Wie kann KI Qualitätsprobleme in bestehender Software zielgerichtet beheben?

Dr. Benjamin Hummel

CQSE

Was können LLMs heute und wo geht die Reise hin?

Wie kann KI bei der Softwareentwicklung am besten unterstützen?

FORBES › LEADERSHIP › CAREERS

AI Writes Over 25% Of Code At Google—What Does The Future Look Like For Software Engineers?

Jack Kelly Senior Contributor ⓘ

Jack Kelly covers career growth, job market and workplace trends.

Follow

0

Nov 1, 2024, 06:30am EDT

Introducing Devin, the first AI software engineer

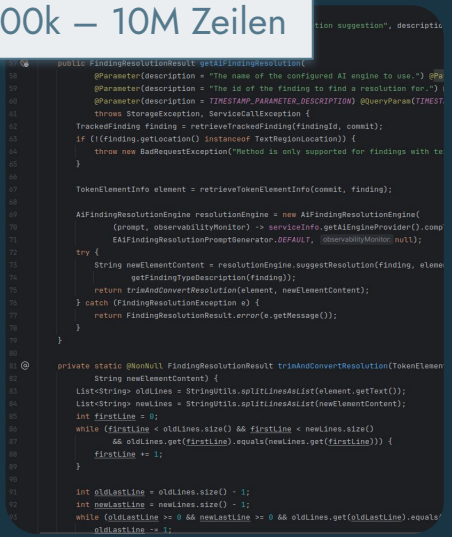
March 12, 2024 • by Scott Wu

Setting a new state of the art on the SWE-bench coding benchmark. Meet Devin, the world's first fully autonomous AI software engineer.

Wir backen erstmal kleinere Brötchen

Statische Analyse

100k – 10M Zeilen



10k Regeln

Regelwerk

- Scanner
- Parser
- AST
- Datenfluss
- Kontrollfluss
- Suffix Trees
- Machine Learning

10k – 10M Findings

Findings

- Unused Code
- Naming Issues
- Klone
- Speicherfehler
- Fehlende Kommentare

Was mache ich jetzt mit den ganzen Findings?

- Baselining + änderungsgetriebenes Aufräumen
- Annotation von Pull-Requests
- Real-Time Feedback in der IDE

Auf gar keinen Fall: Alles (von Hand) Aufräumen

Quick Fix

```
2
3   import java.util.List;
4   import java.util.Set;
5
6   import jakarta.ws.rs.BadRequestException;
7   import jakarta.ws.rs.GET;
8   import jakarta.ws.rs.NotFoundException;
9   import jakarta.ws.rs.Path;
10  import jakarta.ws.rs.PathParam;
11  import jakarta.ws.rs.QueryParam;
12
13  import org.checkerframework.checker.nullness.qual.NonNull;
14  import org.checkerframework.checker.nullness.qual.Nullable;
15  import org.conqat.engine.commons.findings.location.TextRegionLocation;
```

Wo helfen Quick Fixes?

```
import jakarta.ws.rs.NotFoundException;
import jakarta.ws.rs.Path;
import jakarta.ws.rs.ApplicationPath;
impo
impo
import org.checkerframework.checker.nullness.qual.NonNull;
import org.checkerframework.checker.nullness.qual.Nullable;
```

Unused import statement

Remove unused imports Alt+Shift+Enter More actions... Alt+Enter

Flächendeckendes Aufräumen der Codebasis?

Warum haben wir nicht für alles einen Quick Fix?

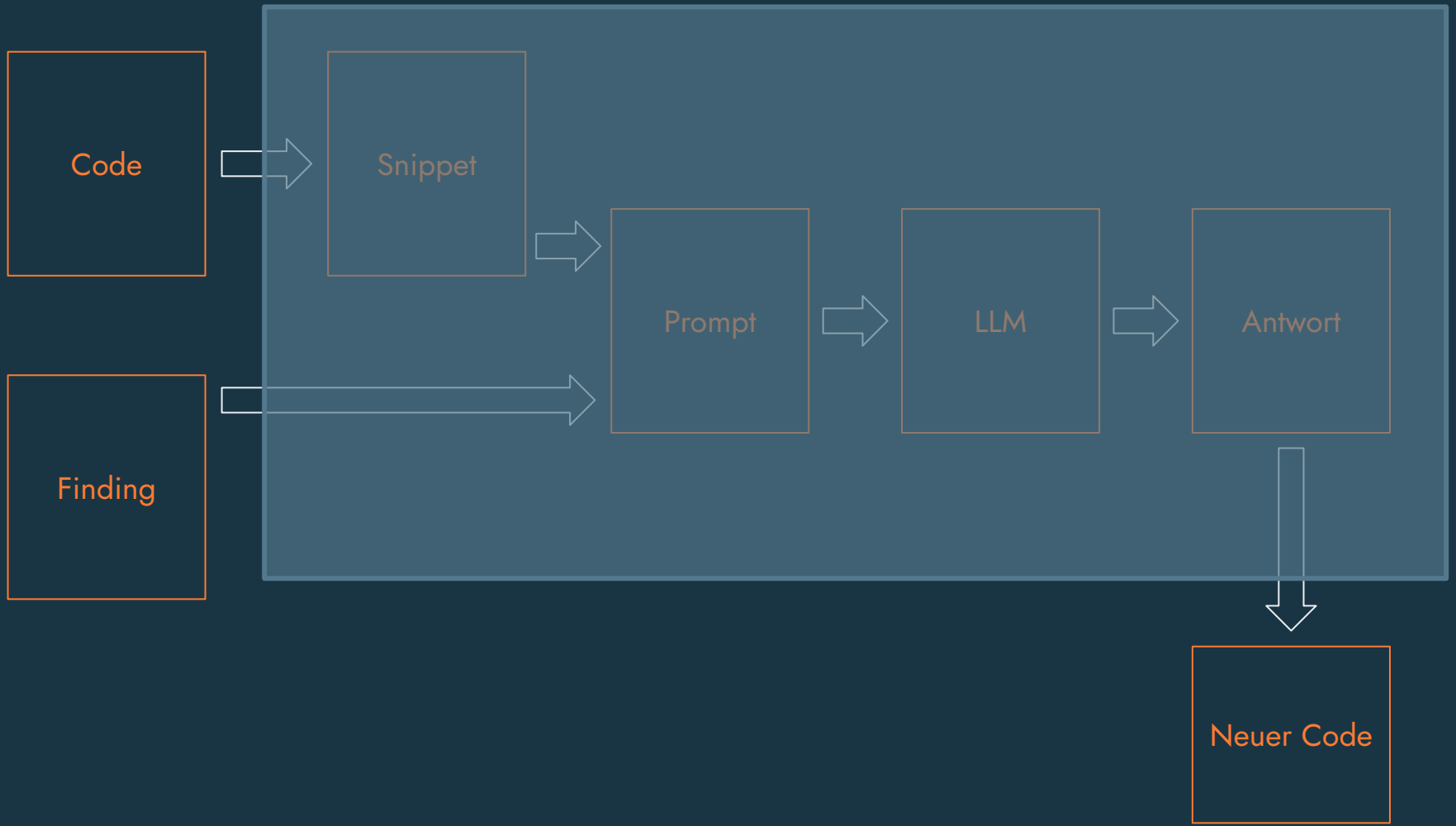
- Quick Fixes sind nicht einfach zu schreiben
- Oft hängt die Lösungsvariante auch vom Kontext ab
- Viele Regeln ⇒ viele Quick Fixes
- Oft reicht reines Codeverständnis nicht aus; man braucht Sprachverständnis

