KARLSRUHER ENTWICKLERTAG

30 years

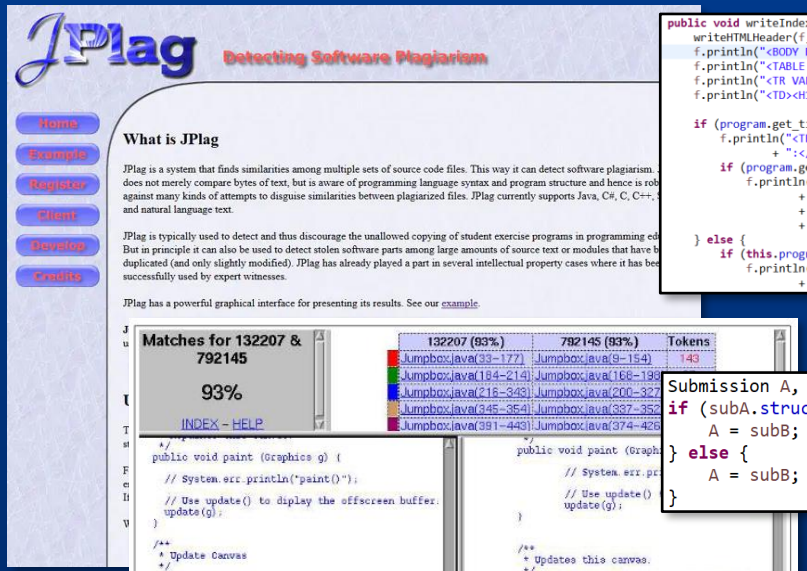
SDQ Software Design and Quality

Reviving JPlag

5 Jahre Open-Source-Modernisierung des weltweit führenden Code-Plagiatsdetektors aus Karlsruhe

Timur Sağlam und Sebastian Hahner

STELL DIR VOR...




```
public void writeIndexBegin(HTMLFile f, String title) {
    writeHTMLHeader(f, title);
    f.println("<BODY BGCOLOR=#ffffff LINK
    f.println("<TABLE ALIGN=center CELLPA
    f.println("<TR VALIGN=middle ALIGN=ce
    f.println("<TD><H1><BIG>" + title + "

    if (program.get_title() != null) {
        f.println("<TR BGCOLOR=\"#aaaaff\
                + ":</BIG></BIG><TD><BIG>
        if (program.get_original_dir() != null)
            f.println("<TR BGCOLOR=#aaaaff VALIGN=top><TD>" + "<BIG>" + msg.getString("Report.Directory")
                    + ":</BIG></TD><TD><BIG><CODE>" + program.get_original_dir()
                    + (program.get_sub_dir() == null ? "" : File.separator + "*" + File.separator + program.get_sub_dir())
                    + "</CODE></BIG></TD></TR>");
    } else {
        if (this.program.get_original_dir() == null)
            f.println("<TR BGCOLOR=#aaaaff VALIGN=top><TD>" + "<BIG><BIG>" + msg.getString("Report.Directory")
                    + ":</BIG></BIG><TD><BIG><BIG><CODE>" + msg.getString("Report.Not_available") + "</CODE></BIG></BIG></TD></TR>");
```

```
public static void usage() {
    System.out.print(Program.name_long
            + ", Copyright (c) 2004-2017 KIT - IPD Tichy, Guido Malpohl, and others.\n"
            + "Usage: JPlag [ options ] <root-dir> [-c file1 file2 ...]\n"
            + " <root-dir>        The root-directory that contains all submissions.\n\n"
            + "options are:\n"
```

```
Submission A, B, tmp;
if (subA.struct.size() > subB.struct.size()) {
    A = subB;  B = subA;
} else {
    A = subB;  B = subA;
}
```

```
case 's': // hidden Option
    this.language = new jplag.javax.Language(null);// WARNING!!!!!BOMB
    this.min_token_match = this.language.min_token_match();
    is.suffixes = this.language.suffixes();
    is.verbose_quiet = true;
    is.exp = true;
    eak;
```

```
inner:          for (int i = 1; i <= elemsB[0]; i++) {// elemsB[0] contains the length of the Array
                    int y = elemsB[i];
                    if (B[y].marked || maxmatch > lengthB - y) continue;

                    int j,hx,hy;
                    for (j = maxmatch - 1; j >= 0; j--) { // begins comparison from behind
                        if (A[hx = x + j].type != B[hy = y + j].type || A[hx].marked || B[hy].marked)
                            continue inner;
                    }
                    // expand match
                    j = maxmatch;
                    while(A[hx = x + j].type == B[hy = y + j].type && !A[hx].marked && !B[hy].marked)
                        j++;

                    if (j != maxmatch) {  // new biggest match? -> delete current smaller
```

```
public void create_hashes(Structure s, int hashLength, boolean makeTable) {
    // Hier wird die obere Grenze der Hash-Laenge festgelegt.
    // Sie ist bestimmt durch die Bitzahl des 'int' Datentyps und der Anzahl
    // der Token.
    if (hashLength<1) hashLength = 1;
    hashLength = (hashLength<26 ? hashLength : 25);
```

