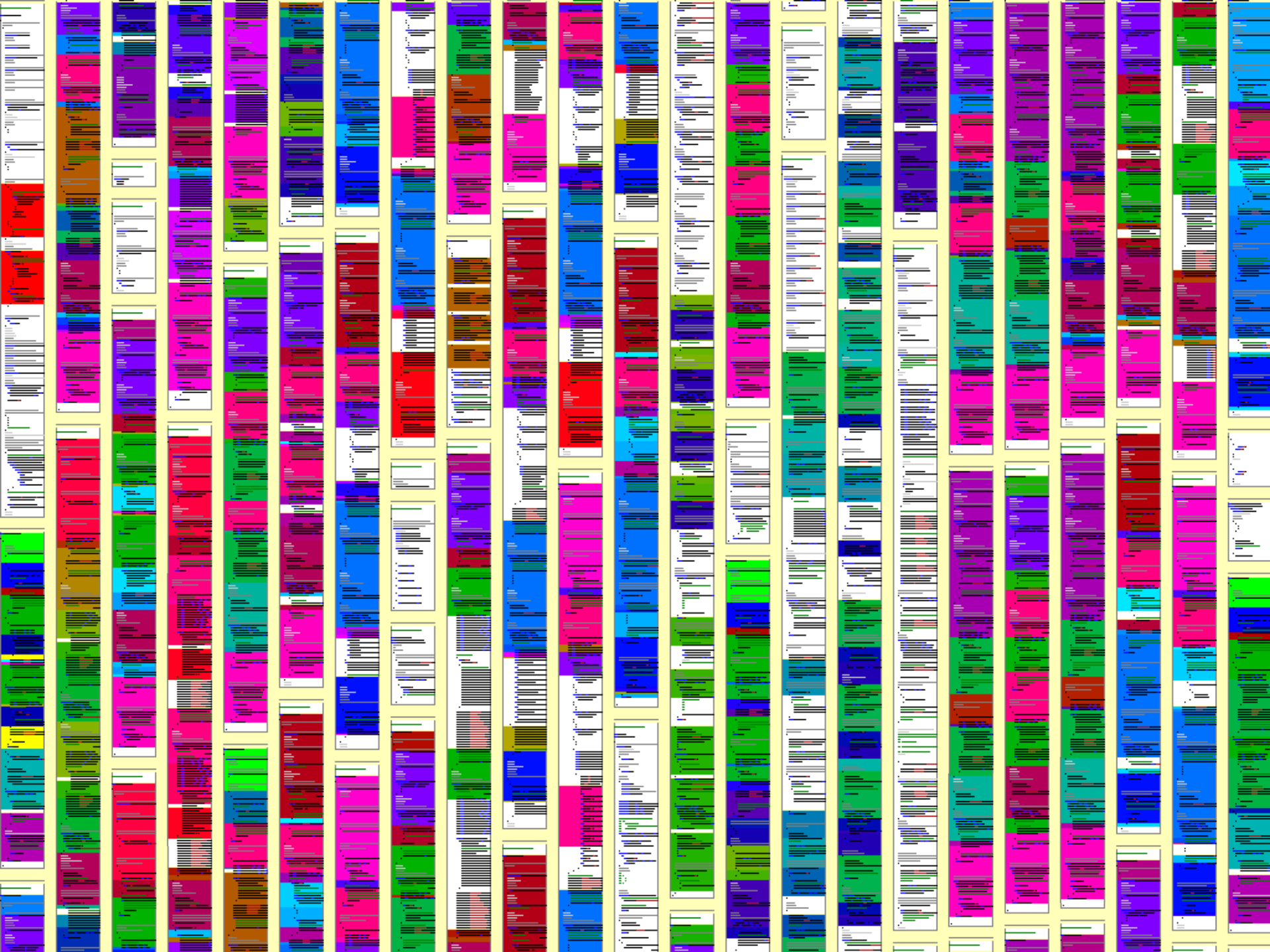
HV: Halstead Volume CC: Cyclomatic Complexity
LOC: lines of code perCM: % Comment Lines

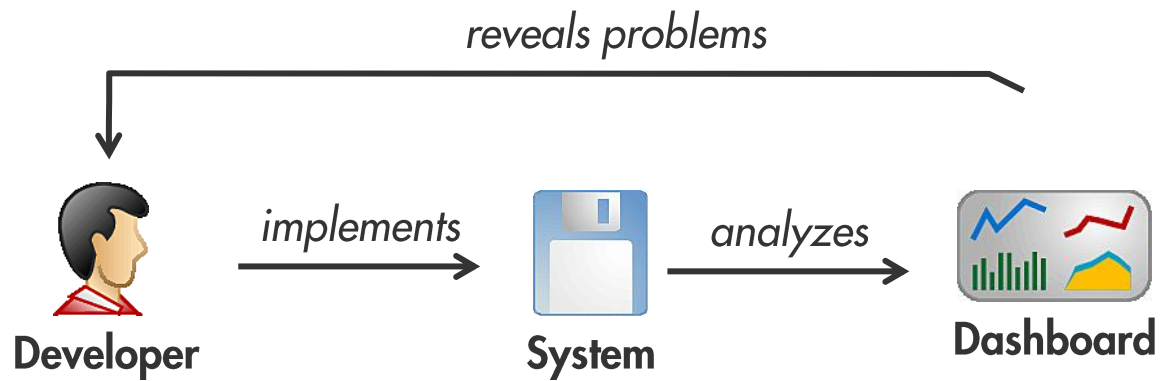


Best Practice: Zielorientierte Analysen

- Objektiv
- Auswirkungen von Code-Änderungen verständlich
- Actionable
- Nachvollziehbarer Zusammenhang zu Wartungstätigkeit