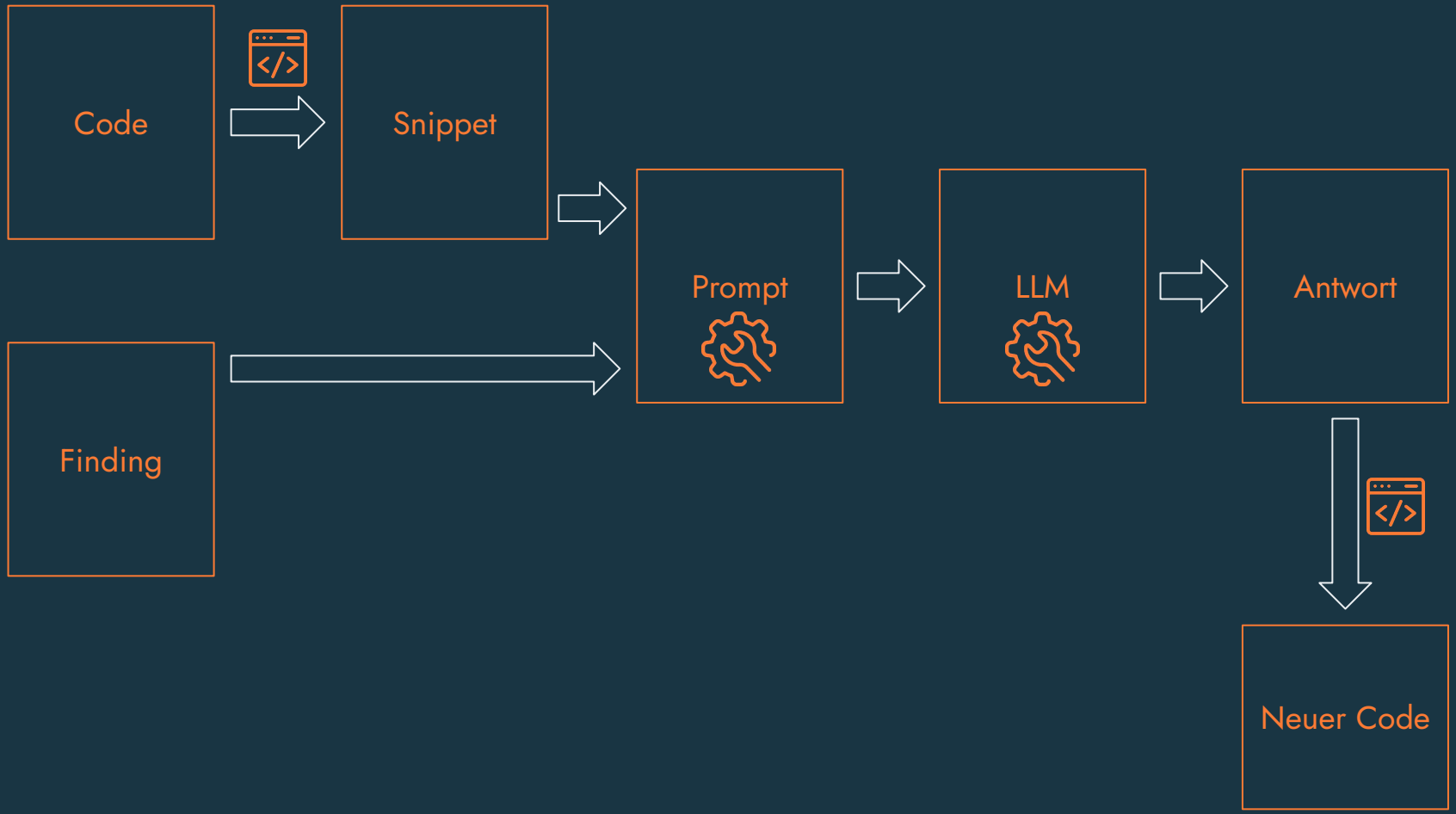
```
/**
 * Trims and converts the resolution of a given token element by comparing it with new element content.
 * It calculates the differences between the old and new content, identifies the differing lines,
 * and then returns a result containing the partial content and diff information.
 *
 * @param element The token element information that contains the original text content.
 * @param newElementContent The new text content to compare against the original element.
 * @return A FindingResolutionResult containing the processed old and new partial contents, their respecti
 */
private static @NonNull FindingResolutionResult trimAndConvertResolution(TokenElementInfo element,  1 usage
```

Kann das nicht alles “dieses AI” für mich machen?