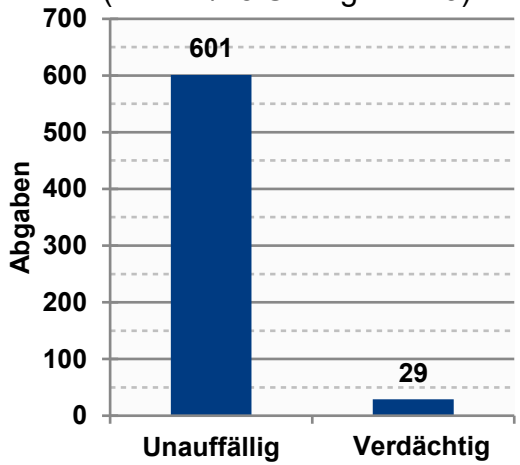
Plagiarismus in der Informatikbildung

The New York Times

As Computer Coding Classes Swell, So Does Cheating (2017)

A Computer Science 50 course at Harvard in 2013. Last fall, more than 60 computer science students were referred to the university's honor council, which investigates cheating allegations. Joseph Ong

Plagiatsprüfung
KIT-Vorlesung „*Programmieren*“
(WS-24/45 Übungsblatt 3)



nature

Explore content | About the journal | Publish with us | Subscribe

nature › news feature › article

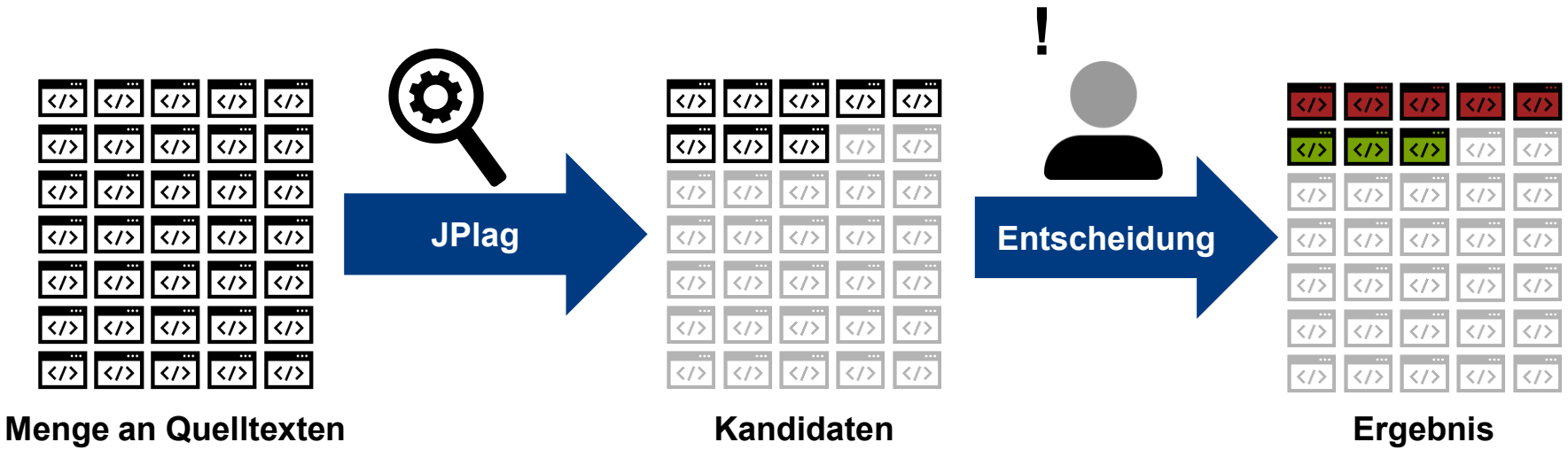
NEWS FEATURE | 30 July 2024

AI is complicating plagiarism... How should scientists respond?

The explosive uptake of generative artificial intelligence in writing is raising difficult questions about when use of the technology should be allowed.

By Diana Kwon

TL;DR: JPlag


JPlag
!
Entscheidung
Menge an Quelltexten
Kandidaten
Ergebnis

Herausforderung Software-Plagiarismus

Es ist unbekannt, …

… ob überhaupt Plagiate vorliegen.

… welche Programmabschnitte übereinstimmen.

Plagiatserkennung ist ein Skalierungsproblem!

- 30 Studierende ⇒ 435 paarweise Vergleiche
- 200 Studierende ⇒ 19.900 paarweise Vergleiche
- 630 Studierende ⇒ 198.135 paarweise Vergleiche

Suche nach der Nadel im Heuhaufen

Obfuscation Attacks

Welche strukturellen Änderungen erschweren die Erkennung?

```
1  void printSquares() {
2    int i = 0;
3
4    while (i <= 10) {
5      int square = i * i;
6      i++;
7      System.out.println(square);
8    }
9  }
```

Code Injection →

Statement Reordering →

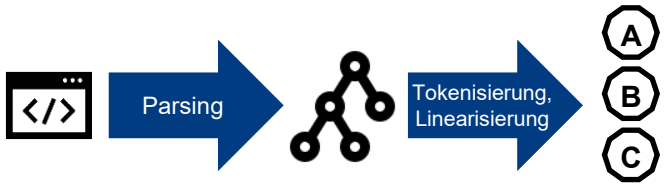
```
1  void printSquares() {
2    int i = 0;
3    boolean debug = false;
4    while (i <= 10) {
5      int square = i * i;
6      System.out.println(square);
7      i++;
8    }
9  }
```

Semantisch neutrale Transformation

- Einfügen von toten Anweisungen
- Umordnung zweier unabhängiger Anweisungen

⇒ leicht automatisierbar

Wie funktioniert JPlag?