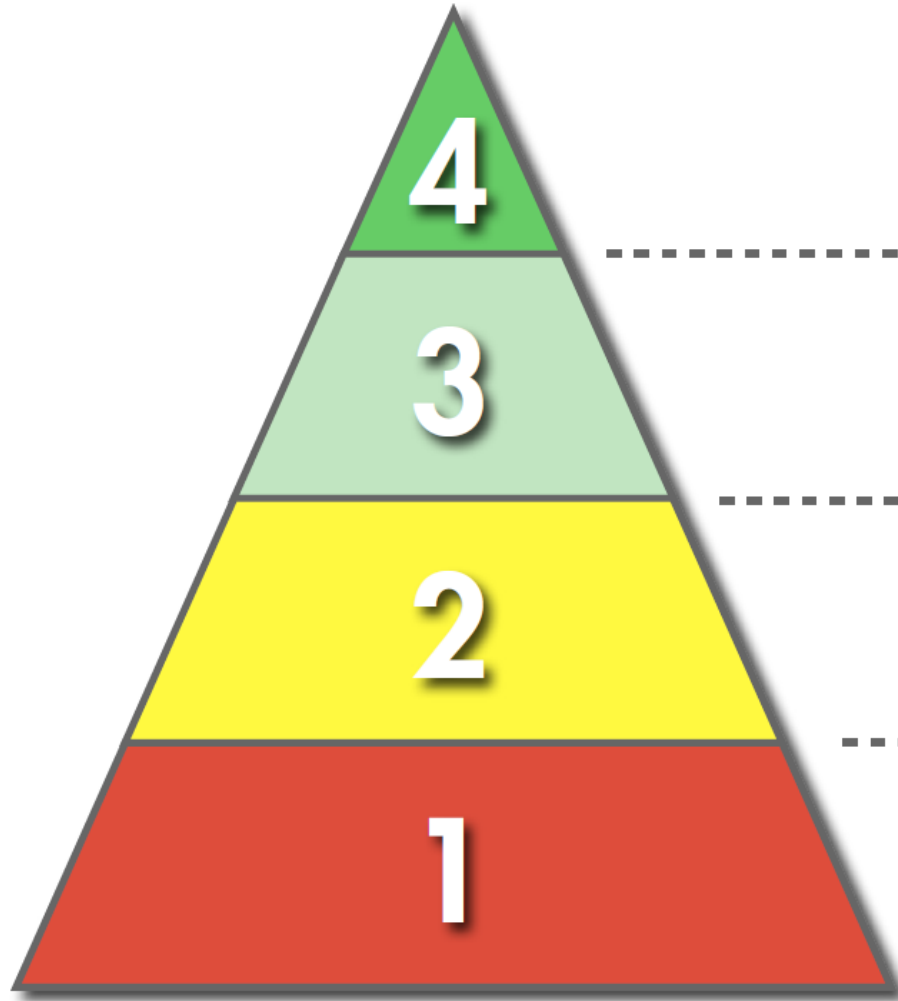






- Information overflow
=> Analyseergebnisse werden ignoriert



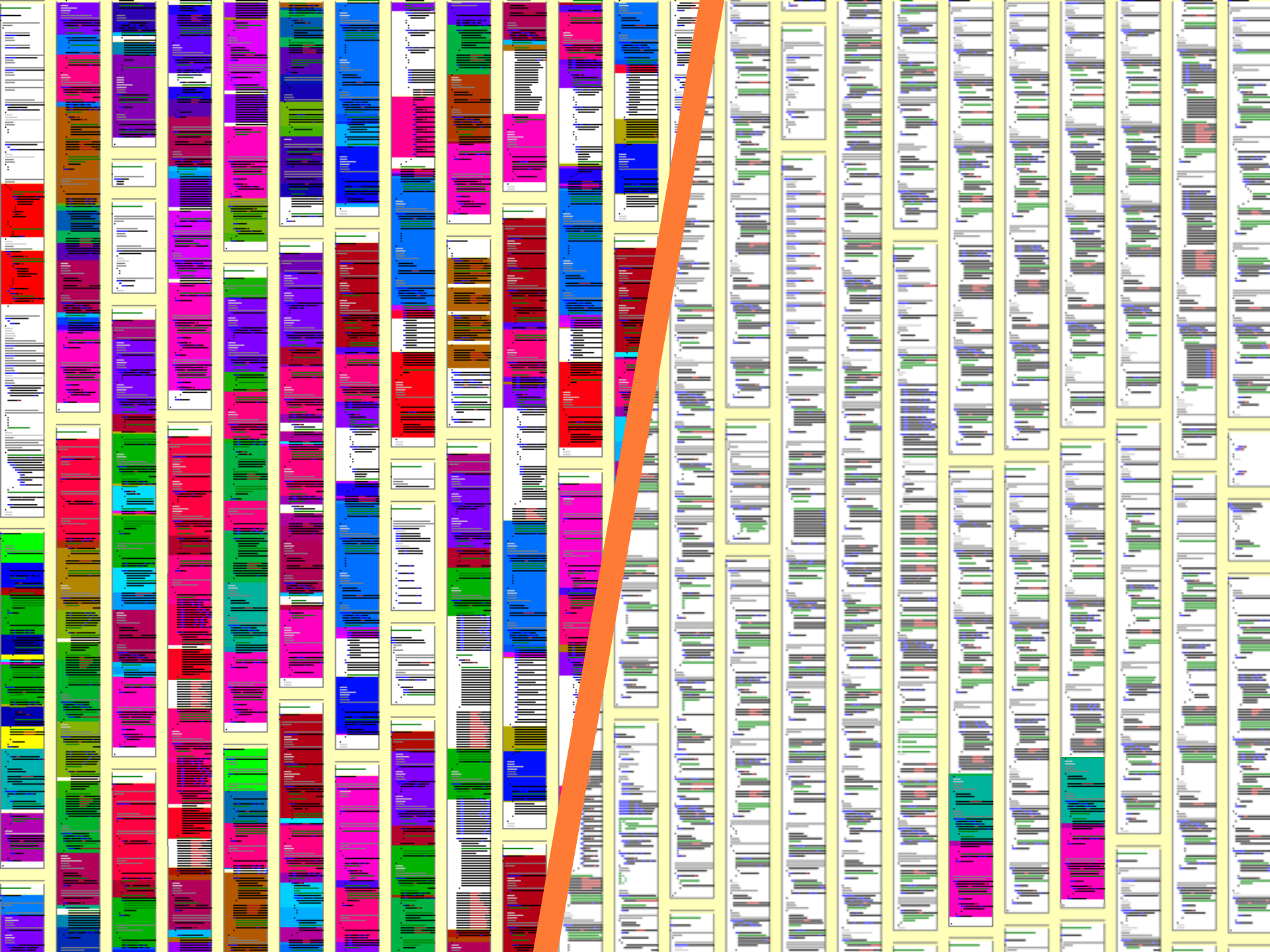


Keine Defizite

Keine Defizite in geändertem Code

Keine neuen Defizite

Egal





ACT-1270 Fixing Inconsistent handling of serializable process variables

by [Victoria King](#) in revision [e1aa41b4b133d269980fff3f81d008da8f21a109](#) (git)