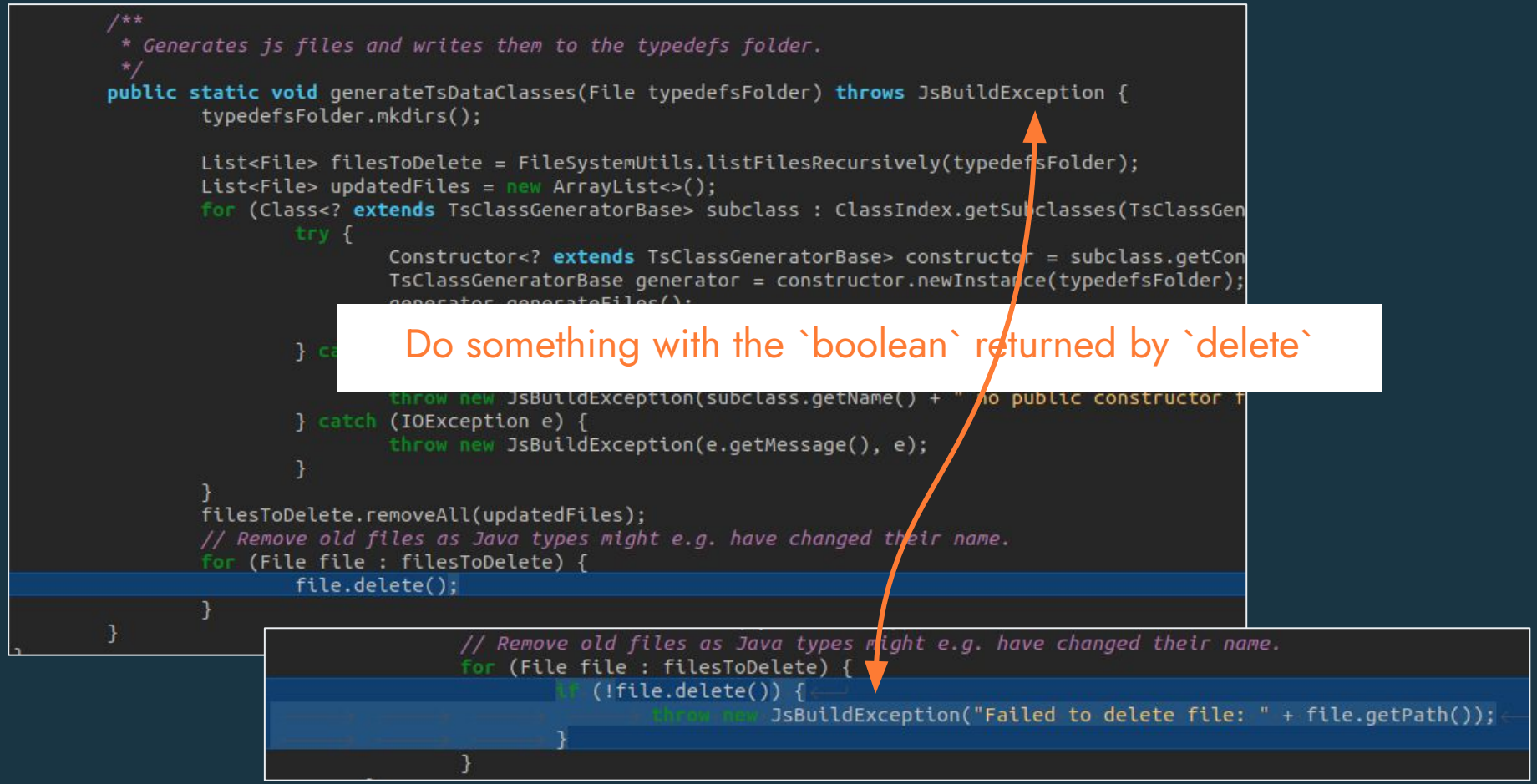
Warum sind Quick Fixes ein guter Startpunkt?

- Klar umrissenes Problem
- Relevante Hilfe bei der Entwicklung
- Überschaubarer Lösungsraum (nicht gleich “komplette Codebasis”)


Code
Finding
Snippet
Prompt
LLM
Antwort
Neuer Code


Code
Snippet
Finding
Prompt
LLM
Antwort
Neuer Code

The Good


/**
* Generates js files and writes them to the typedefs folder.
*/
public static void generateTsDataClasses(File typedefsFolder) throws JsBuildException {
typedefsFolder.mkdirs();
List<File> filesToDelete = FileSystemUtils.listFilesRecursively(typedefsFolder);
List<File> updatedFiles = new ArrayList<>();
for (Class<? extends TsClassGeneratorBase> subclass : ClassIndex.getSubclasses(TsClassGen
try {
Constructor<? extends TsClassGeneratorBase> constructor = subclass.getCon
TsClassGeneratorBase generator = constructor.newInstance(typedefsFolder);
} catch (IOException e) {
throw new JsBuildException(e.getMessage(), e);
}
}
filesToDelete.removeAll(updatedFiles);
// Remove old files as Java types might e.g. have changed their name.
for (File file : filesToDelete) {
file.delete();
}
}
Do something with the `boolean` returned by `delete`
// Remove old files as Java types might e.g. have changed their name.
for (File file : filesToDelete) {
if (!file.delete()) {
throw new JsBuildException("Failed to delete file: " + file.getPath());
}
}

```java
package com.teamscale.index.repository.git.common;

import org.conqat.lib.commons.js_export.ExportToTypeScript;

/**
 * Stores the constant name values for the Voting and Detailed Line Comments
 * Connector Options
 */
@ExportToTypeScript
@SuppressWarnings("unused")
public enum EVotingConnectorOption {

        FINDINGS_INTEGRATION_OPTION(

        TEST_GAP_INTEGRA

        COMMIT_ALERTS_INTEGRATION_OPTION(
                        VoteSupportingGitRepositoryConnectorDescriptorBase.ENABLE_COMMI

        private final String optionName;

        EVotingConnectorOption(String optionName) {
                this.optionName = optionName;
        }

        public String getOptionName() {
                retur
        }
}
```