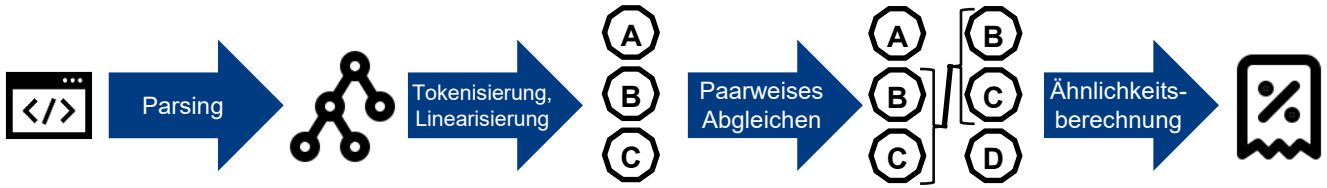
Eingabeprogramme Syntaxbaum Token-Sequenz

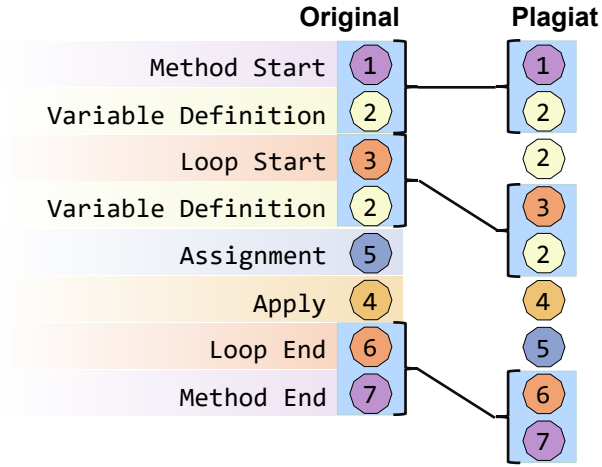
Token sind eine *Abstraktion* von Programmcode:

Code-Plagiatsdetektoren vergleichen die Programmstruktur.

```
1  voi                                          
2    i                                          on
3    boolean debug = false;                     2  Variable Definition
4    while (i <= 10) {                          3  Loop Start
5      int square = i * i;                      2  Variable Definition
6      System.out.println(square);              4  Apply
7      i++;                                     5  Assignment
8    }                                          6  Loop End
9  }                                            7  Method End
```

Wie funktioniert JPlag?

