

Statische Codeanalyse: Stolz und Vorurteil... und Zombies

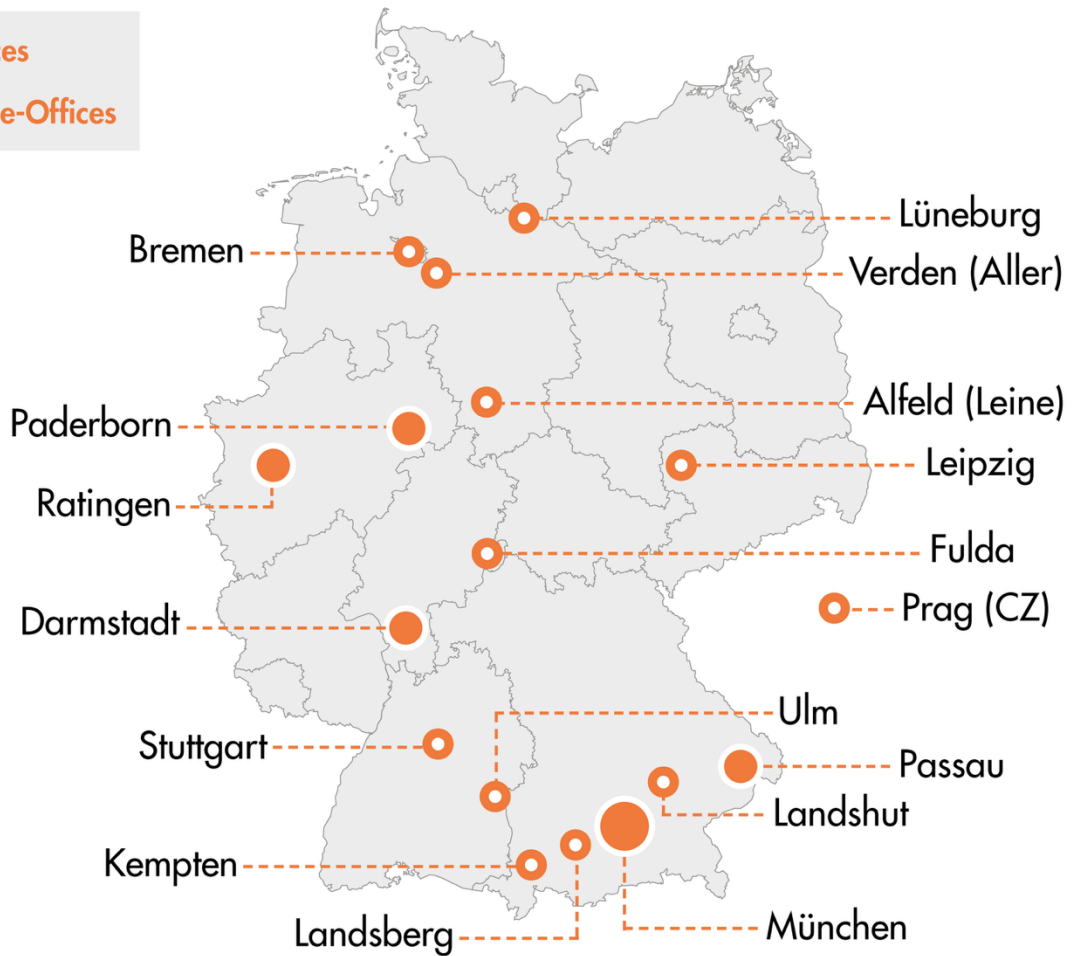
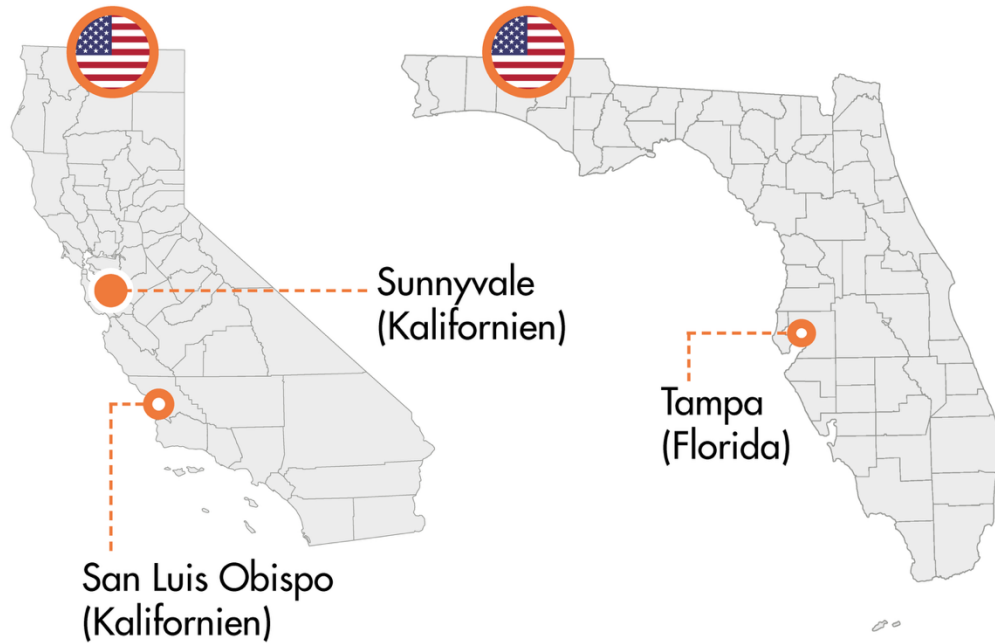