Jun 29 2012

16:05

changed 2 files

-4 findings



ACT-1258 Merging Pablo's work into trunk

by [Jacob Nelson](#) in revision [9e664a1f0676cedcbe03415a253e8c3e4a58944c](#) (git)

Jun 29 2012

14:41

added 3 files, changed 2 files

-1 findings



Fix for ACT-1059: Task#setDelegationState(DelegationState) was not saved in database

by [Michael Harris](#) in revision [1f48dcad04bc4a621e60af047fb121ae161bca30](#) (git)

Jun 28 2012

21:45

changed 3 files

+2 findings



ACT-991 Removed user id from exception message in order not to leak sensitive information

by [Michael Harris](#) in revision [e9a09424e6309c854c44ac5d08740a8ffb082fc9](#) (git)

Jun 28 2012

15:26

changed 2 files

-2 findings



100k

4.7%
no change



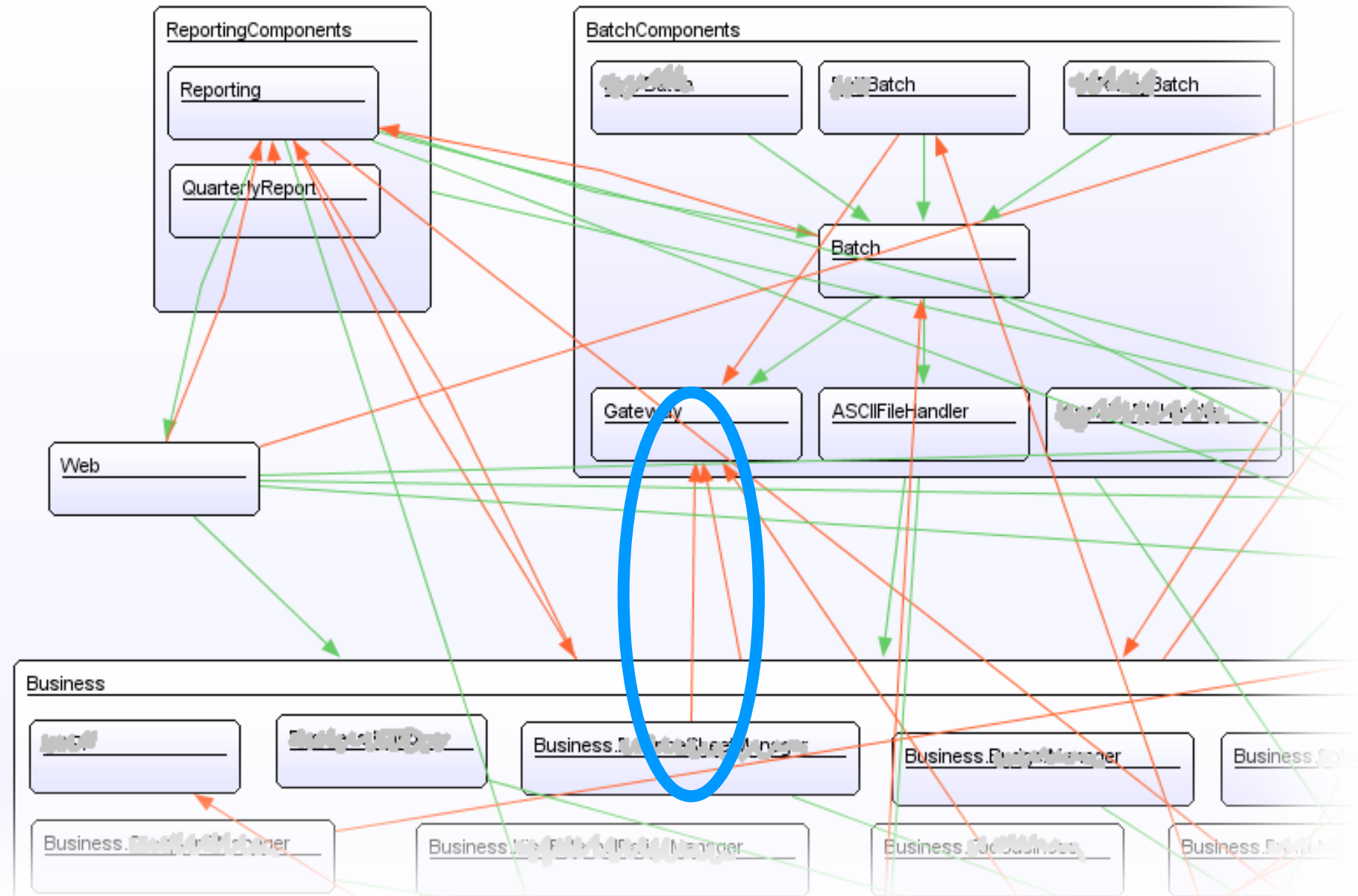
488

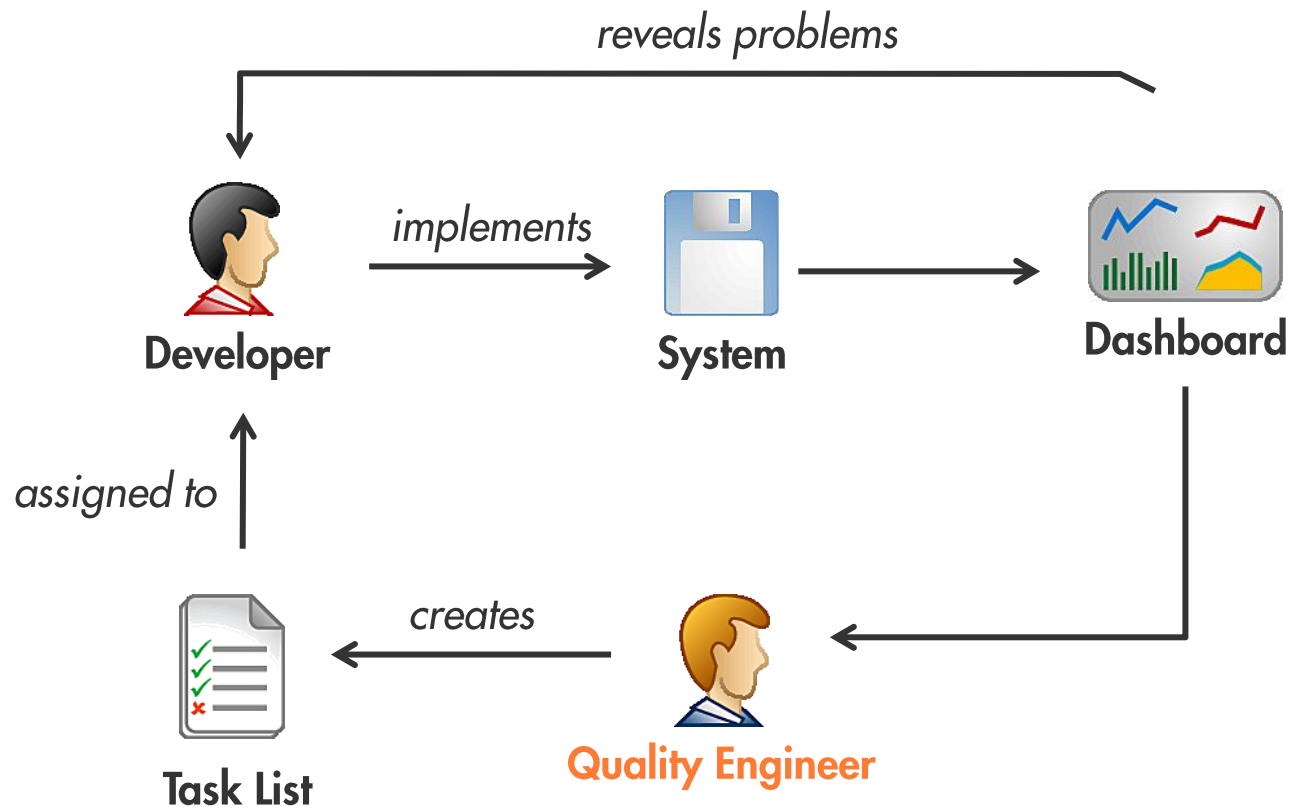
Bei Interesse: Demo im
Anschluss durch Elmar /
Thomas.