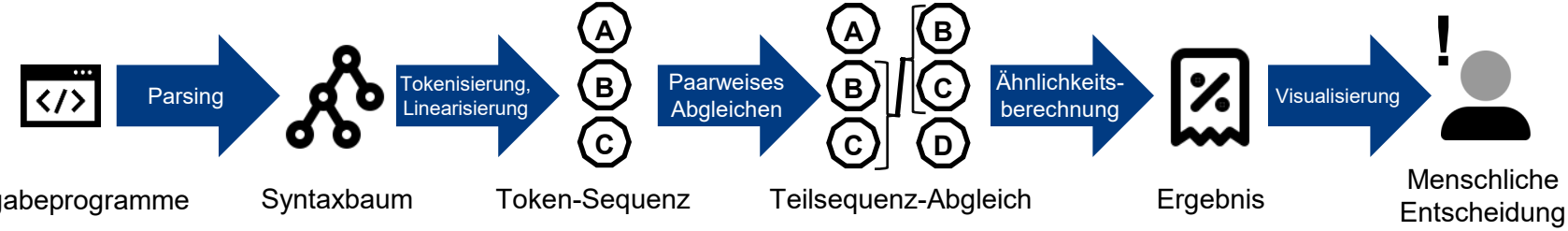


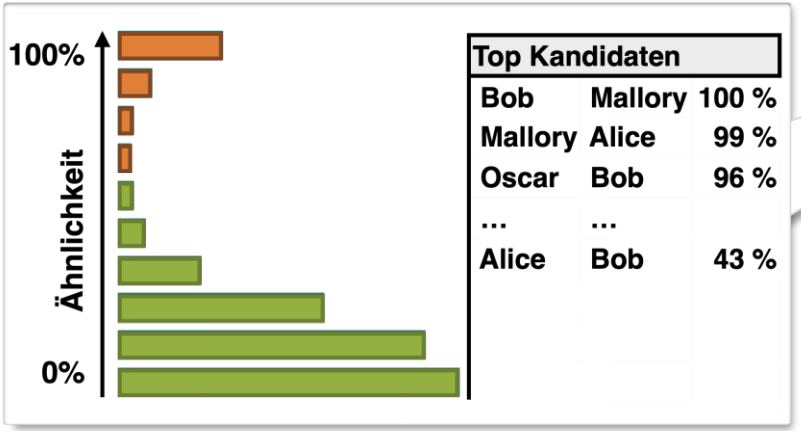


⇒ 71 % Ähnlichkeit

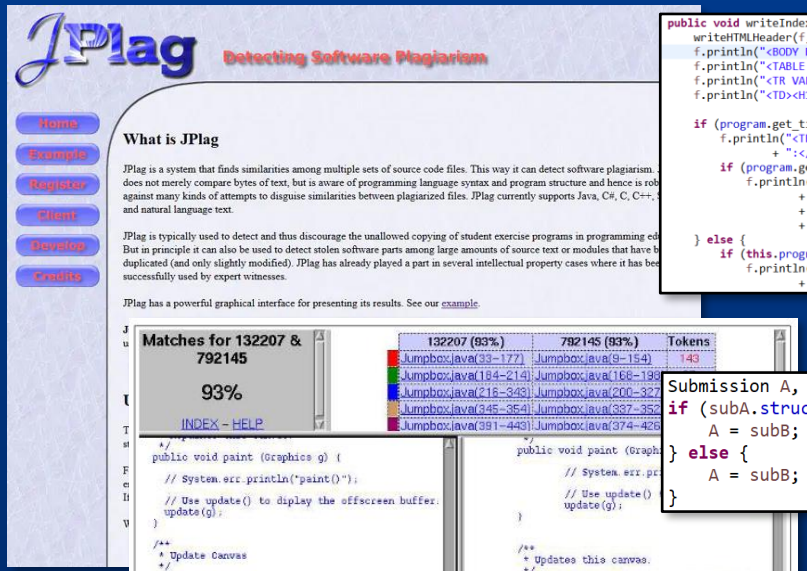
(Vereinfachte Darstellung)

Wie funktioniert JPlag?





! Ein Plagiatsdetektor ermittelt nur verdächtige Kandidaten.




```
public void writeIndexBegin(HTMLFile f, String title) {
    writeHTMLHeader(f, title);
    f.println("<BODY BGCOLOR=#ffffff LINK
    f.println("<TABLE ALIGN=center CELLPA
    f.println("<TR VALIGN=middle ALIGN=ce
    f.println("<TD><H1><BIG>" + title + "

    if (program.get_title() != null) {
        f.println("<TR BGCOLOR=\"#aaaaff\
                + ":</BIG></BIG><TD><BIG>
        if (program.get_original_dir() != null)
            f.println("<TR BGCOLOR=#aaaaff VALIGN=top><TD>" + "<BIG>" + msg.getString("Report.Directory")
                    + ":</BIG></TD><TD><BIG><CODE>" + program.get_original_dir()
                    + (program.get_sub_dir() == null ? "" : File.separator + "*" + File.separator + program.get_sub_dir())
                    + "</CODE></BIG></TD></TR>");
    } else {
        if (this.program.get_original_dir() == null)
            f.println("<TR BGCOLOR=#aaaaff VALIGN=top><TD>" + "<BIG><BIG>" + msg.getString("Report.Directory")
                    + ":</BIG></BIG><TD><BIG><BIG><CODE>" + msg.getString("Report.Not_available") + "</CODE></BIG></BIG></TD></TR>");
```

```
public static void usage() {
    System.out.print(Program.name_long
            + ", Copyright (c) 2004-2017 KIT - IPD Tichy, Guido Malpohl, and others.\n"
            + "Usage: JPlag [ options ] <root-dir> [-c file1 file2 ...]\n"
            + " <root-dir>        The root-directory that contains all submissions.\n\n"
            + "options are:\n"
```

```
Submission A, B, tmp;
if (subA.struct.size() > subB.struct.size()) {
    A = subB;  B = subA;
} else {
    A = subB;  B = subA;
}
```

```
case 's': // hidden Option
    this.language = new jplag.javax.Language(null);// WARNING!!!!!BOMB
    this.min_token_match = this.language.min_token_match();
    is.suffixes = this.language.suffixes();
    is.verbose_quiet = true;
    is.exp = true;
    eak;
```

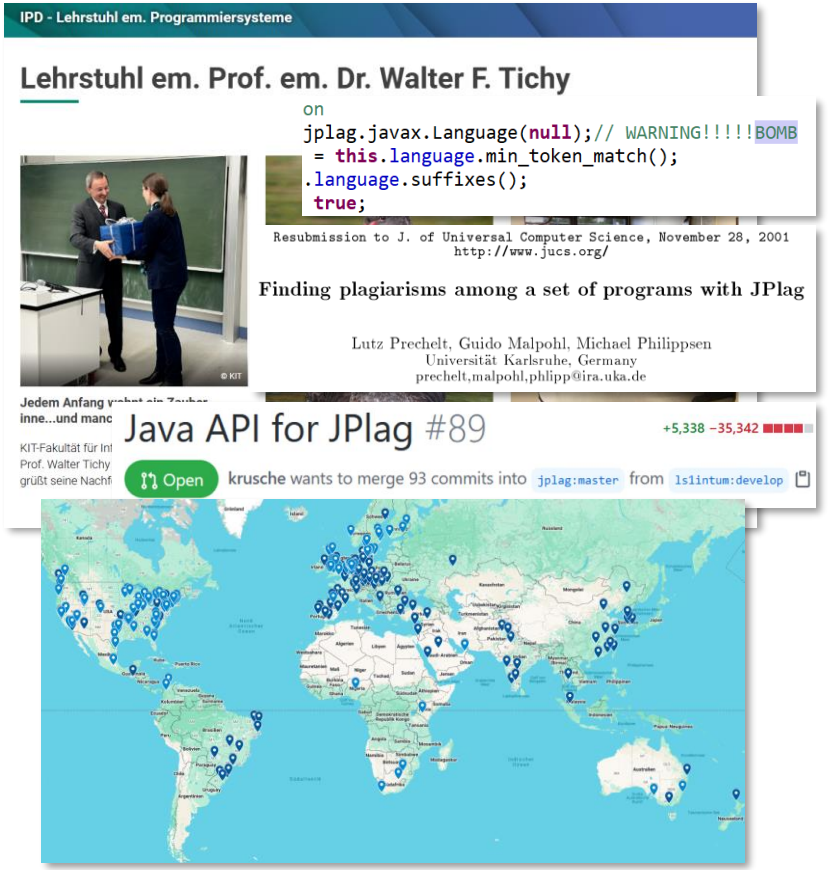
```
inner:          for (int i = 1; i <= elemsB[0]; i++) {// elemsB[0] contains the length of the Array
                    int y = elemsB[i];
                    if (B[y].marked || maxmatch > lengthB - y) continue;

                    int j,hx,hy;
                    for (j = maxmatch - 1; j >= 0; j--) { // begins comparison from behind
                        if (A[hx = x + j].type != B[hy = y + j].type || A[hx].marked || B[hy].marked)
                            continue inner;
                    }
                    // expand match
                    j = maxmatch;
                    while(A[hx = x + j].type == B[hy = y + j].type && !A[hx].marked && !B[hy].marked)
                        j++;

                    if (j != maxmatch) {  // new biggest match? -> delete current smaller
```

```
public void create_hashes(Structure s, int hashLength, boolean makeTable) {
    // Hier wird die obere Grenze der Hash-Laenge festgelegt.
    // Sie ist bestimmt durch die Bitzahl des 'int' Datentyps und der Anzahl
    // der Token.
    if (hashLength<1) hashLength = 1;
    hashLength = (hashLength<26 ? hashLength : 25);
```

