

# Hochschule Darmstadt

– Kurs: KI und nun?–

## Wie kann KI den Prozess der Dokumentation im Software- Design-Prozess unterstützen?

vorgelegt von

**Lukas Schandl**

Matrikelnummer: 770165

Prof. Andrea Krajewski

## ZUSAMMENFASSUNG

---

Heutzutage ist Software aus dem Alltag nicht mehr wegzudenken. Sie erleichtert viele Aufgaben und kann Effizienzsteigerungen bewirken. Diese Software muss allerdings auch von Entwicklern erstellt werden. Dabei ist es im Software-Design-Prozess wichtig, Dokumentationen zu erstellen, um qualitativ hochwertige Programme zu entwickeln. Die Erstellung ist zwar ein wichtiger Bestandteil, wird jedoch von vielen Entwicklern als lästig angesehen und weist weitere Defizite auf.

Das Aufkommen von KI ist ein spannender Aspekt, um den lästigen Dokumentationsprozess voranzubringen. Aus diesem Grund wird in diesem Paper die Forschungsfrage: **Welche Schritte in der Dokumentation von Software können durch KI unterstützt werden?** bearbeitet. Um dies zu erreichen, werden verschiedene Forschungsgegenstände mithilfe einer Literaturrecherche beantwortet und am Ende diskutiert.

Die Ergebnisse zeigen dabei, dass KI großes Potenzial hat, den alten Dokumentationsprozess zu verbessern. Hierbei ist zu beachten, dass die momentanen Lösungen noch nicht komplett einsatzbereit sind und weitere Forschung erfordern.

## EINLEITUNG

---

In der heutigen Zeit ist die Nutzung von Software im Alltag nicht mehr wegzudenken. Sei es die Nutzung von Grafikprogrammen, Computerspielen, Planern und verschiedensten anderen Programmen. Software hilft uns im Grunde, verschiedenste Aufgaben zu erleichtern und diese effizienter erledigen zu können. Wie wichtig Software ist, kann durch eine Statistik, welche von Bitkom im Januar 2023 durchgeführt wurde, untermauert werden. Diese zeigt den Umsatz von Software in Deutschland im Zeitraum von 2007 bis 2023. Hier ist zu erkennen, dass der Umsatz in der Branche jedes Jahr gestiegen ist. Wenn man hier das Jahr 2022 mit 2023 vergleicht, ist zu erkennen, dass es hier einen Anstieg um ca. 3 Milliarden Euro gegeben hat, was für die Relevanz des Marktes spricht [2].

Diese Software kommt natürlich nicht aus dem Nichts. Diese muss erst, in einem meist langen Prozess, durch Softwareentwickler erstellt werden. Dieser Prozess besteht meist aus folgenden Phasen [1]:

- Planung
- Definition von Anforderungen
- Design und Prototyperstellung
- Software-Entwicklung
- Testen
- Bereitstellung

In diesem Software-Design-Prozess ist die Erstellung von Dokumentationen ein weiterer wichtiger Bestandteil [1]. Dieser ermöglicht es beispielsweise Entwicklern, die an einem Projekt neu anfangen, leichter einsteigen zu können. Des Weiteren wird durch eine gute Dokumentation des Software-Designs die Erweiterung durch neue Features erleichtert.

Obwohl es einige Vorteile bietet, sich gut um die Dokumentation zu kümmern und diese nicht zu vernachlässigen, ist es für die meisten Entwickler eine Aufgabe, die den Spaß raubt. Dadurch wird diese oft vernachlässigt und wird als nervige Aufgabe gesehen. Dies wird vor allem klar, wenn das Paper mit dem Titel „How software engineers use documentation: the state of the practice“ betrachtet wird. Hier wurden Entwickler bezüglich der Dokumentation von Software befragt und deren generelle Haltungen zusammengefasst. Diese wurden in „The Good“, „The Bad“ und „The Ugly“ eingeteilt [11]. Unter „The Good“ wurde aufgezählt, dass Dokumentation der Architektur und anderen abstrakten Informationen meist valide ist und dass Wartung dadurch unterstützt wird. Des Weiteren wird angemerkt, dass

Code-Kommentare auch zur Wartbarkeit der Software beitragen. Auch wenn es diese positiven Punkte gibt, überwiegen die negativen. Diese werden folgendermaßen aufgelistet:

- Dokumentation aller Typen ist regelmäßig veraltet.
- Systeme haben meist zu viel Dokumentation.
- Dokumentation ist meist schlecht geschrieben.
- Nützliche Informationen zu finden, kann so schwer sein, dass Leute es nicht versuchen, diese überhaupt zu finden.
- Die Erstellung vieler Dokumentationen ist so zeitaufwändig, dass die Kosten den Nutzen übersteigen können.
- Ein Großteil der Dokumentation ist nicht glaubwürdig.

Bei dieser Problemstellung besteht die Möglichkeit, durch KI diesen Prozess zu unterstützen. Diese ermöglicht es beispielsweise, automatisch aus einer bereits vorhandenen Code-Base Dokumentation zu erstellen. Hier hört es aber nicht auf. Dabei ist es interessant zu sehen, dass in einer Umfrage von StackOverflow 83 Prozent der Entwickler angegeben haben, KI-Tools zu nutzen. Hier ist in der Abbildung 1.1 zu sehen, dass die Nutzung von KI zur Dokumentation von ungefähr 34 Prozent der Entwickler bereits genutzt wird und 50 Prozent interessiert sind. Damit ist es auf dem dritten Platz direkt hinter dem Coden selbst und dem Debuggen und Hilfe bekommen. Dies zeigt nochmals, wie wichtig es ist, sich mit diesem Thema zu beschäftigen.

