

Checkliste zum sicheren Einsatz KI-gestützter Coding-Assistenten

Dr. Hendrik Schöttle / Tim Schmetzer

Osborne Clarke

Herbstakademie 2026

Vibe Coding Tools

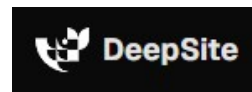
Code Generation



Smart IDEs



Frontend + Deployment



Funktionsweise KI-gestützter Code-Generatoren

KI gestützte Code Generatoren basieren in der Regel auf vortrainierten LLMs, die neben natürlicher Sprache auch Quellcode ausgeben können.

Gezieltes
Prompting
durch den
Nutzer

Auto-
Vervollständigung in
IDE; basierend auf
dem vorhandenen
Quellcode

erstelle mir einen python code, welche alle .doc dateien in einem ordner auflistet

Erstellt liste_doc_dateien.py +23-0 >

Kurz getestet, ob es fehlerfrei läuft:

Tested the script on current folder >

Läuft fehlerfrei (hier gibt's im Ordner keine .doc-Dateien, daher die entsprechende Meldung).

liste_doc_dateien.py:

```
import sys
from pathlib import Path

def liste_doc_dateien(ordner: str, rekursiv: bool = False) -> list[Path]:
    pfad = Path(ordner)
    muster = "*.doc"
    dateien = pfad.rglob(muster) if rekursiv else pfad.glob(muster)
    return sorted(dateien)

if __name__ == "__main__":
    ordner = sys.argv[1] if len(sys.argv) > 1 else "."
    rekursiv = "-r" in sys.argv

    dateien = liste_doc_dateien(ordner, rekursiv)

    if not dateien:
        print(f"Keine .doc-Dateien in '{ordner}' gefunden.")
    else:
        print(f"{len(dateien)} .doc-Datei(en) gefunden:")
        for datei in dateien:
            print(datei)
```

Regulatorische Anforderungen: KI-VO

Gängige LLM-basierte Coding-Assistenten unterfallen i.d.R. dem Begriff eines KI-Systems nach Art. 3 Nr. 1 KI-VO.



Insbes. Relevant: Transparenzpflichten (Art. 50 Abs. 2 KI-VO)

- **Quellcode** lässt sich zwar nach dem Wortlaut der KI-VO unter den Begriff der „synthetischen Textinhalte“ subsumieren
- **Aber** nach Auslegung der EU-Kommission soll Quellcode als solcher nicht unter die Transparenzpflicht des Art. 50 Abs. 2 fallen, da er nicht zur Wahrnehmung, Interpretation, Überprüfung oder als Handlungsgrundlage durch / für natürliche Personen bestimmt ist.

(Guidelines on the implementation of the transparency obligations for certain AI systems under Article 50 of Regulation (EU) 2024/1689 (the 'AI Act') Rn. 67 f.)

Checkliste: KI-VO

Einordnung der eingesetzten KI-Coding-Assistenten als KI-Systeme i.S.d. Art. 3 Nr. 1 KI-VO; Bestimmung der anwendbaren Risikoklasse, Rollen und Pflichten

Sicherstellung ausreichender KI-Kompetenz der betrauten Entwickler und Entscheidungsträger, gem. Art. 4 KI-VO.

Prüfung der Nutzungsbedingungen der eingesetzten KI-Coding-Assistenten, insbesondere:

- Verwendung eingegebenen Codes als Trainingsdaten,
- Vertraulichkeit proprietären Quellcodes
- etwaige Nutzungsbeschränkungen für generierte Outputs.

Regulatorische Anforderungen: CRA

Der CRA verpflichtet den Hersteller von Produkten mit digitalen Elementen eine „Software Bill of Materials“ (SBOM) zu erstellen und zu pflegen.

Dokumentiert sind sämtliche Softwarekomponenten und Abhängigkeiten, Art 31 Abs. 1 i.V.m. Anhang I Teil II Abs. 2 lit. a CRA.