Ausgangssituation

- JPlag kommt aus den 90ern vom Lehrstuhl Prof. Walter F. Tichy, KIT, Karlsruhe
- Funktionsfähig, aber veraltet, viele Code Smells, Monolith, weder Tests noch CI/CD
- Keine Dokumentation außer Paper, kein Wissen über Code Base vorhanden

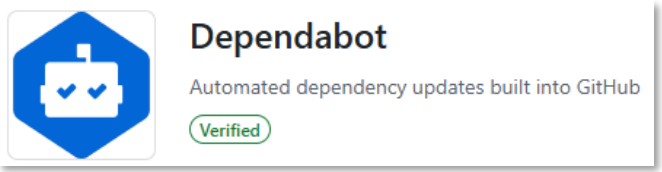
- Feature-Druck: Die TU München möchte JPlag in ihre Lernplattform integrieren
- JPlag hat sich in den letzten 20 Jahren weit verbreitet und wird weltweit eingesetzt



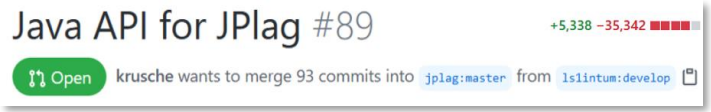
Erste Schritte

Housekeeping

- Erstmal Dependabot hinzufügen ☺
- Minimale Tests
- Code aufräumen, Logging + CLI, toten Code löschen, …
- Open-Source: Issues reorganisieren und labeln, uralte Pull Requests schließen, …




```
public static void usage() {
    System.out.print(Program.name_long
                     + ", Copyright (c) 2004-2017 KIT - IPD Tichy, Guido Malpohl, and others.\n"
                     + "Usage: JPlag [ options ] <root-dir> [-c file1 file2 ...]\n"
                     + " <root-dir>        The root-directory that contains all submissions.\n\n"
                     + "options are:\n"
```



Stabilisierungs-Release

- Ein Jahr aufräumen und Kontrolle gewinnen
- Weniger Features als zuvor, aber: Bessere API, CLI, Bugfixes

v3 **Baseline:** Stabilisierung + “Legacy-MVP”

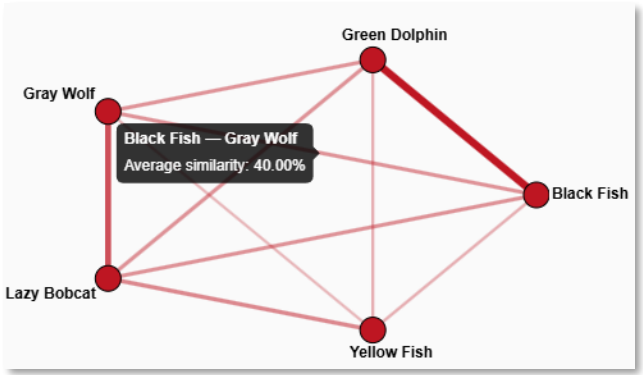
Neue Vision und ein wachsendes Team

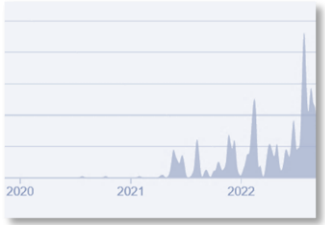
Vision für die Zukunft von JPlag

- Vollständig neue, web-basierte UI
- Mehr Features, z.B. Gruppenplagiarismus
- Umfangreiche End-2-End Tests, Qualität+++
- Aktives gepflegtes GitHub-Projekt mit Community, Docs, Releases, …

Team vergrößern

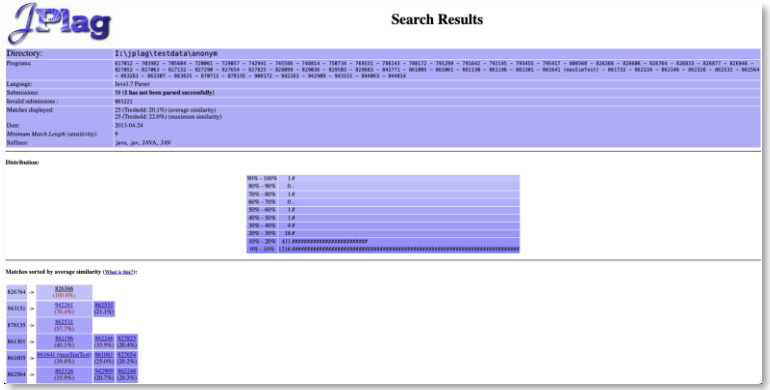
- Mehr Kollegen begeistern ☺
- Studentische Entwickler, Praktika, …