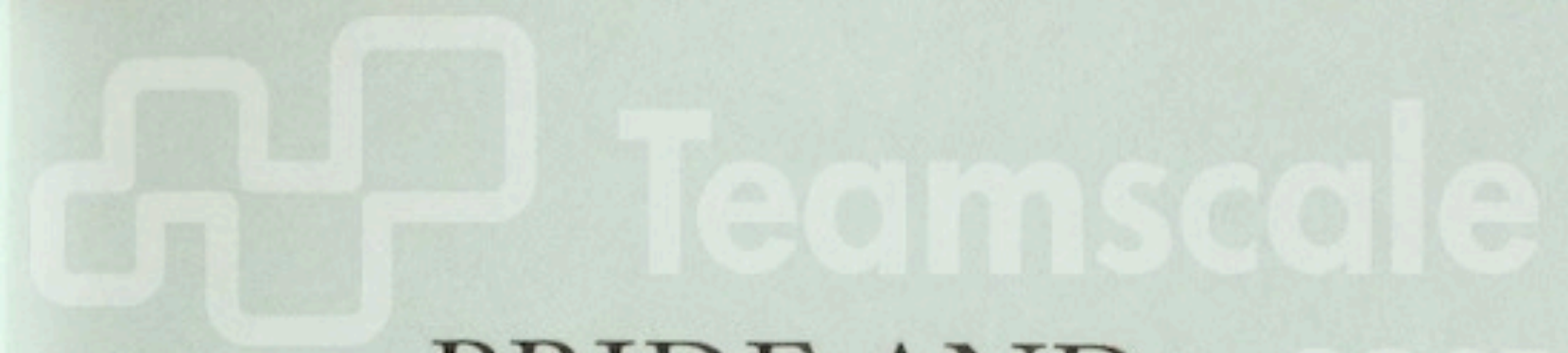
Ben Wolf
INNOQ



Sven Amann
Teamscale

- Offices
- Home-Offices





PRIDE AND by CQSE PREJUDICE

AND ZOMBIES

BENJAMIN WOLF

SOFTWARE INTELLIGENCE TALK

SEPTEMBER 23, 2026

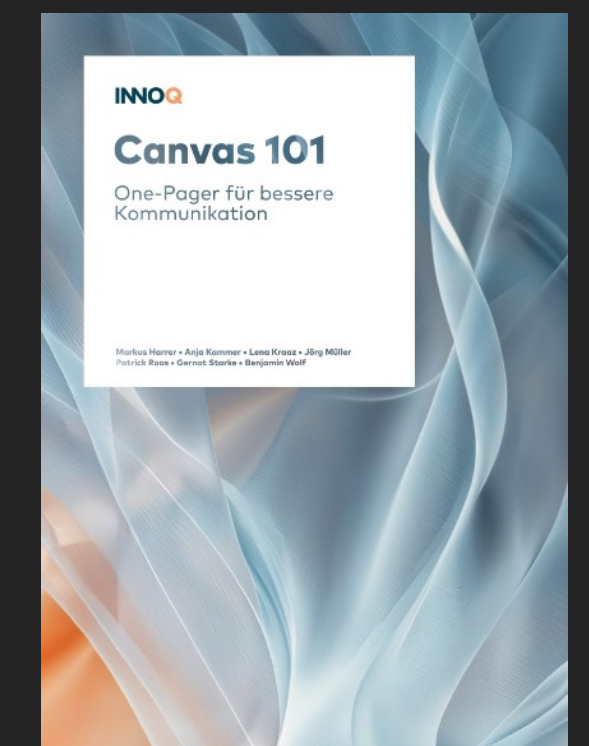
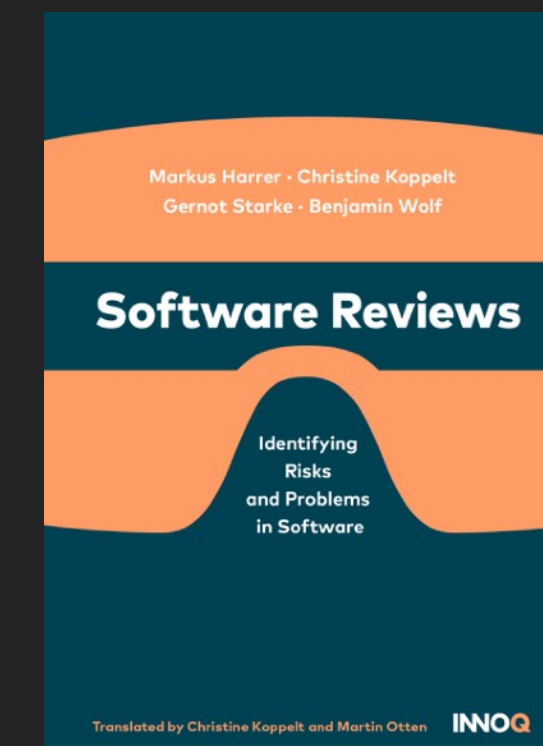
INNOQ

**„Jedes Problem lässt
sich mit gutem
Kaffee lösen.“**

Benjamin Wolf

Coffee Consultant bei INNOQ

Consultant für Architekturentwicklung / -dokumentation
Trainer für iSAQB Foundation, IMPROVE und ADOC
arc42-Fan und -Maintainer
Kaffeegenusspecht



Statische Codeanalyse?

Statische Codeanalyse ist eine Methode zur Untersuchung von Quellcode, ohne ihn tatsächlich auszuführen.

Ziel ist es, Fehler, Schwachstellen, Verstöße gegen Codierungsrichtlinien oder potenzielle Optimierungen im Code frühzeitig zu identifizieren.

Diese Analyse erfolgt in der Regel mithilfe von speziellen Tools, die den Code auf bestimmte Muster, Syntaxprobleme oder andere Qualitätskriterien prüfen.

Statische Codeanalyse?





Photo by [Daniel Hering](#) on [Unsplash](#)

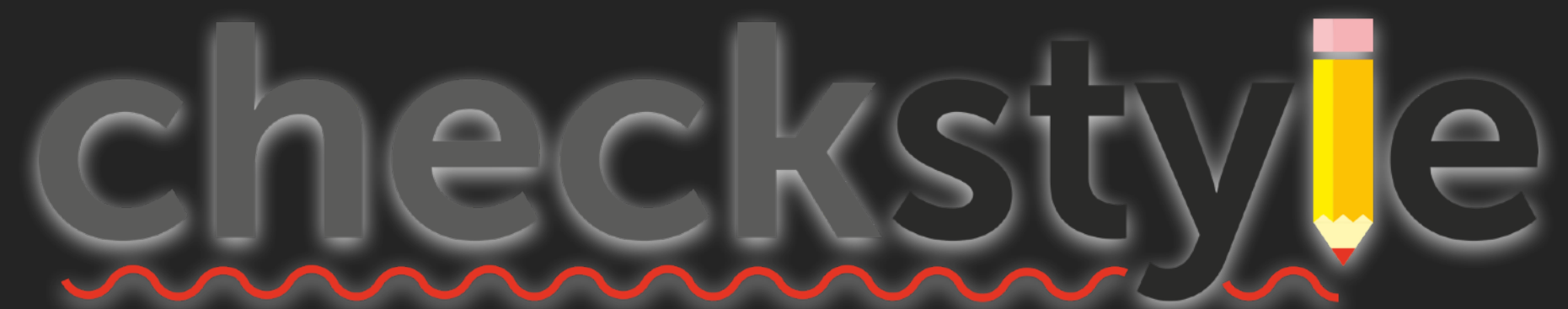


Photo by [Jonathan Cosens](#) Photography on [Unsplash](#)

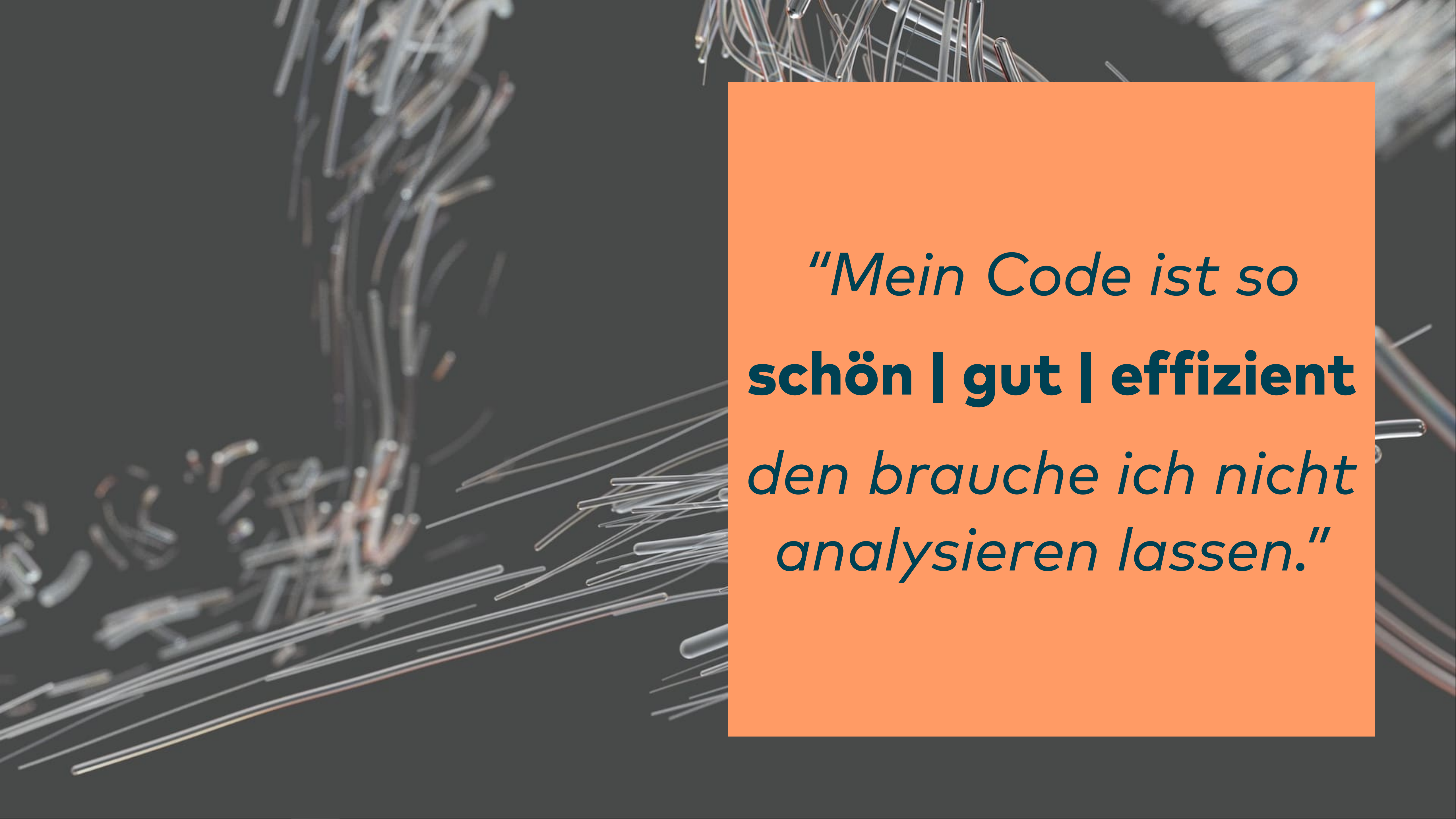
Arten von Codeanalyse

- Compiler-Warnungen
- Style-Checker
- Fehler-Checker
- Architekturkonformitätsanalyse
- Komplexitäts-Checker
- Speicherverbrauch
- Call-Stack
- Logs
- ...

Success Story



PRIDE



*"Mein Code ist so
schön | gut | effizient
den brauche ich nicht
analysieren lassen."*





Code eines ehemaligen Kollegen

```
// cyclomatic complexity of 74  
// npath complexity of 717.230  
public MyObject doSomeFancyStuff(final String input) {  
  
    ...  
}
```

Lösungsansätze

- Erklären, wieso der Code für andere schwierig zu verstehen ist.