Problem

KI-Coding Assistenten binden autonom Drittbibliotheken ein

Risiko

- Es fehlen Komponenten in der SBOM
 - Verlust des Überblicks über die Softwarezusammensetzung
 - **Lückenhaftes Management der Code Basis** kann gegen Art. 6 und 13 Abs. 1 i.V.m. Anhang I Teil 1 Abs.2 lit. a CRA verstoßen
- **Verstoß gegen**

Checkliste: CRA

- Anpassung bestehender SBOM-Prozesse zur **lückenlosen Erfassung** auch automatisiert durch KI-Coding-Assistenten eingebundener Softwarekomponenten und Abhängigkeiten.
- **Erweiterung des Vulnerability-Managements** auf KI-generierte Code-Bestandteile; Sicherstellung der Freiheit von bekannten Schwachstellen und Erfüllung der Aktualisierungspflichten nach dem CRA.
- **Interne Dokumentation von Art, Umfang und Kontext** des KI-Einsatzes sowie der getroffenen Risikominimierungsmaßnahmen

Urheberrechtliche Risiken

Vervielfältigt der KI-gestützte Coding-Assistent den urheberrechtlich geschützten Code i.S.d. §16 UrhG?



Falls ja, wann wäre eine solche Vervielfältigung dem Anwender des KI-gestützten Coding-Assistenten zurechenbar? Anbieter oder Anwender?

Urheberrechtliche Risiken - Vervielfältigung

Vervielfältigung ist **jede körperliche Festlegung des Werkes**, die geeignet ist, das Werk dem menschlichen Sinnen mittelbar oder unmittelbar wahrnehmbar zu machen, unabhängig von der zeitlichen Dauer der Festlegung und der verwendeten Technik.



- Identische Reproduktion von Code-Snippets: Vervielfältigung (+)
- Ähnelt der generierte Code nur einem bereits bestehenden Werk ist die Rspr. des EuGH heranzuziehen, ob die Wiedererkennbarkeit gegeben ist, Vervielfältigung (+)
- **(P) Rein zufällige, unabhängige Doppelschöpfung durch das KI-System:** Überhaupt noch möglich nach Urteil des LG München GEMA gegen OpenAI (42 O 14139/24)?

Urheberrechtliche Risiken - Zurechnung

Nach Rspr des BGH:

- Täter ist, wer den zur Verletzung führenden Kausalverlauf beherrscht
- Teilnehmer ist, wer einem mit Tatherrschaft handelnden Dritten Hilfe leistet oder Tatentschluss hervorruft
- Fehlen diese Merkmale, dann allenfalls Verantwortlichkeit als Störer



- **Erste Vervielfältigung** in IDE/Nutzeroberfläche des KI-Assistenten: Nutzer führt einen einfachen Prompt aus
 - Kausalverlauf wird durch Nutzer nicht beherrscht
 - Aktiver Eingriff an einer Vervielfältigungshandlung fehlt
- Tatherrschaft wohl zu bejahen, wenn Nutzer den KI-Coding Assistenten zielgerichtet für die Reproduktion eines urheberrechtlich geschützten Codes einsetzt, bspw. Eingabe des geschützten Werkes als Vorlage

Rechtsfolgen und Risikoquantifizierung

Welche Rechtsfolgen können drohen?

- Verschuldensunabhängige Unterlassungs- und Beseitigungsansprüche, § 97 Abs. 1 UrhG
- Verschuldensabhängige Schadensersatzansprüche, § 97 Abs. 2 UrhG
- Straf- und ordnungsrechtliche Konsequenzen, § 106 UrhG; §§ 130, 30 OWiG