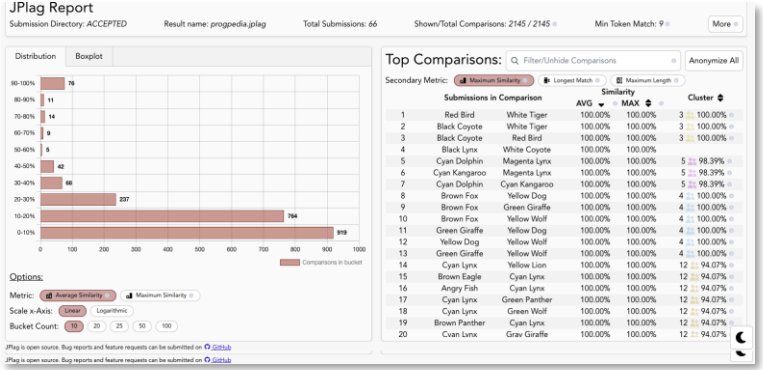


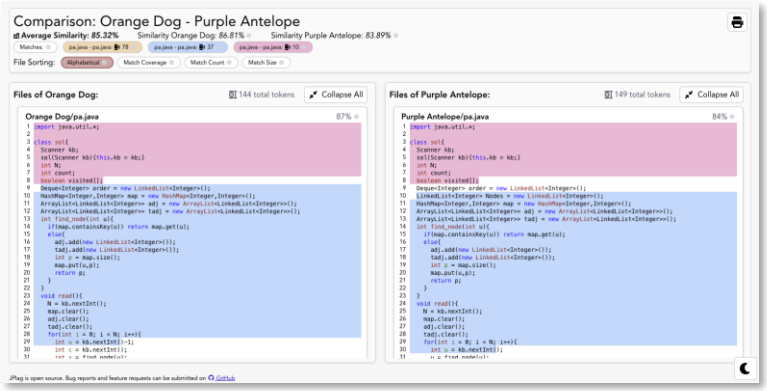


v4 **Rebuild:** Moderne Architektur + Report Viewer

JPlag Report Viewer



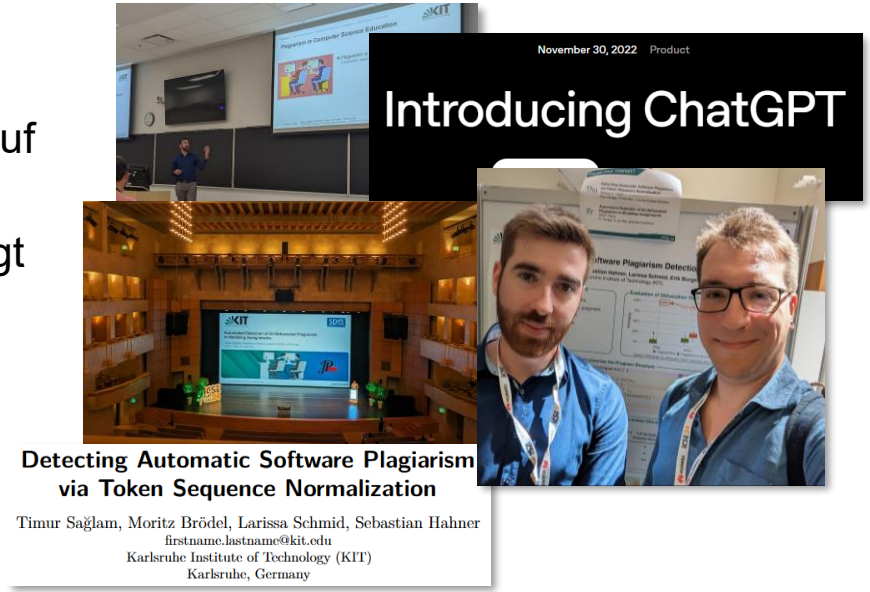


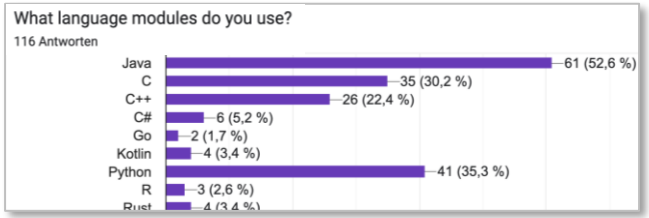


Resilienz gegen Obfuscation Attacks

Neue Angriffsmöglichkeiten

- Oktober 2020: Paper demonstriert Angriff auf JPlag durch Einfügen von Dead Code
- November 2022: ChatGPT wird angekündigt
- ICSE 2024
  - Vorstellung von Abwehrmechanismen
  - Unmittelbar Research-to-Production
- Start der JPlag User Survey





v5 **Expand:** Verbesserte Resilienz + UX

TL;DR: 30 Jahre JPlag

1996 Initiale Entwicklung

2015 Umzug auf GitHub

Ende 2020 **Projekt Übernahme**

2022

2023

2024

2025

v3 **Baseline:** Stabilisierung + “Legacy-MVP”