Lösungsansätze

- Erklären, wieso der Code für andere schwierig zu verstehen ist.
- Mit Testfällen den Code der Kolleginnen "kaputt" machen.

Lösungsansätze

- Erklären, wieso der Code für andere schwierig zu verstehen ist.
- Mit Testfällen den Code der Kolleginnen "kaputt" machen.
- Code trotzdem analysieren und dann Problemstellen vorstellen.

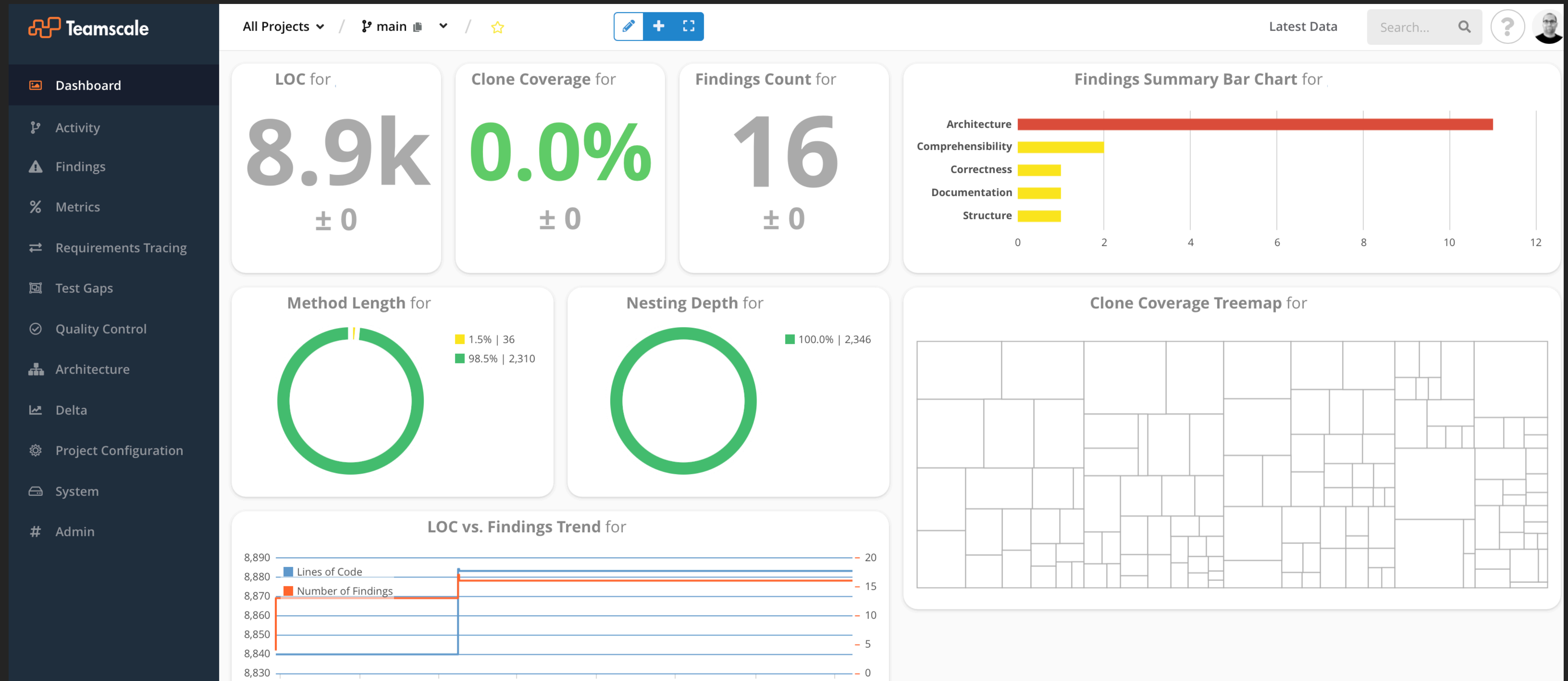
Lösungsansätze

- Erklären, wieso der Code für andere schwierig zu verstehen ist.
- Mit Testfällen den Code der Kolleginnen "kaputt" machen.
- Code trotzdem analysieren und dann Problemstellen vorstellen.

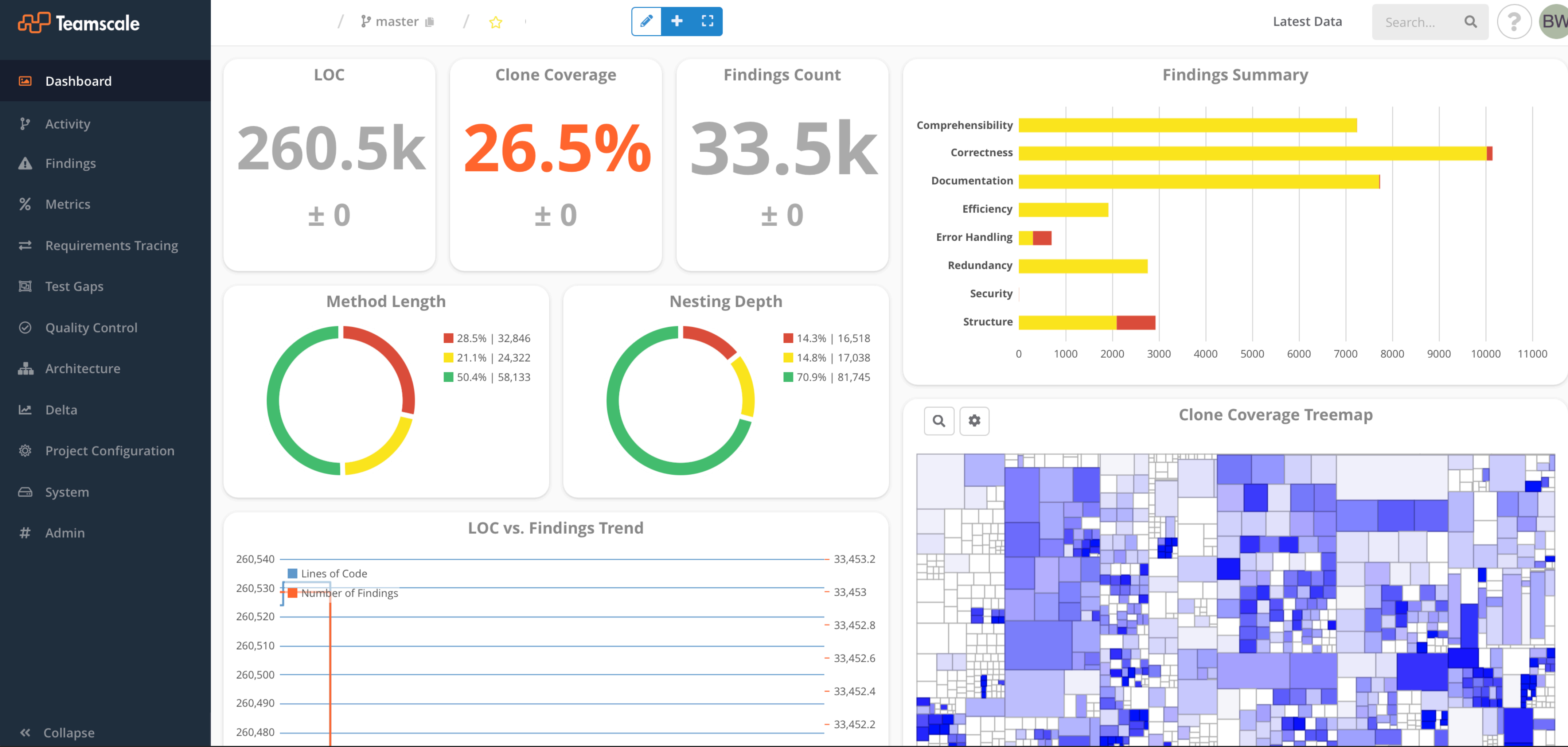
NOTA BENE: Keine Schuldzuweisungen, keine Kritik an der Person üben!

*"Wir haben bei uns
statische
Codeanalyse
eingerrichttet!"*

How it started ...



... how it's going



Lösungsansätze

- Analyse in den Entwicklungsprozess integrieren:
 - In die Pipelines
 - In die IDE
 - In Reviews

Lösungsansätze

- Analyse in den Entwicklungsprozess integrieren:
 - In die Pipelines
 - In die IDE
 - In Reviews
- Verantwortlichkeiten zuweisen, z. B. Tech Leads oder spezifischen Rollen wie einer „Qualitätsingenieurin“.

Lösungsansätze

- Analyse in den Entwicklungsprozess integrieren:
 - In die Pipelines
 - In die IDE
 - In Reviews
- Verantwortlichkeiten zuweisen, z. B. Tech Leads oder spezifischen Rollen wie einer „Qualitätsingenieurin“.
- Technische Schulden (Warnungen) aktiv angehen / reduzieren und Anreize schaffen, dies auch zu tun (z.B. Gamification)



*"Wir haben die
Analysen vor 8 Jahren
eingeführt und
eingerrichtet.*

Die Regeln gelten."