Checkliste: Urheberrecht

- Aktivierung sämtlicher vom Anbieter der KI-Coding-Assistenten bereitgestellter **Schutzmechanismen**, bspw. Duplicate-Detection-Funktionen zur Erkennung von Übereinstimmungen mit öffentlich verfügbarem Quellcode und Unterdrückung entsprechender Ausgaben.
- Implementierung **ergänzender Prüfmechanismen** (Software-Composition-Analysis-Tools, Code-Reviews)
- **Erlass interner Policies**, die die zielgerichtete Reproduktion geschützter Werke mittels KI-Coding-Assistenten untersagen.
- Prüfung und Dokumentation der von KI-Anbietern angebotenen **Freistellungszusagen** und Einhaltung der jeweiligen Voraussetzungen.
- Differenzierung der rechtlichen Prüfindensität nach Vertriebsszenario; **strenge Prüf- und Freigabeprozesse** bei nicht OTA-updatefähigen oder regulierten Produkten

Open-Source Compliance: KI-gestützte Replikation von Open-Source Quellcode

KI-Coding Assistent reproduziert einen Quellcode, der unter der Open-Source-Lizenz veröffentlicht wurde:



Interne Nutzung von Code-Snippets ist in der Regel urheberrechtlich unproblematisch

Soll KI-Code verbreitet oder anderweitig an Dritte zugänglich gemacht werden:

- Erkennung und Einhaltung der geltenden (FOSS-) Lizenzen erforderlich
- (P) FOSS-Lizenzen werden von KI nicht oder falsch übernommen

Open Source Compliance: KI-gesteuerte Einbindung von Dritt-Bibliotheken in eigenen Code

Agentische Coding-Assistenten:

- Treffen Auswahl geeigneter (Open-Source-) Programm-bibliotheken vollautomatisiert
- Binden Dritt-Bibliotheken autonom in den eigenen Code ein
- Führen ggf. auch aufwändigere technische Integration ohne menschliches Zutun durch

Library Requirements & Setup Guide

Required Libraries

1. **cJSON** (JSON Parsing)

- **Purpose:** Parse and manipulate JSON data
- **Website:** <https://github.com/DaveGamble/cJSON>
- **License:** MIT

Installation on Windows:

Using vcpkg (recommended):

```
vcpkg install cJSON:x64-windows
```

Or download manually:

```
git clone https://github.com/DaveGamble/cJSON.git
cd cJSON
mkdir build && cd build
cmake -G "Visual Studio 16 2019" -A x64 ..
cmake --build . --config Release
cmake --install .
```

Installation on Linux (Ubuntu/Debian):

```
sudo apt-get install libcjson-dev
```

Installation on macOS:

```
brew install cJSON
```

Checkliste: Open-Source Compliance

- **Erweiterung bestehender FOSS-Compliance-Prozesse** auf KI-generierte Code-Bestandteile
- Verwendung technischer Schutzmaßnahmen der Anbieter wie **Duplikate-Filter**
- **Einsatz von Software-Composition-Analysis-Tools** mit Snippet-Scan-Funktion zur Identifikation reproduzierter Open-Source-Fragmente und Zuordnung der einschlägigen Lizenzen vor Vertrieb der Software an Dritte.
- **Einbindung** automatisiert durch KI-Agenten integrierter Dritt-Bibliotheken in die etablierten **Prüf- und Freigabemechanismen** der Open-Source-Compliance, auch ohne bewusste Auswahlentscheidung des Entwicklers

Fazit und Ausblick

- Ein rechtssicherer Einsatz von KI-gestützten Coding-Assistenten erfordert ein **strukturiertes Risikomanagement**.
- Dieses Risikomanagement ist eng mit den Vorgaben der KI-VO, des CRA, urheberrechtlichen Sorgfaltspflichten sowie Open-Source-Compliance-Prozessen verzahnt.
- Neue Entwicklungen – etwa die weitere Rspr. zur Memorisierung in KI-Modellen und technische Weiterentwicklungen – können das unternehmerische Risikomanagement weiter konkretisieren.
- Unternehmen sollten ihre Maßnahmen laufend an die sich fortentwickelnde Rechtslandschaft und Technologie anpassen und überwachen.

Fragen?