Findings Summary for teamscale

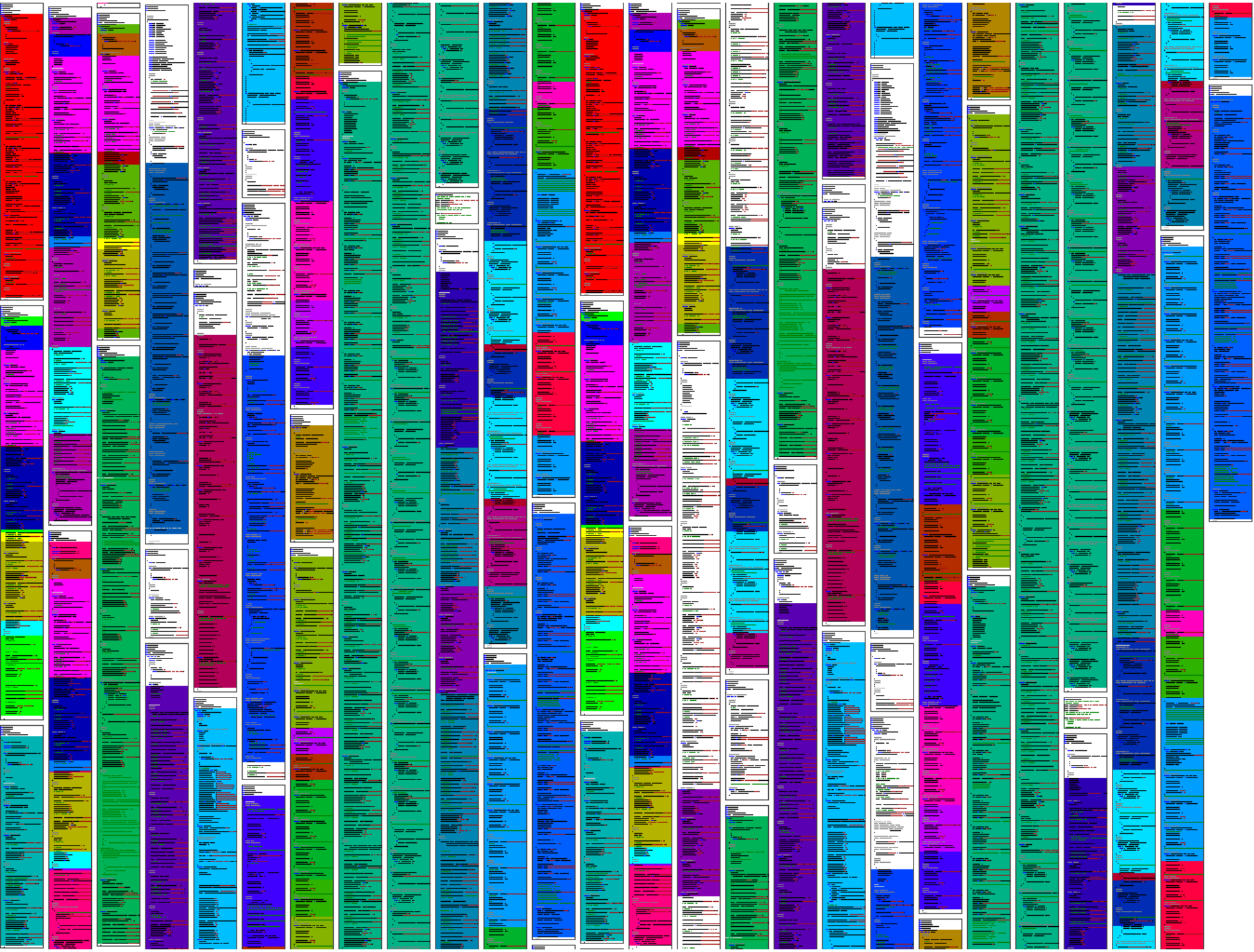
<http://www.teamscale.com>





- Awareness of Management?
=> Keine Ressourcen, wenig Verbesserung







Komponente A



Komponente B

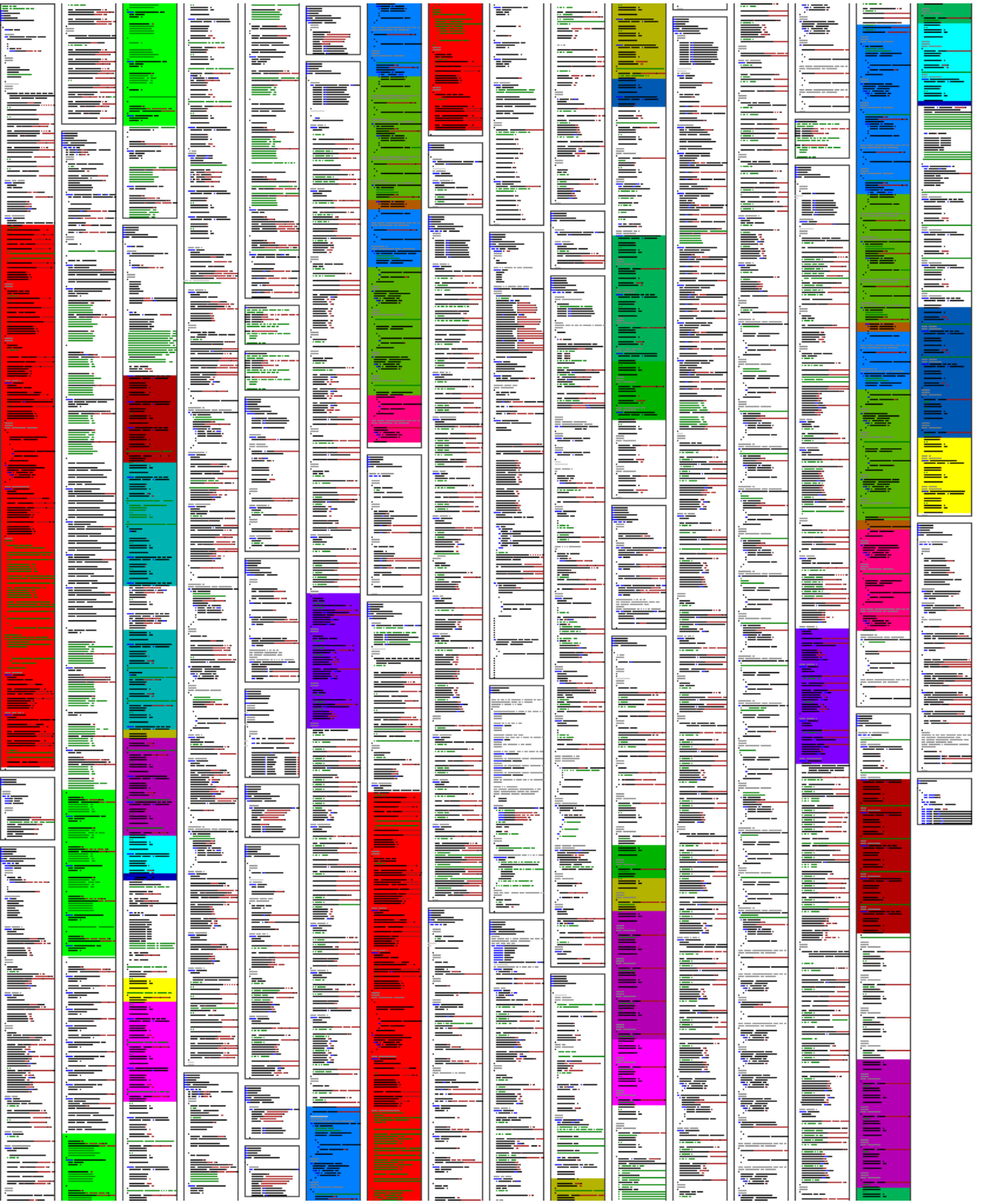


Figure 1: Genomic data visualization across 22 chromosomes.

Figure 2: Genomic data visualization across 22 chromosomes.

Figure 3: Genomic data visualization across 22 chromosomes.

Figure 4: Genomic data visualization across 22 chromosomes.

Figure 5: Genomic data visualization across 22 chromosomes.

Figure 6: Genomic data visualization across 22 chromosomes.

Figure 7: Genomic data visualization across 22 chromosomes.

Figure 8: Genomic data visualization across 22 chromosomes.

Figure 9: Genomic data visualization across 22 chromosomes.

Figure 10: Genomic data visualization across 22 chromosomes.

Figure 11: Genomic data visualization across 22 chromosomes.

Figure 12: Genomic data visualization across 22 chromosomes.

Figure 13: Genomic data visualization across 22 chromosomes.

Figure 14: Genomic data visualization across 22 chromosomes.

Figure 15: Genomic data visualization across 22 chromosomes.

Figure 16: Genomic data visualization across 22 chromosomes.

Figure 17: Genomic data visualization across 22 chromosomes.

Figure 18: Genomic data visualization across 22 chromosomes.

Figure 19: Genomic data visualization across 22 chromosomes.

Figure 20: Genomic data visualization across 22 chromosomes.

Figure 21: Genomic data visualization across 22 chromosomes.

Figure 22: Genomic data visualization across 22 chromosomes.

Figure 23: Genomic data visualization across 22 chromosomes.

Figure 24: Genomic data visualization across 22 chromosomes.

Figure 25: Genomic data visualization across 22 chromosomes.

Figure 26: Genomic data visualization across 22 chromosomes.

Figure 27: Genomic data visualization across 22 chromosomes.

Figure 28: Genomic data visualization across 22 chromosomes.

Figure 29: Genomic data visualization across 22 chromosomes.

Figure 30: Genomic data visualization across 22 chromosomes.

Figure 31: Genomic data visualization across 22 chromosomes.

Figure 32: Genomic data visualization across 22 chromosomes.

Figure 33: Genomic data visualization across 22 chromosomes.

Figure 34: Genomic data visualization across 22 chromosomes.

Figure 35: Genomic data visualization across 22 chromosomes.