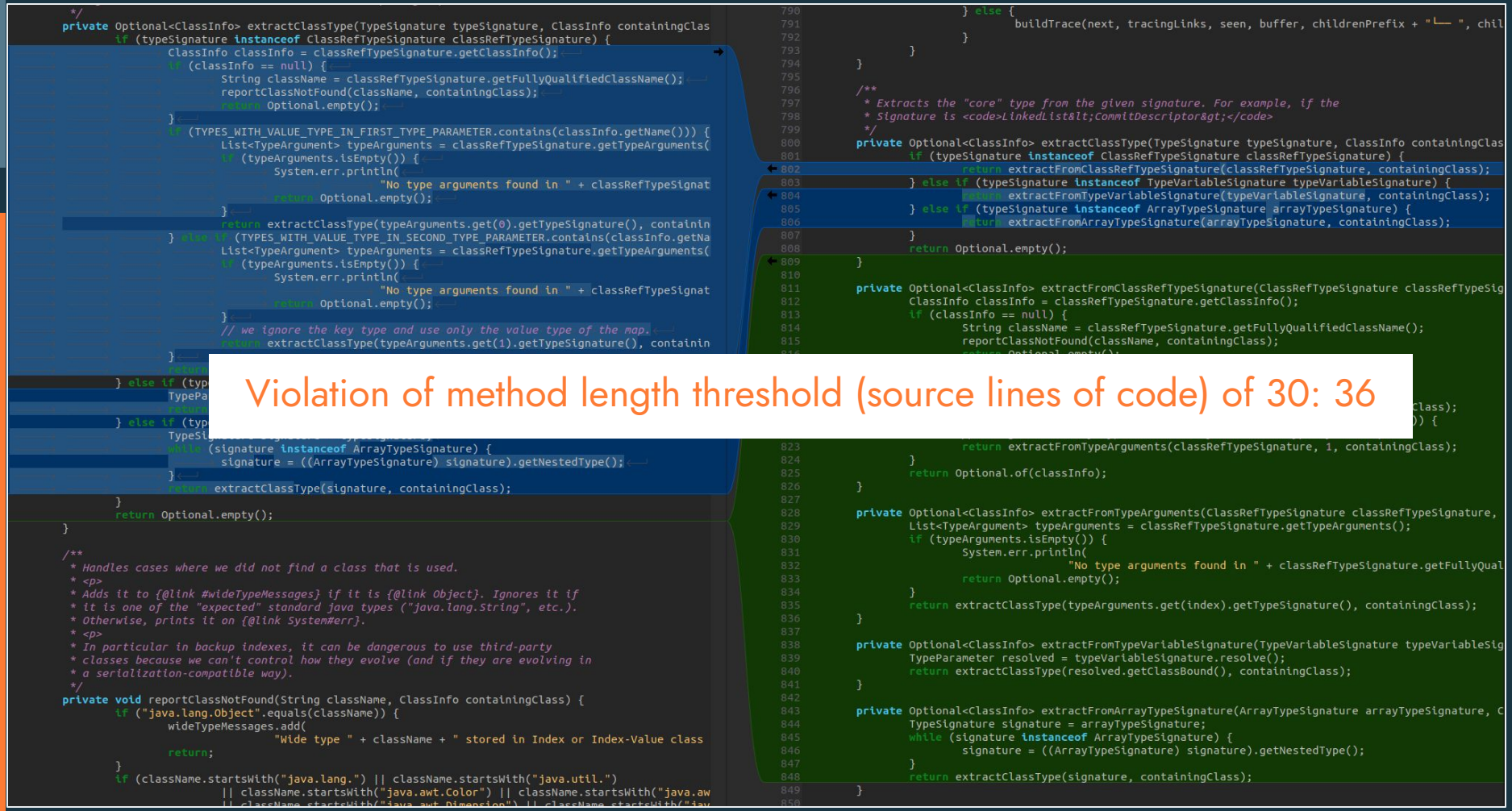
Undocumented SuppressWarnings

```java
@ExportToTypeScript
// Suppress 'unused' warning because the class is used only in TypeScript
@SuppressWarnings("unused")
public enum EVotingConnectorOption {
```

```
    /**
     * Encodes a byte array as a hex string following the method described here:
     * http ://stackoverflow.com/questions/9655181/convert-from-byte-array-to-hex-
     * string-in-java
     */
    public static String encodeAsHex(byte[] data) {
        char[] hexChars = new char[data.length * 2];
        for (int j = 0; j < data.length; j++) {
```

Comment should not contain `//`

The AI model decided that the finding is a false positive.


Violation of method length threshold (source lines of code) of 30: 36

The Bad

```
const line = ' name?: string;';
const styleRegions: StyleRegion[] = [];
let sourceFormatterContext: SourceFormatterContext | null = null;
```

Constant `styleRegions` violates naming conventions.
Should be one of `([A-Z][_A-Z0-9]*)|(E[A-Z][a-zA-Z]+)|([A-Z][a-zA-Z0-9]+Context)`

```
const line = ' name?: string;';
let styleRegions: StyleRegion[] = [];
let sourceFormatterContext: SourceFormatterContext | null = null;
```

```java
@Override
public int hashCode() {
        final int prime = 31;
        int result = super.hashCode();
        result = prime * result + (int) (lastChangeTimestamp ^ (lastChangeTimestamp >>> 32));
        return result;
}
```

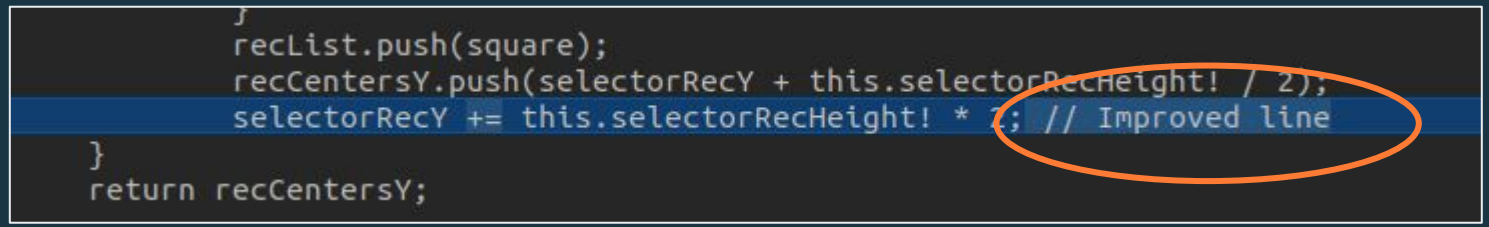
Warum nicht `Objects.hash()`?