Lösungsansätze

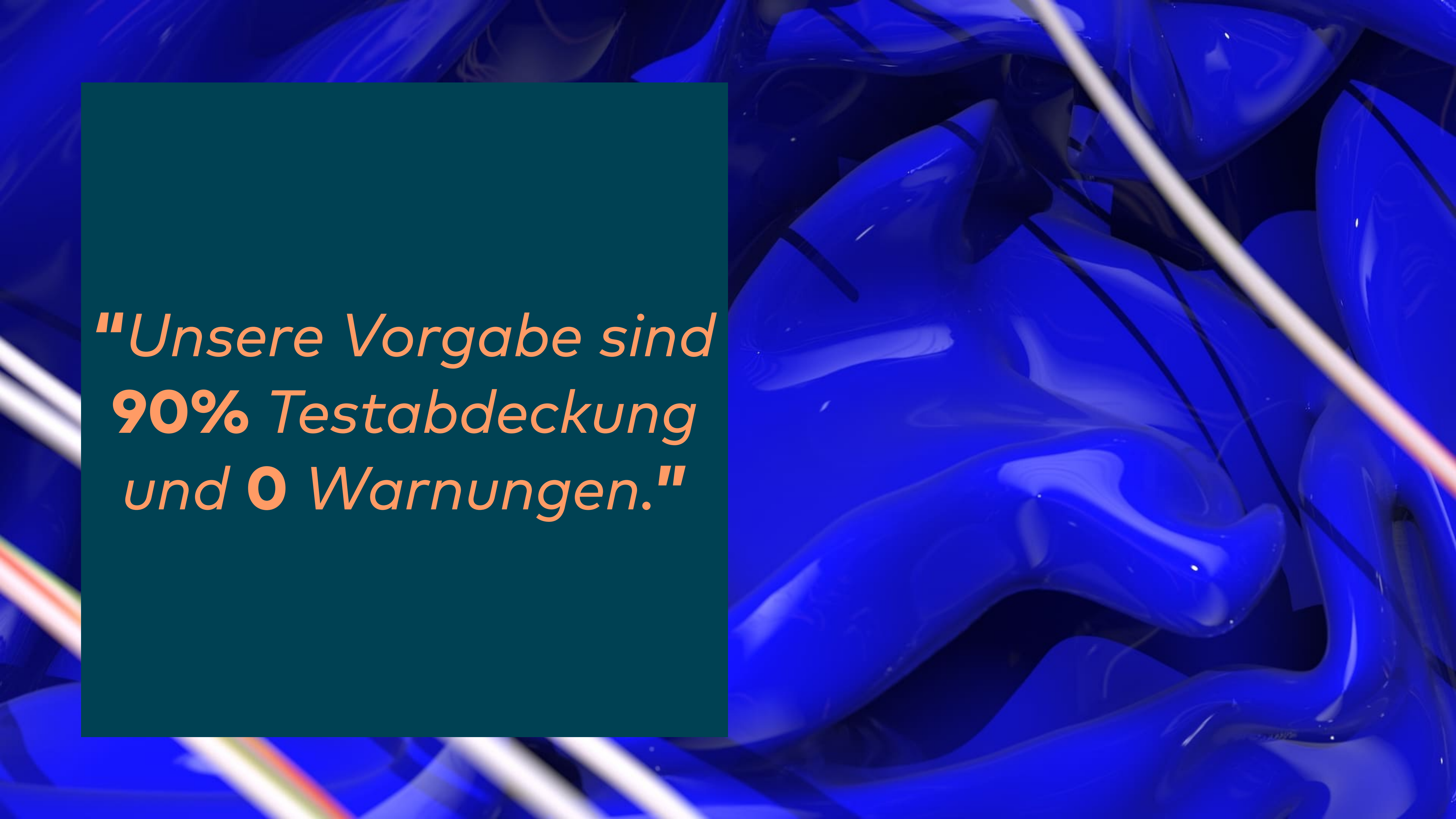
- Aufzeigen, wie sich die (technische) Welt geändert hat:
 - Neue Technologien
 - Neue Technologie-Features
 - Geänderte Programmierparadigmen

Lösungsansätze

- Aufzeigen, wie sich die (technische) Welt geändert hat:
 - Neue Technologien
 - Neue Technologie-Features
 - Geänderte Programmierparadigmen
- Den Austausch suchen und Änderungswünsche begründen

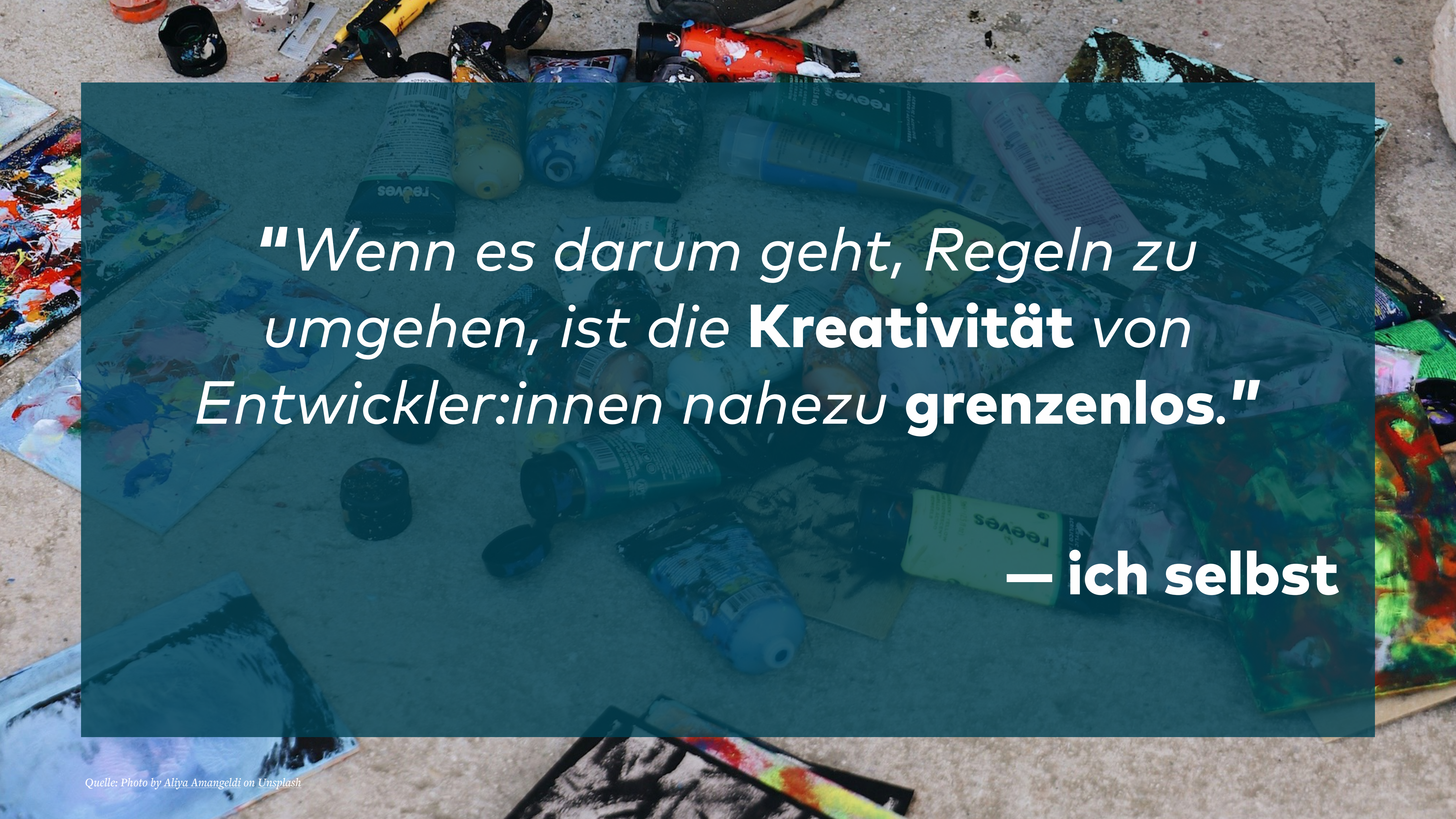
Lösungsansätze

- Aufzeigen, wie sich die (technische) Welt geändert hat:
 - Neue Technologien
 - Neue Technologie-Features
 - Geänderte Programmierparadigmen
- Den Austausch suchen und Änderungswünsche begründen
- Kompromissbereitschaft zeigen