v4 **Rebuild:** Moderne Architektur + Report Viewer

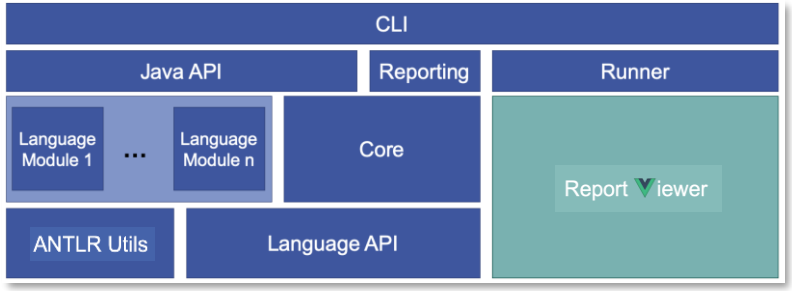
v5 **Expand:** Verbesserte Resilienz + UX

v6 **Polish:** Integrierte Plattform + bessere Analyse

JPlag Heute

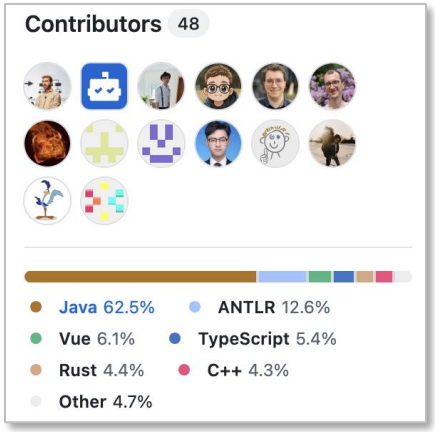
Modernisierte Architektur

- Web-UI in Java-Anwendung integriert
- API für Drittanbieter-Integration
- Einheitliches Sprachmodul-System
- Hohe Performance durch Parallelisierung
- Always-Up-To-Date Web-Demo



Best-Practice Software Engineering

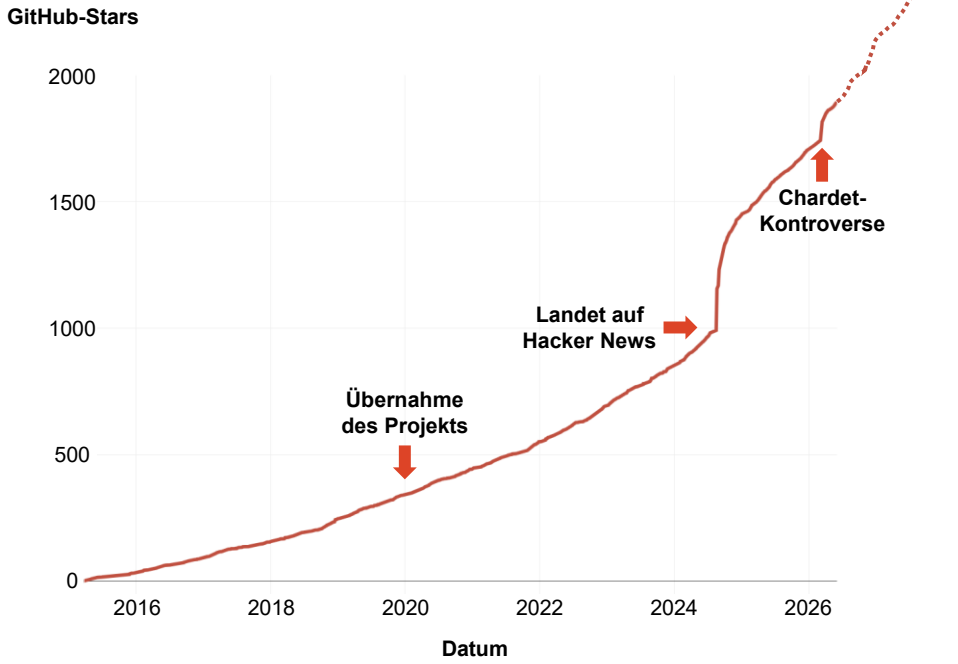
- Umfassende Tests: Unit/Integration/E2E/Language (~84% Coverage)
- Moderne CI/CD-Pipeline (build/lint/scan/test/doc/deploy/release)
- Documentation-as-Code
- Schnelle und verlässliche Release-Prozesse
- Aktive, internationale Open-Source-Community



JPlag in Zahlen

Stetiges Wachstum der Community:

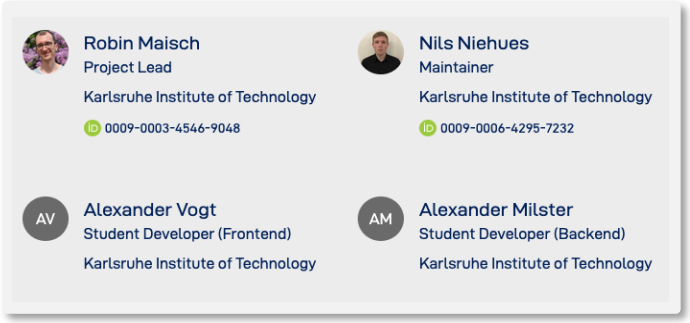
	Metrik	2020	Heute
★	Stars	~300	~1.900
⊸	Commits	~100	~6.000
⎇	Pull Requests	19	2.560
ⓘ	Issues	81	380 (54 offen)
👥	Contributors	12	48