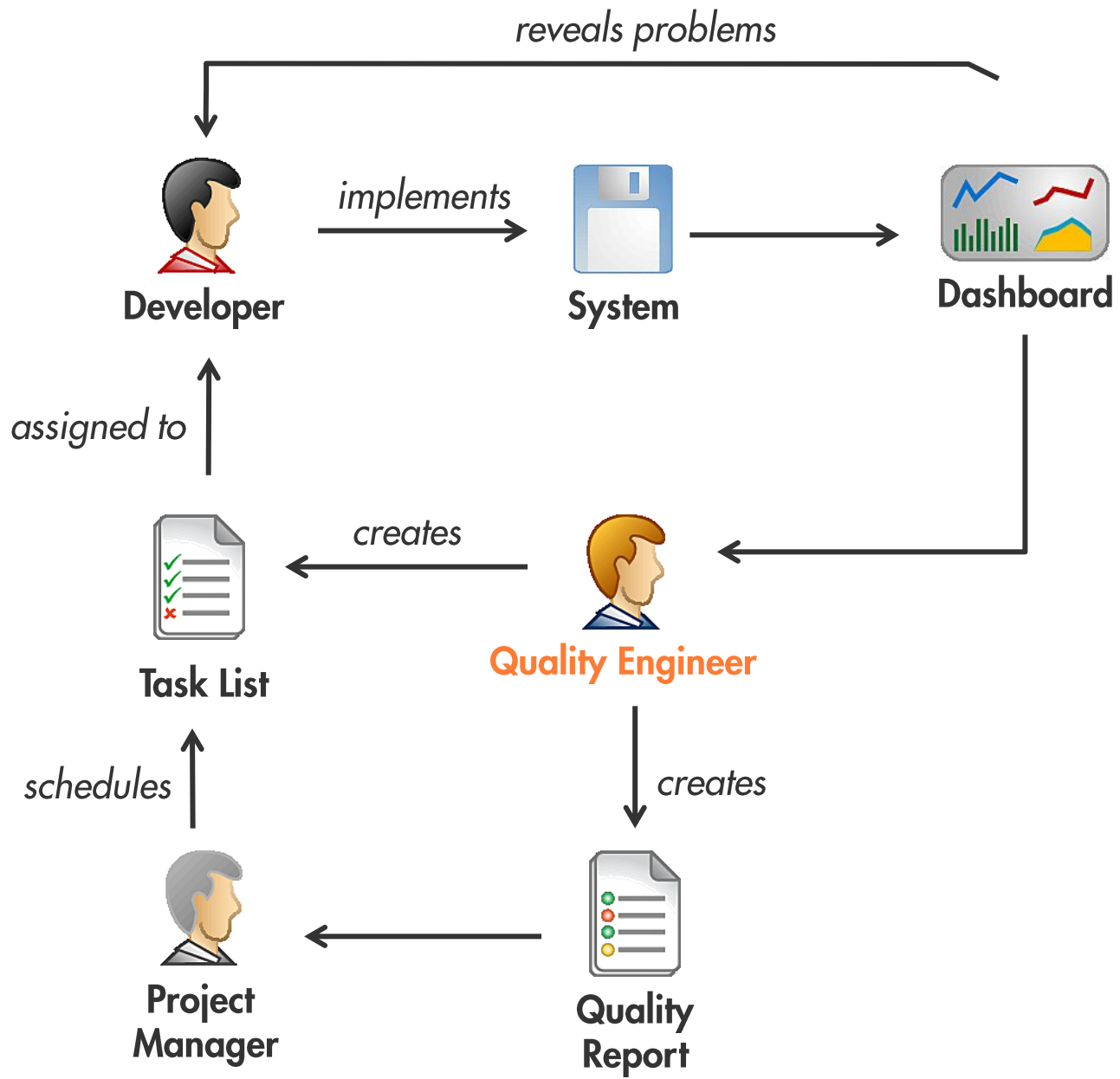
Figure 36: Genomic data visualization across 22 chromosomes.

Figure 37: Genomic data visualization across 22 chromosomes.

Figure 38: Genomic data visualization across 22 chromosomes.

Figure 39: Genomic data visualization across 22 chromosomes.

Figure 40: Genomic data visualization across 22 chromosomes.

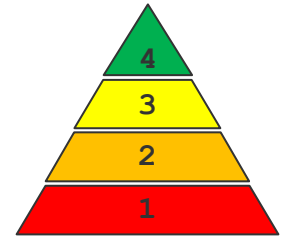


Best Practice: Quality Reports

Quality Indicator	Full Assessment		Delta Assessment	
Stability of Regular Team Builds	Build is stable.	✓	At the time of the last report, there was no automated build at all.	➔
Architecture Conformance	There are zero violations.	✓	Architecture specification was completed and is now fully	➔
Stability of U Tests	<div style="border: 1px solid black; padding: 10px;"> <h3>1.7 Duplicated Code ✗ ➔</h3> <p>TQE Target: Clone coverage of less than 10%.</p> <div style="text-align: center;"> <h4>Clone Coverage</h4> <p>0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 %</p> </div> </div>			
Code Coverage				
Compiler Warnings				
Coding Guidelines Violations				
Duplicated Code	With a clone coverage of 12.2% the threshold of 10% is slightly violated.	✗	Clone coverage did not change significantly.	➔
File Size	The threshold regarding files > 400 LOC is violated.	✗	Use of partial classes improves the metric values but does not improve code quality	➔

Best Practice: TQE Tasks

- Bei jedem Report identifiziert
- Passend zu Quality Goal des Projekts
- Projekt entscheidet *ob* und *wann*



form routine.

99902 [*Duplicated Code*] Remove duplicated code in function modules

(from l. 169),

(from l. 274) and (from ll. 279, 577 and 1037) which recently were equally modified in 5 clone instances. Each duplicate is 91 lines long and equal unless two literals.


log parts.

99914 [*Nesting Depth*] Restructure function module


to reduce the deep nesting which was added. E.g. by extracting code within loops to helper function modules.