*"Unsere Vorgabe sind
90% Testabdeckung
und **0** Warnungen."*





*"Wenn es darum geht, Regeln zu
umgehen, ist die **Kreativität** von
Entwickler:innen nahezu **grenzenlos**."*

— ich selbst

Lösungsansätze

- Hintergründe der einzelnen Regeln erklären und motivieren

Lösungsansätze

- Hintergründe der einzelnen Regeln erklären und motivieren
- Vorteile aufzeigen, die die Einhaltung der Regeln mit sich bringt

Lösungsansätze

- Hintergründe der einzelnen Regeln erklären und motivieren
- Vorteile aufzeigen, die die Einhaltung der Regeln mit sich bringt
- Sinnvolle Zielvorgaben machen

Lösungsansätze

- Hintergründe der einzelnen Regeln erklären und motivieren
- Vorteile aufzeigen, die die Einhaltung der Regeln mit sich bringt
- Sinnvolle Zielvorgaben machen
- Weniger auf absolute Werte fokussieren, sondern den Trend analysieren



*"Wir nutzen statische
Analysen, daher sind
wir auf der **sicheren**
Seite."*



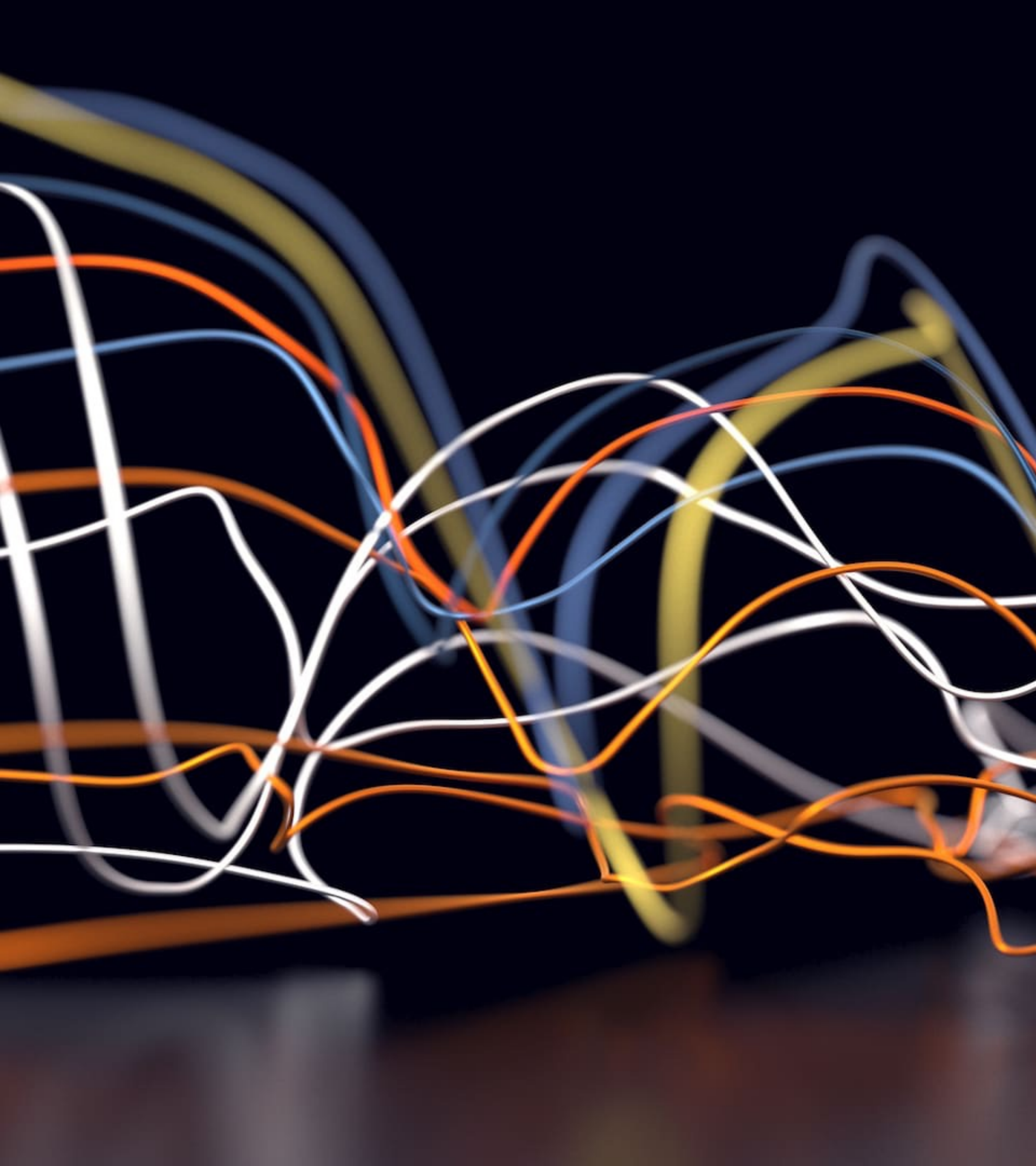
Lösungsansätze

- Statische Codeanalyse ist nur **ein** Werkzeug, um die Qualität zu verbessern.

Lösungsansätze

- Statische Codeanalyse ist nur **ein** Werkzeug, um die Qualität zu verbessern.
- Monitoring, Performance-Tests, dynamische Analysen, Code Reviews usw. sollt ihr auch einsetzen!

PREJUDICE



*"Wir machen jetzt
statische
Codeanalysen und
haben **257.684**
Warnungen in
unserem Projekt."*



A close-up photograph of a black and white cat, likely a tuxedo cat, with striking yellow eyes. The cat is looking directly at the camera with a serious expression. Its fur is predominantly black on the face and ears, with white fur on the chest and muzzle. The background is a soft, out-of-focus green.

WTF Dude!?
Seriously?!



Lösungsansätze

- Regeln im Team festlegen

Lösungsansätze

- Regeln im Team festlegen
- Klaren Fokus setzen

Lösungsansätze

- Regeln im Team festlegen
- Klaren Fokus setzen
- Zahlen immer im aktuellen **Kontext** und **im Team** betrachten

Lösungsansätze

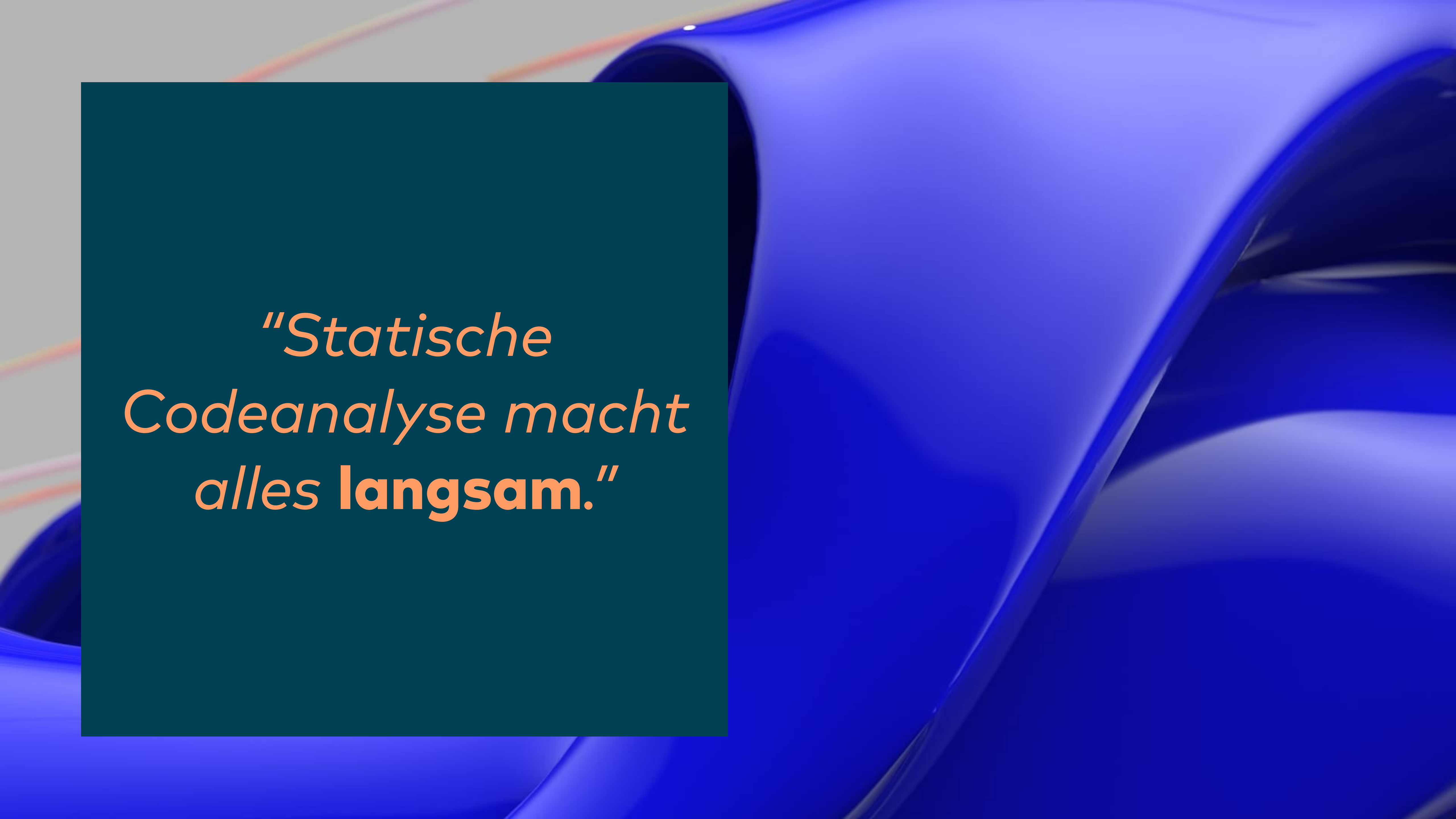
- Regeln im Team festlegen
- Klaren Fokus setzen
- Zahlen immer im aktuellen **Kontext** und **im Team** betrachten
- Zahlen müssen interpretiert werden