JPlag Morgen

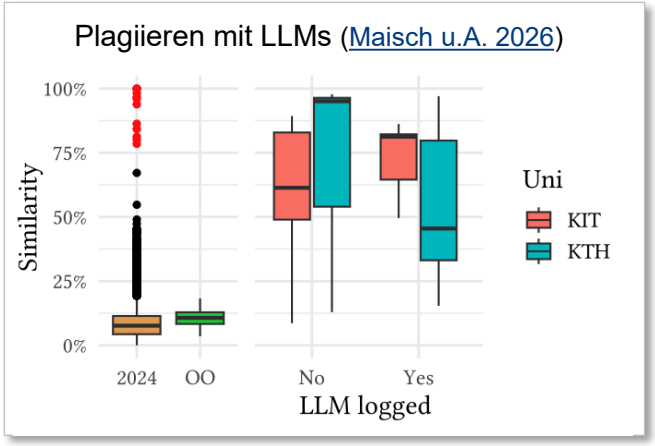
Projekt Lead

- Ab Ende 2025 an nächste Generation übergeben
- Im Hintergrund als Maintainer gelegentlich aktiv



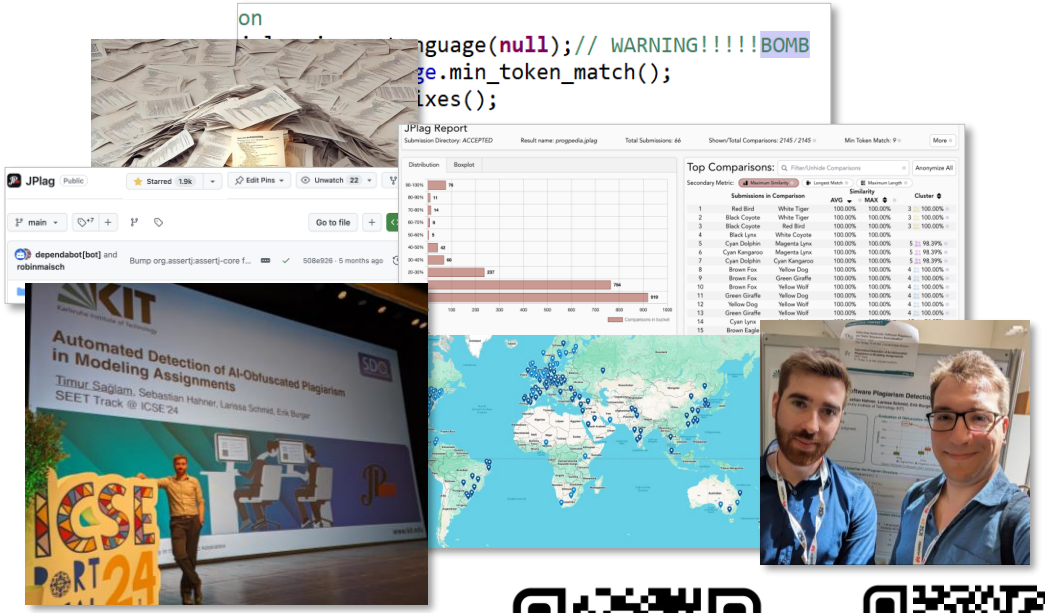
JPlag entwickelt sich weiter

- Engineering: Erkennungsqualität und Explainability
- Forschung: Plagiarismus im Zeitalter von KI
  - KI-basierte Obfuscation verstehen und verhindern
  - KI-generierter Code: Strukturell ähnlicher als gedacht



Zusammenfassung

- **1996** Initiale Entwicklung
- **2015** Umzug auf GitHub
- **Ende 2020** **Projekt Übernahme**
- **2022** v3 **Baseline**
- **2023** v4 **Rebuild**
- **2024** v5 **Expand**
- **2025** v6 **Polish**
- **Ende 2025** **Abgabe Project-Lead**



Mehr zu JPlag…

Vortrag bewerten…

Weiterführende Informationen

Relevante Links

- JPlag auf GitHub: https://www.github.com/jplag/JPlag
- JPlag Online Demo: https://demo.jplag.de/
- Helmholtz RSD: https://helmholtz.software/software/jplag

Ausgewählte Veröffentlichungen

[1] R. Maisch, et al., „Same Same But Different: Preventing Refactoring Attacks on Software Plagiarism Detection", *ICSE*, IEEE/ACM, 2026.

[2] R. Maisch, "Too Hot To Handle: The Effects of Temperature on AI-Generated Code Used to Cheat in Programming Assignments", *ICSE-C*, IEEE/ACM, 2026.

[3] T. Sağlam, et al., "Mitigating Obfuscation Attacks on Software Plagiarism Detectors via Subsequence Merging", CSEE&T, IEEE/ACM, 2025.

[4] T. Sağlam, et al., "Detecting Automatic Software Plagiarism via Token Sequence Normalization", *ICSE*, IEEE/ACM, 2024.

[5] T. Sağlam, et al., "Automated Detection of AI-Obfuscated Plagiarism in Modeling Assignments", *ICSE-SEET*, IEEE/ACM, 2024.

[6] T. Sağlam, et al., "Obfuscation-Resilient Software Plagiarism Detection with JPlag", *ICSE-C*, IEEE/ACM, 2024.

[7] T. Sağlam, et al., "How Students Plagiarize Modeling Assignments", *MODELS-C*, ACM/IEEE, 2023.