99915 [*Nesting Depth*] Restructure function module


Assessment of Overall System


	An automatic regular build needs to be established
--	--


The architecture specification


	Unit tests are not executed automatically
---	---


	The code expansion has many warnings.
---	---------------------------------------











	The number of warnings is slightly below thresholds
---	---

	With 12.8% of files, the amount is slightly above threshold.
---	--

















	File sizes are not within permitted thresholds
---	--

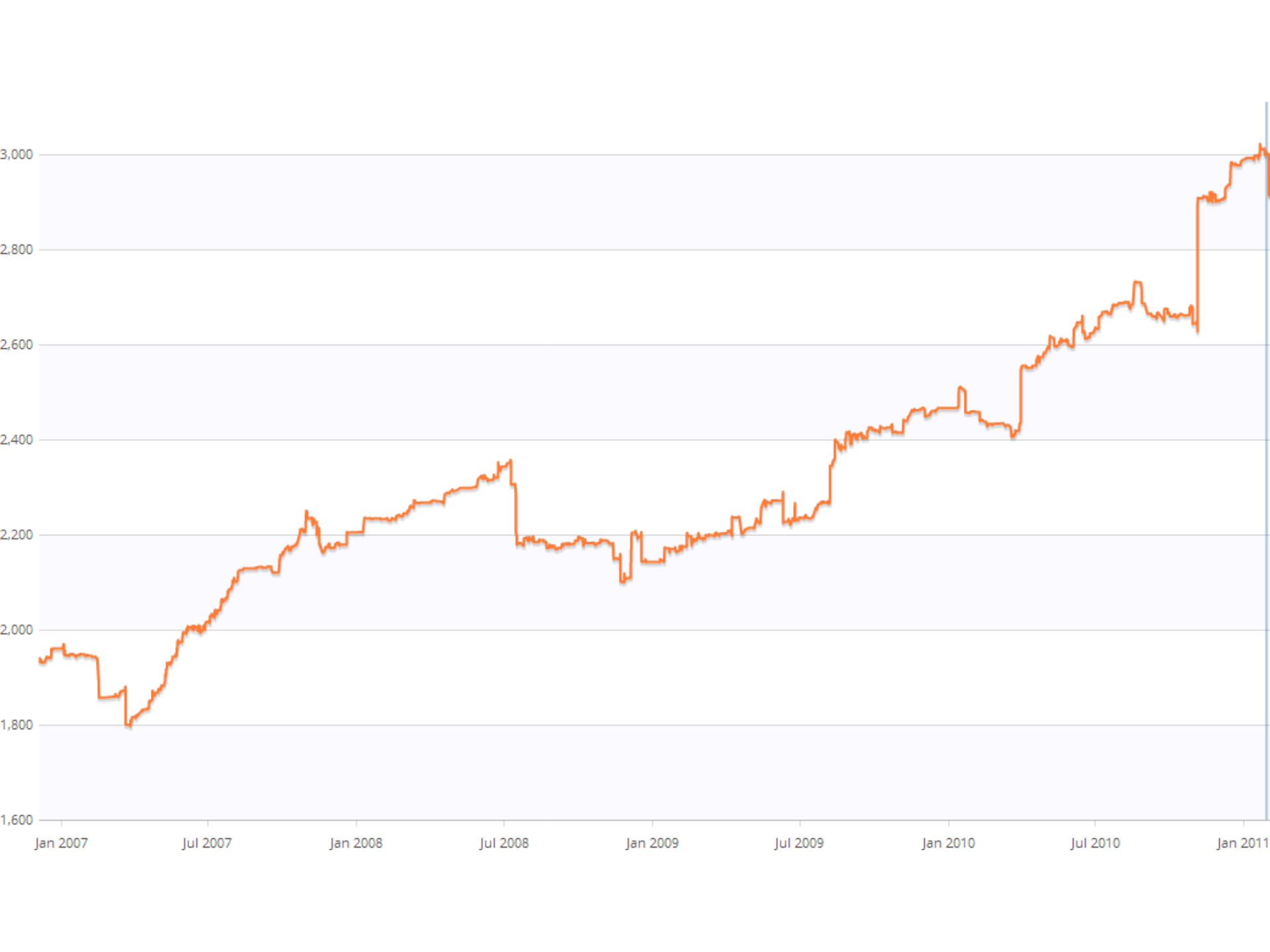
	Nesting depth is above the permitted threshold
---	--

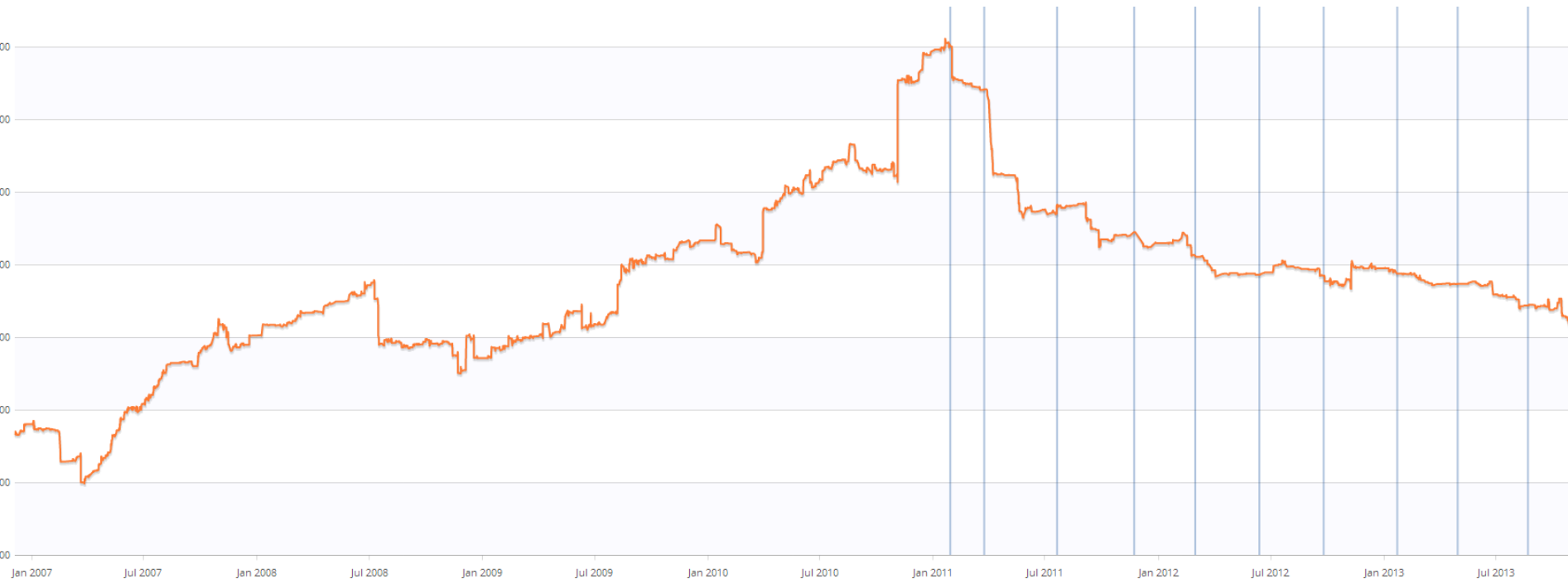
	Method length is above permitted thresholds
---	---