Lösungsansätze

- Regeln im Team festlegen
- Klaren Fokus setzen
- Zahlen immer im aktuellen **Kontext** und **im Team** betrachten
- Zahlen müssen interpretiert werden
- Keine absoluten Zahlen (ans Management) weitergeben
 - Der Kontext wird vergessen
 - Die Zahlen bleiben

Lösungsansätze

- Regeln im Team festlegen
- Klaren Fokus setzen
- Zahlen immer im aktuellen **Kontext** und **im Team** betrachten
- Zahlen müssen interpretiert werden
- Keine absoluten Zahlen (ans Management) weitergeben
 - Der Kontext wird vergessen
 - Die Zahlen bleiben
- Es ist kein Kontrollmechanismus, sondern ein Verbesserungswerkzeug



*"Statische
Codeanalyse macht
alles **langsam.**"*



Lösungsansätze

- Mehr automatisieren

Lösungsansätze

- Mehr automatisieren
- Pipelines und Aussagen analysieren (was genau ist langsam?)

Lösungsansätze

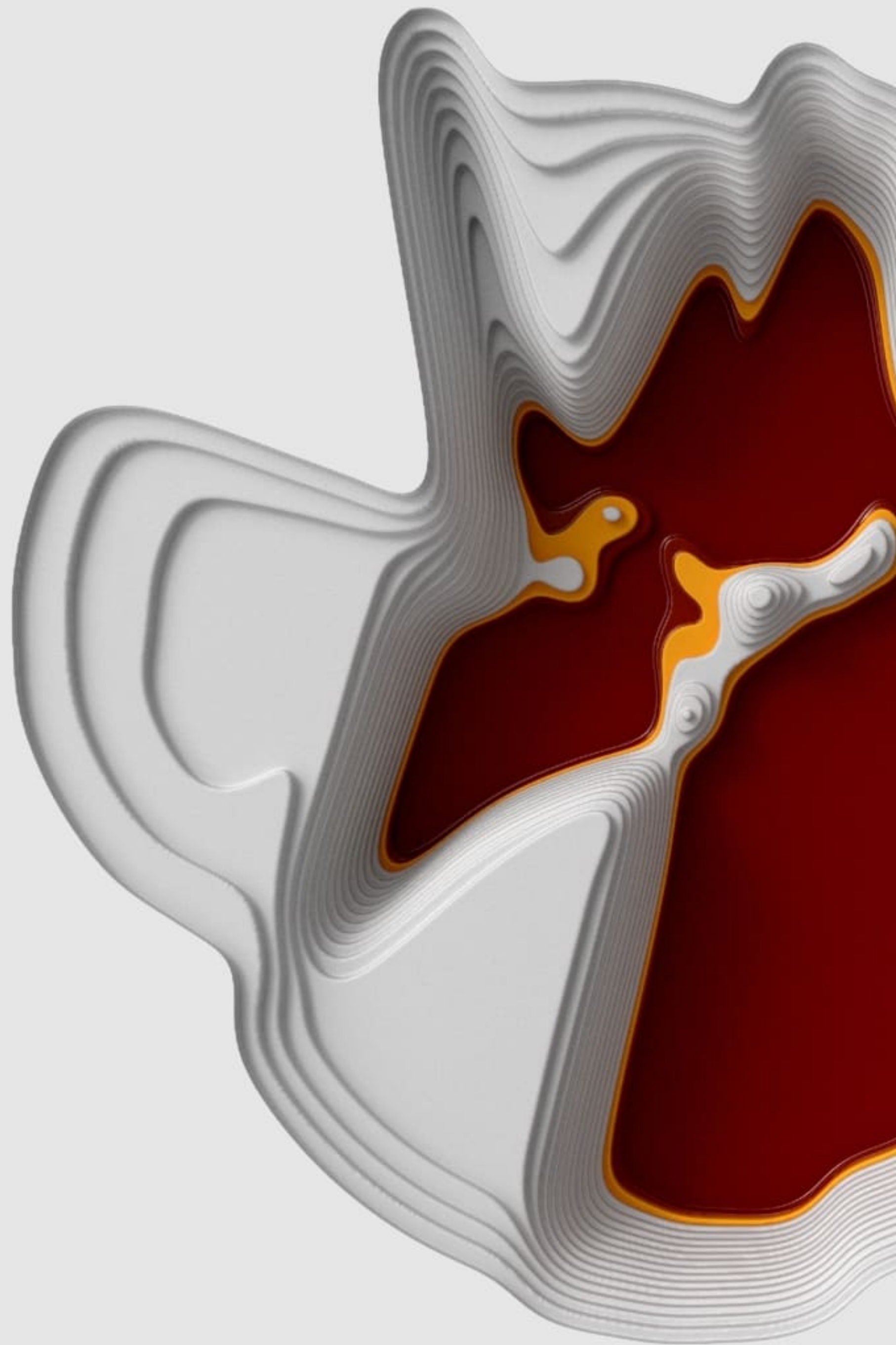
- Mehr automatisieren
- Pipelines und Aussagen analysieren (was genau ist langsam?)
- Tools bestmöglich integrieren

Lösungsansätze

- Mehr automatisieren
- Pipelines und Aussagen analysieren (was genau ist langsam?)
- Tools bestmöglich integrieren
- Alternative Möglichkeiten wie etwa Quality-Pipelines ausprobieren

Lösungsansätze

- Mehr automatisieren
- Pipelines und Aussagen analysieren (was genau ist langsam?)
- Tools bestmöglich integrieren
- Alternative Möglichkeiten wie etwa Quality-Pipelines ausprobieren
- Workflow und Entwicklungsprozess ändern



*"Unser Code lässt
sich nicht so einfach
analysieren."*

*Das Tool macht es
bestimmt **falsch.**"*

IT BEGAN AS A MISTAKE

Lösungsansätze

- Gemeinsam über Regeln sprechen

Lösungsansätze

- Gemeinsam über Regeln sprechen
- Regeln ausschalten oder einzelne Bereiche ausnehmen

Lösungsansätze

- Gemeinsam über Regeln sprechen
- Regeln ausschalten oder einzelne Bereiche ausnehmen
- Für Hintergründe der Regeln sensibilisieren