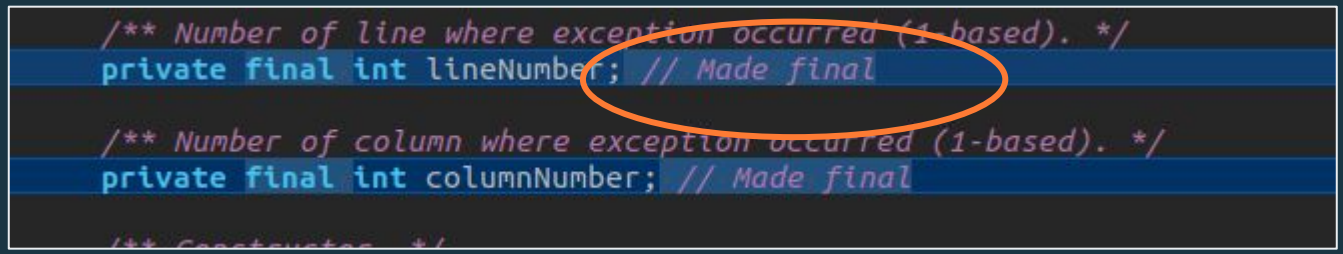
```java
/** {@inheritDoc} */
@Override
public int hashCode() {
        return Objects.hash(super.hashCode(), lastChangeTimestamp);
}
```

```
public @interface ExternalServiceTest {
        /** The URI of the target server (http(s)://hostname or hostname:port). */
        String value();
```

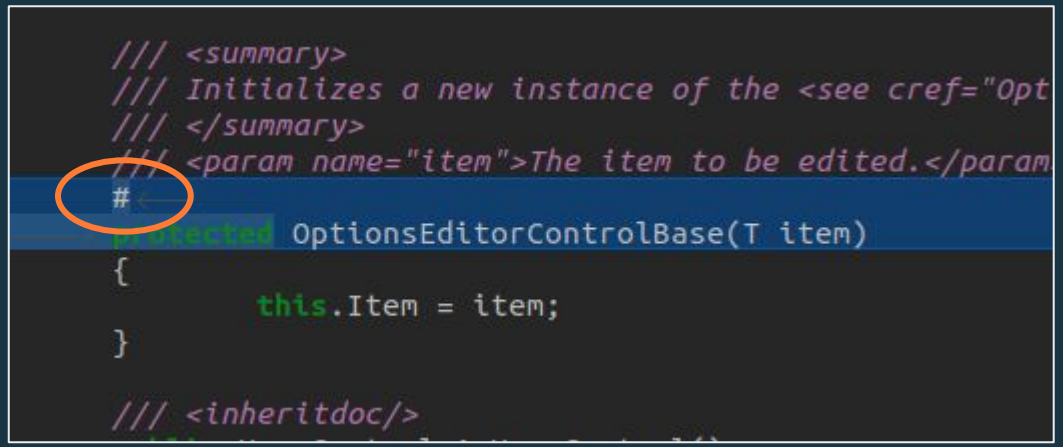
Comment should not contain `//`

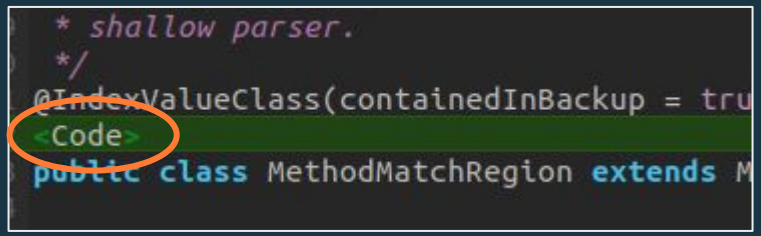
```
public @interface ExternalServiceTest {
        /** The URI of the target server (http(s&#47;&#47;hostname or hostname:port). */
        String value();
```


recList.push(square);
recCentersY.push(selectorRecY + this.selectorRecHeight! / 2);
selectorRecY += this.selectorRecHeight! * 2; // Improved line
}
return recCentersY;


/** Number of line where exception occurred (1-based). */
private final int lineNumber; // Made final
/** Number of column where exception occurred (1-based). */
private final int columnNumber; // Made final

The Ugly


/// <summary>
/// Initializes a new instance of the <see cref="Opt
/// </summary>
/// <param name="item">The item to be edited.</param
#
protected OptionsEditorControlBase(T item)
{
this.Item = item;
}
/// <inheritdoc/>


* shallow parser.
*/
@IndexValueClass(containedInBackup = tru
<Code>
public class MethodMatchRegion extends M

```
87      public static class ClassWithCollections {
88
89              List<String> stringList;
90
91              Integer[] numberArray;
92
93              List<KnownType> listWithKnownType;
94
95              List<UnknownType> listWithObjects;
96
97              @Nullable
98              List<Integer> nullableArray;
99      }
100
```

Interface comment missing (line 89).

```
87      /** A class that contains various collections */
88      public static class ClassWithCollections {
89
90              List<String> stringList;
91
92              Integer[] numberArray;
93
94              List<KnownType> listWithKnownType;
95
96              List<UnknownType> listWithObjects;
97
98              @Nullable
99              List<Integer> nullableArray;
100     }
```

```java
// Remove old files as Java types might e.g. have changed their name.
for (File file : filesToDelete) {
        file.delete();
}
```

Do something with the `boolean` returned by `delete`

```java
// Remove old files as Java types might e.g. have changed their name.
for (File file : filesToDelete) {
        if (!file.delete()) {
                // Handle the failure to delete the file, e.g., log a warning or throw an exception
                System.err.println("Failed to delete file: " + file.getAbsolutePath());
        }
}
```

```
/// <inheritdoc/>
public object Convert(object value, Type targetType, object parameter, System.Globalization.CultureInfo culture)
{
        switch (value as EAssessment?)
        {
                case EAssessment.RED:
                        return Resources.Assessment_Red;
                case EAssessment.YELLOW:
                        return Resources.Assessment_Yellow;
                default:
                        return string.Empty;
        }
}
```

The value of parameter `culture` is never used

```
/// <inheritdoc/>
public object Convert(object value, Type targetType, object parameter, System.Globalization.CultureInfo
{
        switch (value as EAssessment?)
        {
                case EAssessment.RED:
                        return Resources.ResourceManager.GetString("Assessment_Red", culture);
                case EAssessment.YELLOW:
                        return Resources.ResourceManager.GetString("Assessment_Yellow", culture);
                default:
                        return string.Empty;
        }
}
```

ResourceManager gibt es nicht!