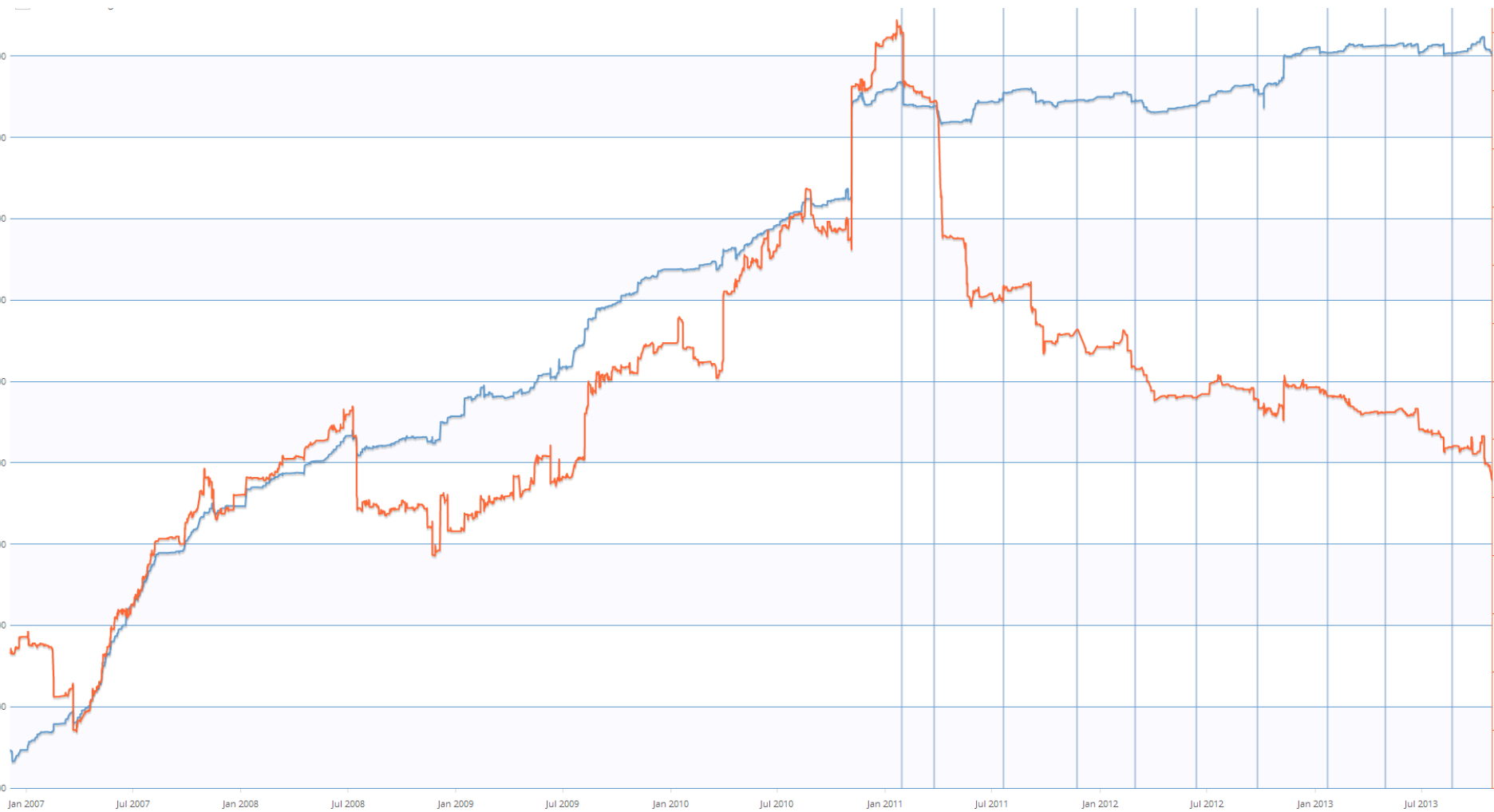
Current		Delta Assessment	
		At the time of the last report, there was no automated build at all.	
		Architecture specification was completed and is now fully adhered to.	
		The number of warnings was significantly higher, but their removal fixes actual quality.	
		The amount of code clones is one FxCop violation.	
		Clone coverage is significantly lower than the target.	
		Use of partial methods regarding files > 248 files with violations.	
		All nesting levels have been removed.	
		The amount of code in long methods has decreased significantly.	

Assessment of Overall System

Assessment of Overall System		Assessment Compared to Baseline	
The build is stable			
No policy is violated		No change	
Failing tests get fixed with delay		Most ignored tests are now passing	
0 compiler warnings		The amount of compiler warnings decreased	
248 files with violations		The amount of violations decreased	
7.3% clone coverage		The clone coverage decreased significantly	
45.6% code in long files		The amount of very long files has been significantly reduced	
1.7% deeply nested code		The amount of findings decreased significantly	
26.1% code in long methods		Less code in long methods	







Weitere Best Practices

- Spezifische Sichten für Stakeholder (aber: Transparenz!)
- Code Peer-Reviews
- Projektspezifisches Tailoring der Analysen
- Manuelle Reviews von KPI Verbesserungen
- Vollautomatische *Messung*. Manuelle *Bewertung*.
- Schnellstmögliches Feedback (Integration in IDE, Daily Builds, ...)
- ...

**Plakate
ankleben
verboten**

**Plakate
annageln
erlaubt?**

Baran.

Fazit

Qualitätsanalysewerkzeuge sind notwendig, aber nicht hinreichend für Verbesserung der Software-Qualität.

Nachhaltige Verbesserung erfordert die Unterstützung von Entwicklern und Management und die Integration in den Entwicklungsprozess.

Kontakt

Ich bin noch länger hier freue mich auf Diskussionen 😊

Dr. Elmar Jürgens · juergens@cqse.eu · +49 179 675 3863

[@ElmarJuergens](#)

[@teamscale](#)

www.cqse.eu/en/blog

CQSE GmbH, Lichtenbergstraße 8,
85748 Garching bei München