*"Wir haben
Quality Gates,
die unsere Qualität
sicherstellen."*



**YOU SHALL
NOT PASS**

Lösungsansätze

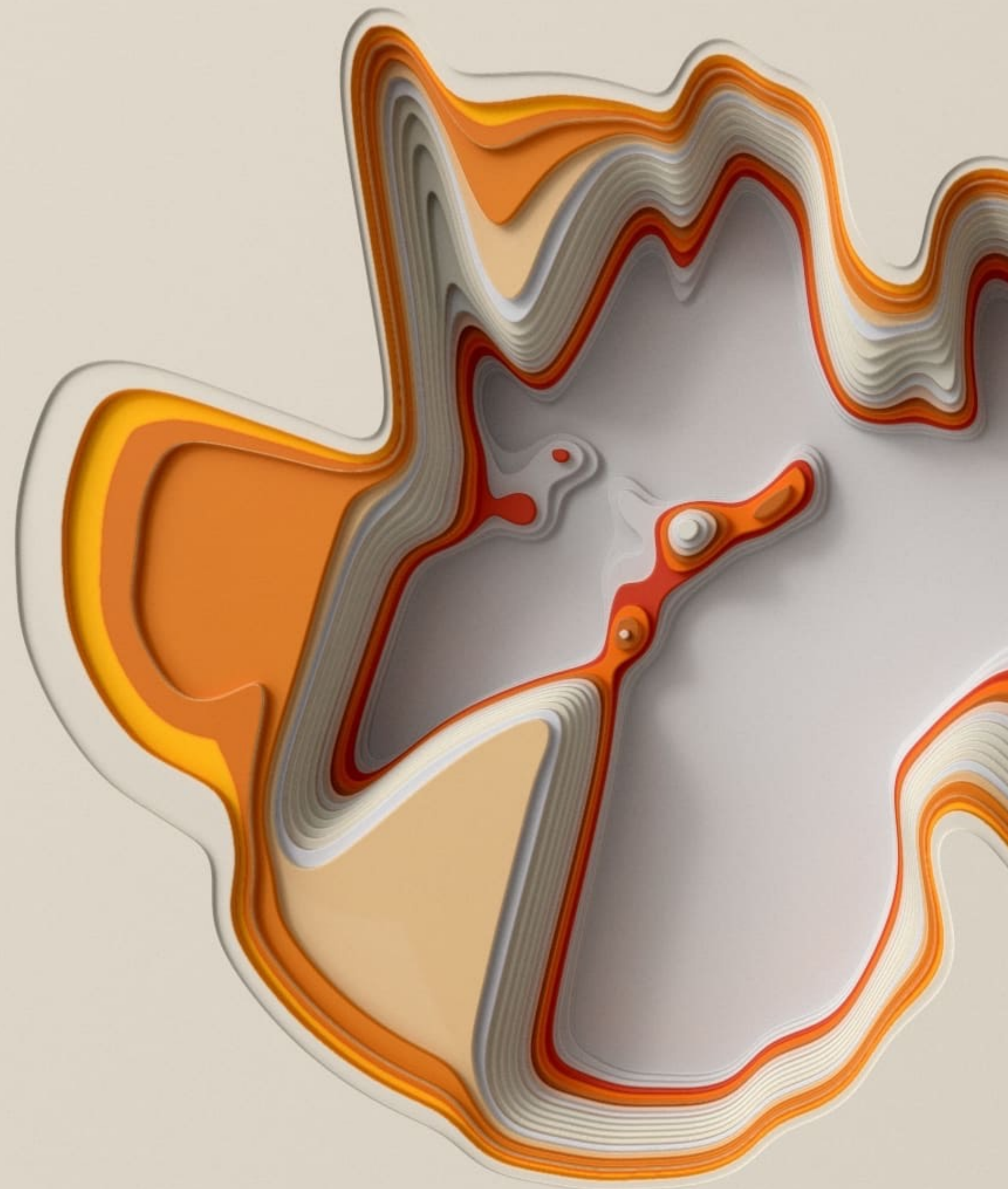
- Gemeinsam über Regeln sprechen

Lösungsansätze

- Gemeinsam über Regeln sprechen
- Teams ermöglichen, spezielle Regelsätze zu haben

Lösungsansätze

- Gemeinsam über Regeln sprechen
- Teams ermöglichen, spezielle Regelsätze zu haben
- Auf ein sinnvolles Mindestset einigen, das für alle Teams gilt



**"Ich führe jetzt alle
Analysen für mein
Team ein."**



Lösungsansätze

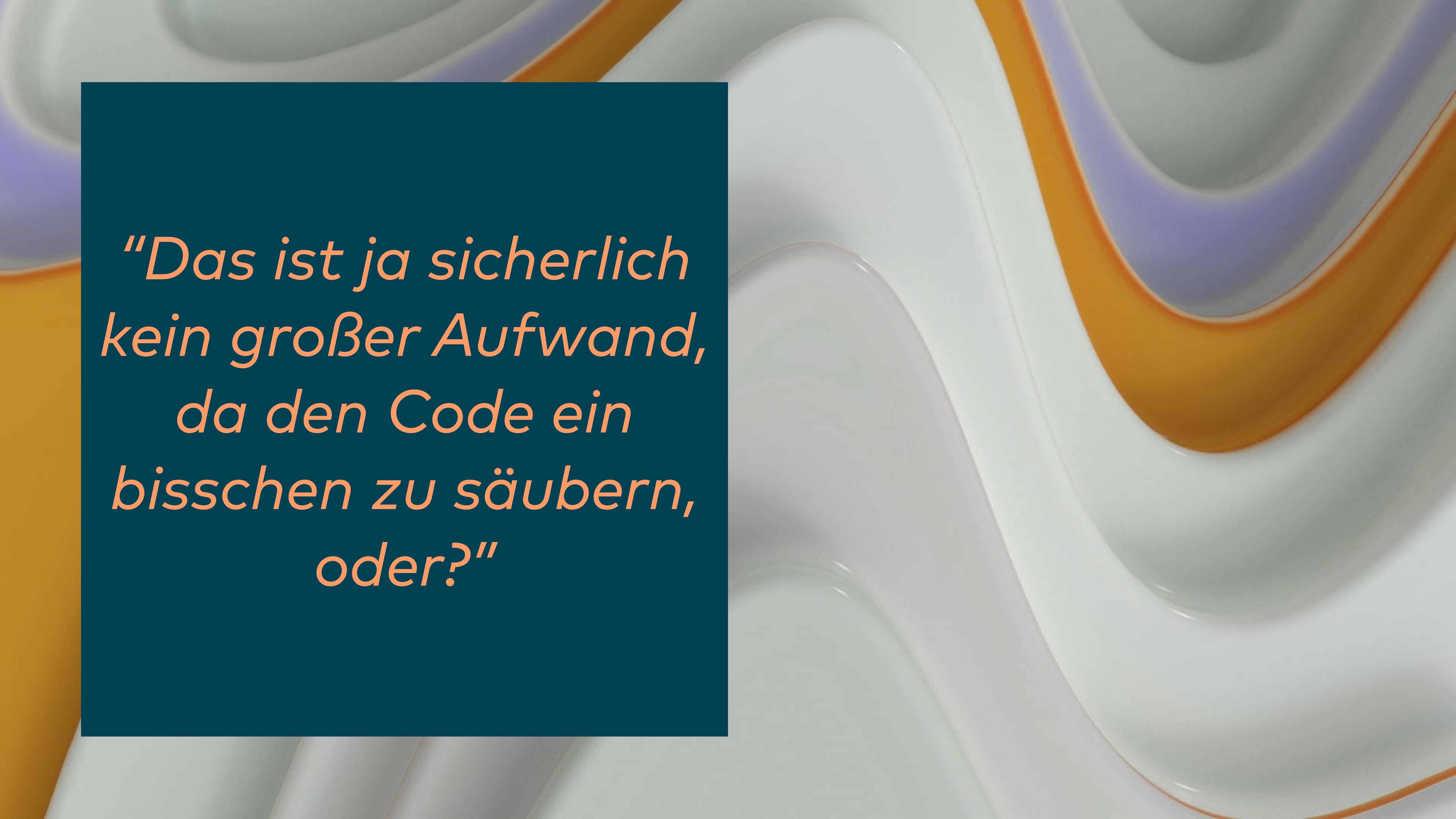
- Kommunikation. Kommunikation. Kommunikation.

Lösungsansätze

- Kommunikation. Kommunikation. Kommunikation.
- Gemeinsam als Team Regelset erarbeiten.

Lösungsansätze

- Kommunikation. Kommunikation. Kommunikation.
- Gemeinsam als Team Regelset erarbeiten.
- Eure Kolleginnen müssen verstehen, **warum** ihr das wollt.



*"Das ist ja sicherlich
kein großer Aufwand,
da den Code ein
bisschen zu säubern,
oder?"*

Doch.