Was heißt das denn jetzt?

Das sind ja wieder nur Anekdoten

Benchmarking

Eigene Codebasis

~ 1,6 MLOC

Java
TypeScript
C#

~ 8.600 Findings

Random Sample

100 Findings

Quick Fix Script

Verschiedene LLMs
Verschiedene Prompts

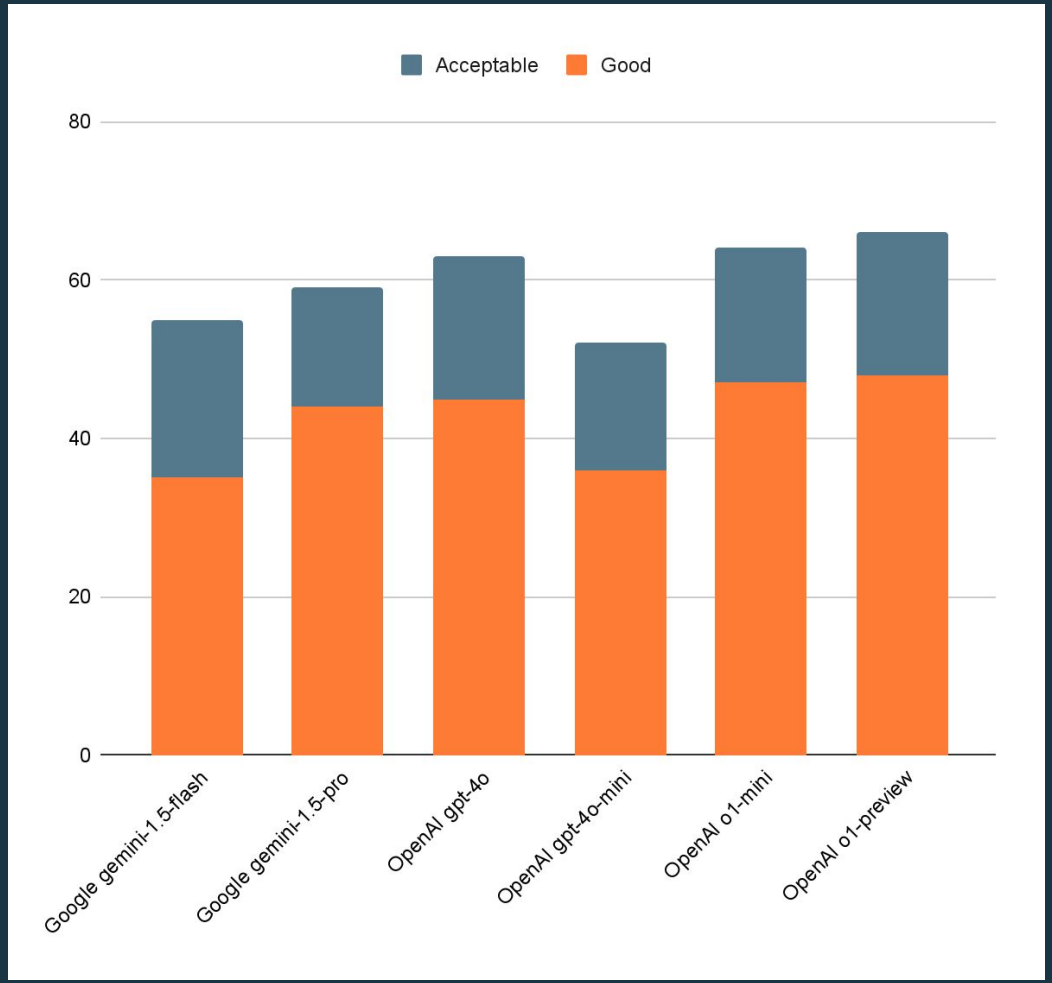
Manuelle Bewertung

Good
Kann man direkt übernehmen

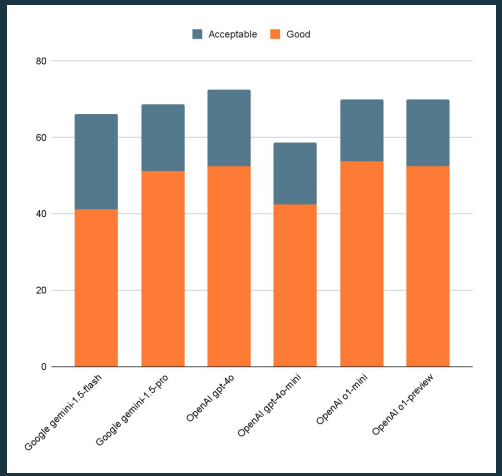
Acceptable
Guter Ansatz, würde man i.d.R. nachbearbeiten

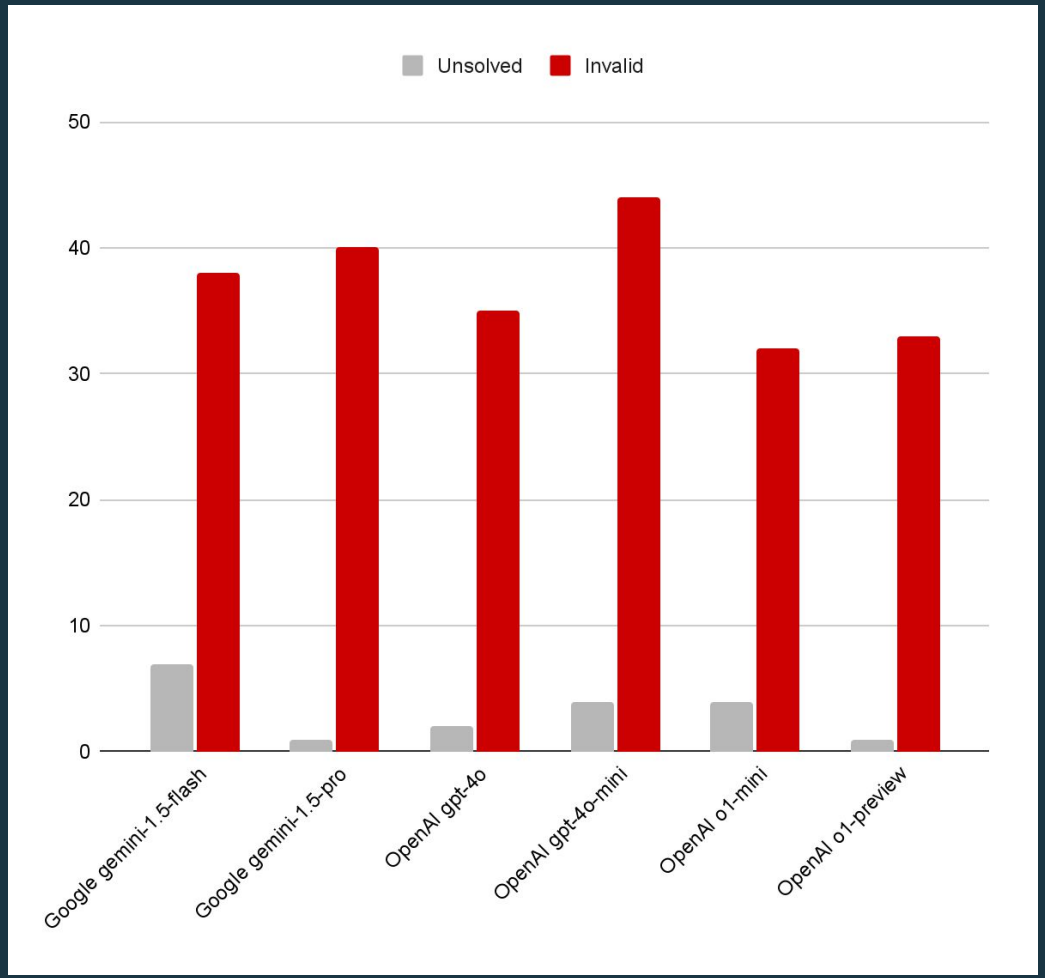
Unsolved
Das LLM sagt, dass es keine Lösung findet