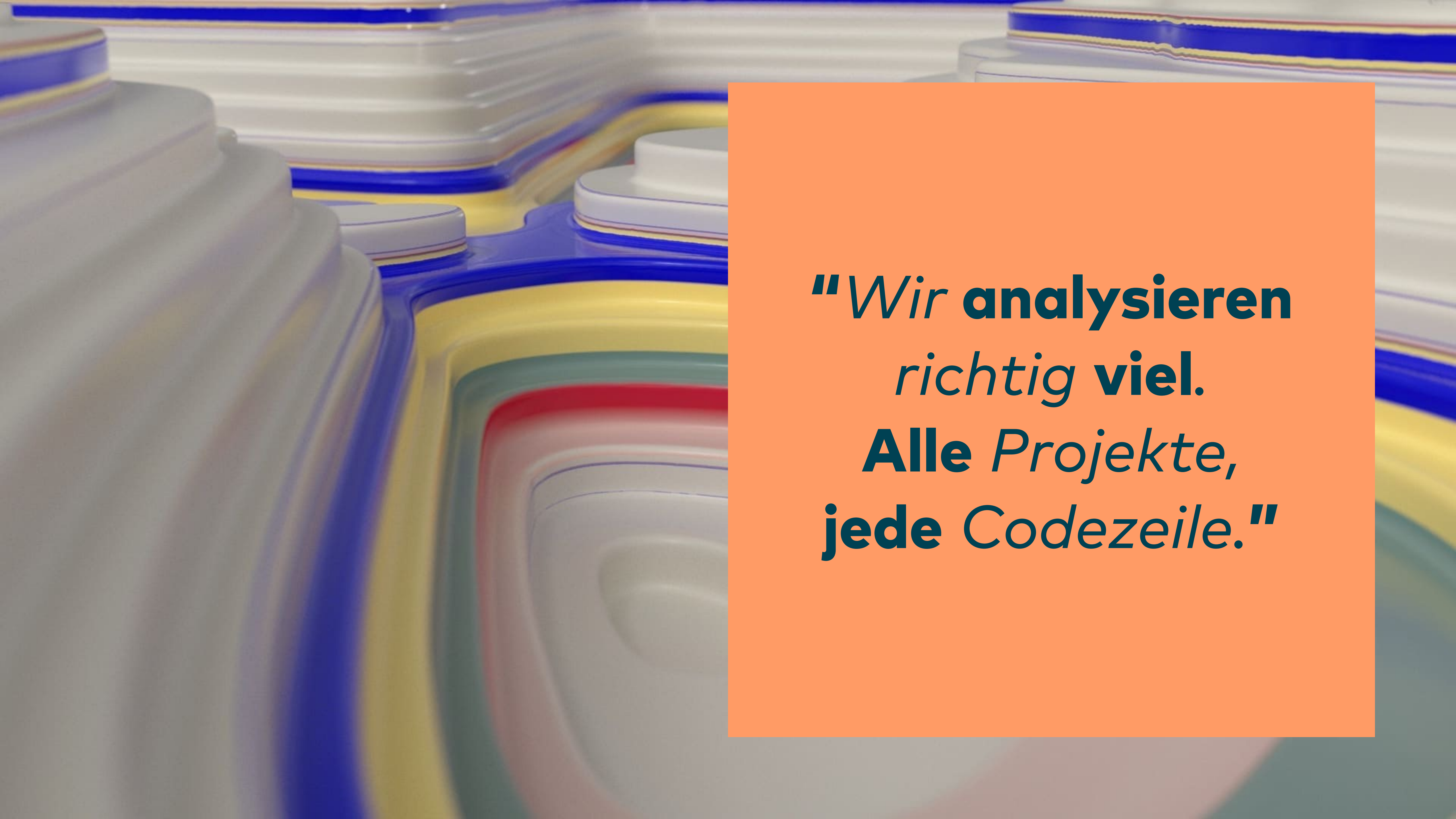
Es ist Arbeit.

**Manches geht schnell,
anderes braucht viel Zeit und Ideen.**

Es ist Arbeit.

**Manches geht schnell,
anderes braucht viel Zeit und Ideen.**

(...auch mit KI)



***"Wir analysieren
richtig viel.
Alle Projekte,
jede Codezeile."***



Lösungsansätze

- Nicht gleich alles auf einmal, fängt klein an und ergänzt iterativ

Lösungsansätze

- Nicht gleich alles auf einmal, fängt klein an und ergänzt iterativ
- Interpretiert die Ergebnisse und leitet Maßnahmen ab

Lösungsansätze

- Nicht gleich alles auf einmal, fängt klein an und ergänzt iterativ
- Interpretiert die Ergebnisse und leitet Maßnahmen ab
- Überprüft die Maßnahmen regelmäßig auf (Miss-)Erfolge

Lösungsansätze

- Nicht gleich alles auf einmal, fängt klein an und ergänzt iterativ
- Interpretiert die Ergebnisse und leitet Maßnahmen ab
- Überprüft die Maßnahmen regelmäßig auf (Miss-)Erfolge
- Rinse and Repeat

AND ZOMBIES ?!



S	W	Name ↓	Last Success
		<u>develop</u>	4 hr 23 min - <u>#35</u>

svn update -r 18907

**Aktuelle Revision:
27649**

```
<fail0nViolation>
```

```
  false
```

```
</fail0nViolation>
```

Takeaways

Takeaways

- Wenige Metriken kontinuierlich während Entwicklung erheben
-> einigt euch als Team darauf!

Takeaways

- Wenige Metriken kontinuierlich während Entwicklung erheben
-> einigt euch als Team darauf!
- Beobachtet die Veränderungen über die Zeit (Trendanalyse).

Takeaways

- Wenige Metriken kontinuierlich während Entwicklung erheben
-> einigt euch als Team darauf!
- Beobachtet die Veränderungen über die Zeit (Trendanalyse).
- Leitet Maßnahmen aus den Ergebnissen ab.

Takeaways

- Wenige Metriken kontinuierlich während Entwicklung erheben
-> einigt euch als Team darauf!
- Beobachtet die Veränderungen über die Zeit (Trendanalyse).
- Leitet Maßnahmen aus den Ergebnissen ab.
- Vergesst nicht den Kontext des Codes:
Die Regeln, die ihr gerade definiert, gab es damals nicht.

Takeaways

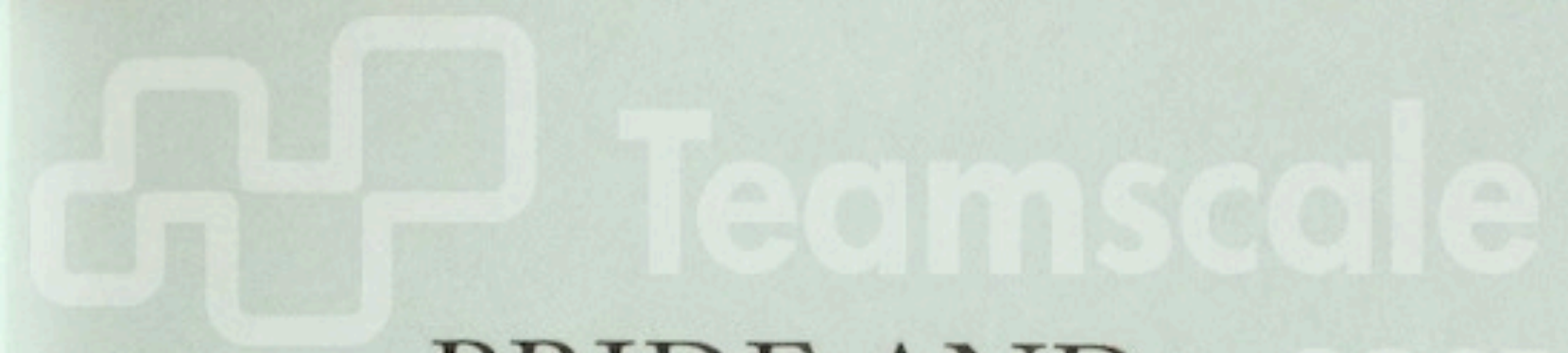
- Wenige Metriken kontinuierlich während Entwicklung erheben
-> einigt euch als Team darauf!
- Beobachtet die Veränderungen über die Zeit (Trendanalyse).
- Leitet Maßnahmen aus den Ergebnissen ab.
- Vergesst nicht den Kontext des Codes:
Die Regeln, die ihr gerade definiert, gab es damals nicht.
- Geht mit der Zeit. Code ändert sich, Technologien ändern sich.
Eure Analysen sollten das auch tun.

Takeaways

- Wenige Metriken kontinuierlich während Entwicklung erheben
-> einigt euch als Team darauf!
- Beobachtet die Veränderungen über die Zeit (Trendanalyse).
- Leitet Maßnahmen aus den Ergebnissen ab.
- Vergesst nicht den Kontext des Codes:
Die Regeln, die ihr gerade definiert, gab es damals nicht.
- Geht mit der Zeit. Code ändert sich, Technologien ändern sich.
Eure Analysen sollten das auch tun.
- Hinterfragt euch und eure Prozesse regelmäßig.

Takeaways

- [illegible]



PRIDE AND by CQSE PREJUDICE

AND ZOMBIES

BENJAMIN WOLF

SOFTWARE INTELLIGENCE TALK

SEPTEMBER 23, 2026

INNOQ

Danke für die Aufmerksamkeit!

Softwarequalität trotz, wegen und mit KI

Wie agentische Entwicklung und verlässliche Softwarequalität zusammenfinden

Mittwoch

11. November

10:30 bis 12:00 Uhr

online und kostenlos – mit Aufzeichnung

Jetzt anmelden: tmscl.me/sit2611-269



Dr. Benjamin Hummel

Agentic Coding jenseits des Hypes

Was CTOs, CEOs und Investoren jetzt wissen müssen

Mittwoch

14. Oktober

10:30 bis 12:00 Uhr

online und kostenlos – mit Aufzeichnung

Jetzt anmelden: tmscl.me/sit2610-269

Contact



Ben Wolf · benjamin.wolf@innoq.com

Dr. Sven Amann · amann@cqse.eu

CQSE GmbH

Centa-Hafenbrädl-Str. 59

81249 Munich