Invalid
Keine gültige Antwort/Lösung

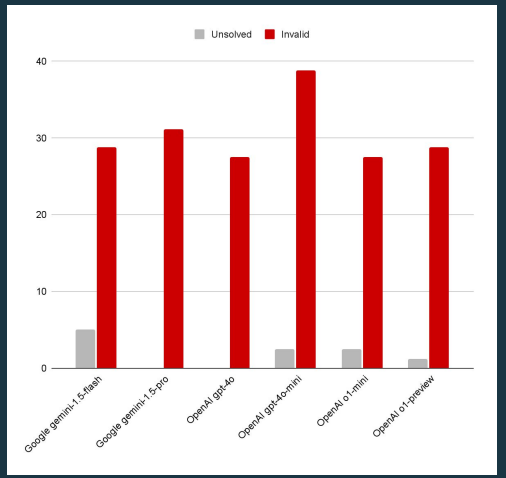


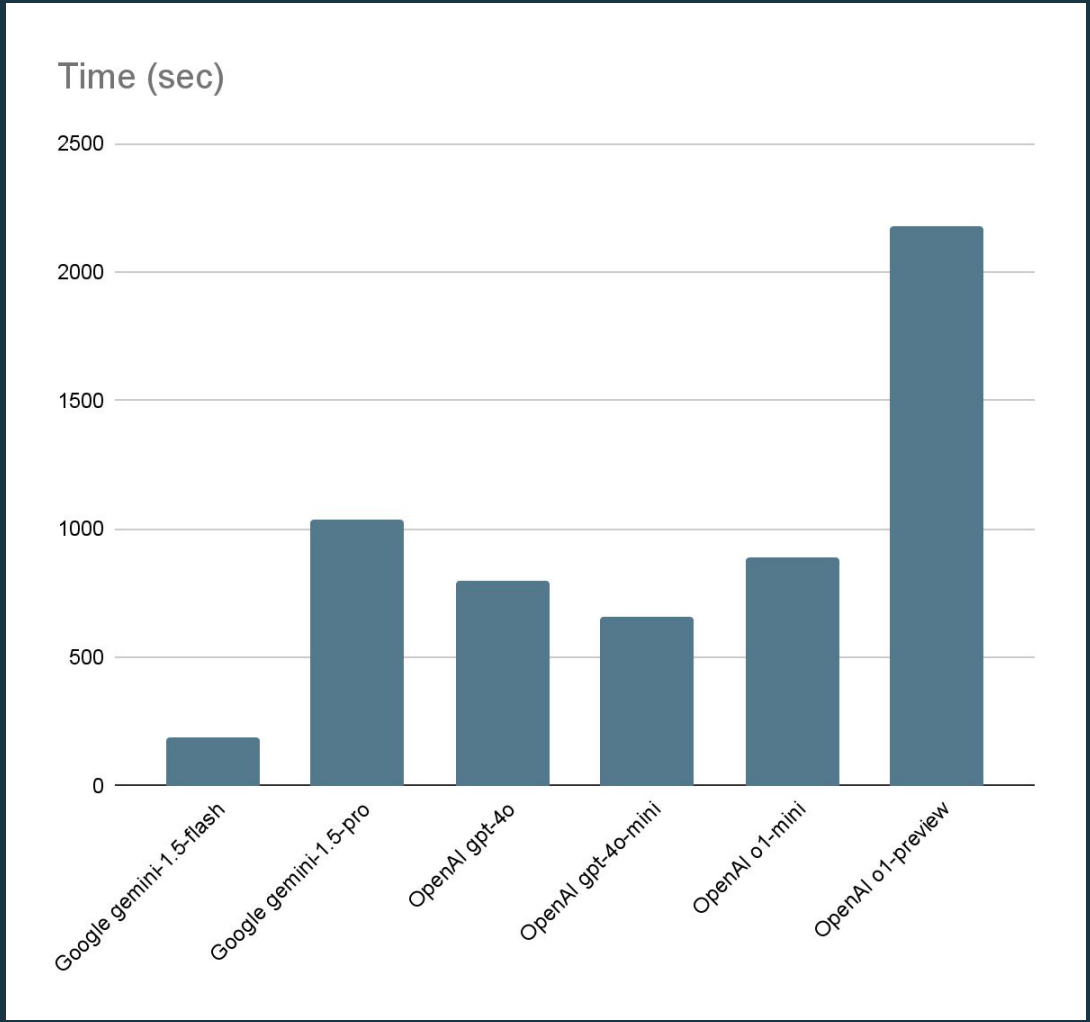
Ohne False Positives

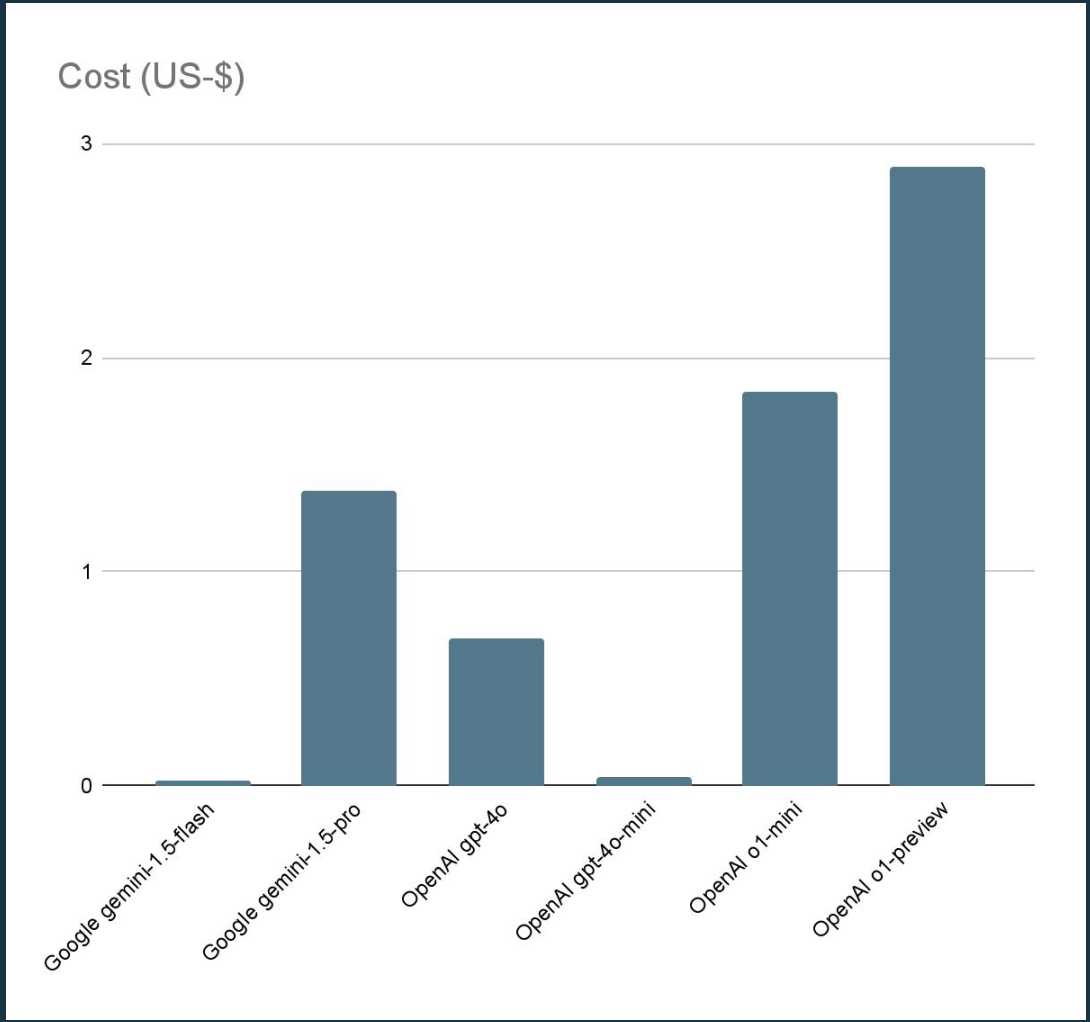




Ohne False Positives




Time (sec)
2500
2000
1500
1000
500
0
Google gemini-1.5-flash
Google gemini-1.5-pro
OpenAI gpt-4o
OpenAI gpt-4o-mini
OpenAI o1-mini
OpenAI o1-preview


Cost (US-$)
3
2
1
0
Google gemini-1.5-flash
Google gemini-1.5-pro
OpenAI gpt-4o
OpenAI gpt-4o-mini
OpenAI o1-mini
OpenAI o1-preview

Fazit

Funktionieren LLM-basierte Quick Fixes?

Flächendeckendes Aufräumen der Codebasis?

Ja, ca. ⅔ der Fixes sind nützlich

Nein, ca. ⅓ ist unbrauchbar

Was können wir noch besser machen?

- Auf bessere Modelle warten ⇐ Wie sicher sind wir, dass die kommen?
- Selbst ein LLM trainieren ⇐ Für "normale" Unternehmen nicht machbar
- Fine-tune Learning ⇐ Braucht viele Daten und macht LLM-Updates schwieriger
- RAG ⇐ Braucht auch viele Daten, aber kann man inkrementell ausbauen

Dr. Benjamin Hummel

linkedin.com/in/benjamin-hummel-cqse

hummel@cqse.eu