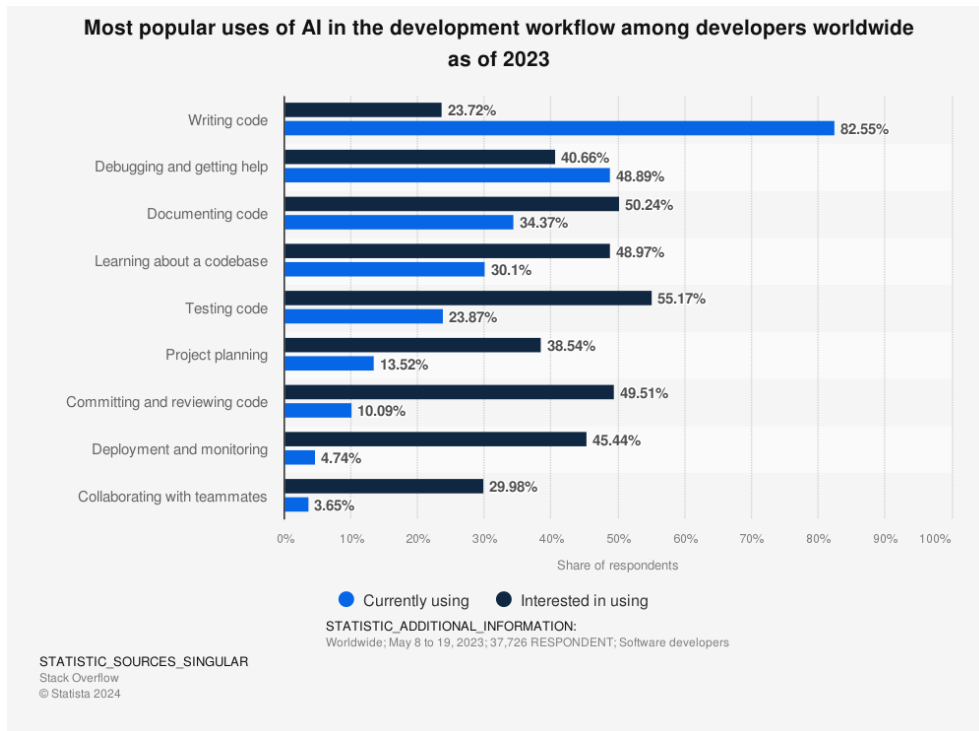


Abbildung 1.1: StackOverflow Umfrage von Statista  
[13]

Folgend wurde aus der Problemstellung und den Erkenntnissen, wo KI möglicherweise Unterstützung bieten kann, eine Forschungsfrage entworfen. Die Forschungsfrage lautet: **Welche Schritte in der Dokumentation von Software können durch KI unterstützt werden?**. Um diese zu beantworten, wurden folgende Forschungsgegenstände definiert:

- Inwiefern kann die automatische Generierung von Dokumentationen die Arbeit von Entwicklern erleichtern?
- Inwiefern kann KI dabei helfen, interaktive Dokumentationen zu erstellen?
- Welche Bedenken gibt es bei der Nutzung von KI im Bereich der Dokumentation von Software?

## 1.1 GRUNDLAGEN

In folgendem Unterabschnitt werden die grundlegenden Begriffe erklärt, die zum Verständnis des Papers wichtig sind.

### 1.1.1 *Software-Design-Prozess*

Dieser Prozess beschreibt im Grunde den gesamten Lebenszyklus eines Softwareprodukts. Dabei werden Schritte wie Planung, Implementierung und

Testen miteinbezogen. Der in diesem Paper untersuchte Prozess der Dokumentation gehört zu einem Unterpunkt, der meist phasenübergreifend durchgeführt werden muss.

### 1.1.2 *Large Language Model*

LLMs sind eine spezifische Art von generativer KI, die auf einer riesigen Datenmenge trainiert werden, um menschenähnlichen Text zu generieren. Diese können zusätzlich als eine Art von Foundation Model eingeordnet werden. Das bedeutet, dass diese, wie bereits erwähnt, auf einer großen Datenmenge trainiert werden, um dabei nicht auf domänenspezifische Use Cases beschränkt zu sein, was vor allem die Kosten und die benötigte Infrastruktur verringert.

Nutzern ist es dabei möglich, über Interfaces wie ChatGPT, Watson.ai oder Metas Llama darauf zuzugreifen [9].

### 1.1.3 *Natural Language Processing*

Natural Language Processing oder kurz NLP ist eine Unterkategorie von KI, die es Computern ermöglicht, menschliche Sprache zu verstehen und damit zu kommunizieren. Im Grunde hat es dazu beigetragen, dass die generative KI so populär geworden ist, da es überhaupt erst ermöglichte, Anfragen des Nutzers in menschlicher Sprache zu verstehen.

Es findet beispielsweise Anwendung in folgenden Bereichen:

- Suchmaschinen
- Chatbots für Kundenservices
- Sprachgesteuerte GPS-Systeme
- Digitale Assistenten auf Smartphones

[8]

Es ist auch wichtig, den Unterschied zu den oben beschriebenen LLMs zu kennen. Im Grunde kann gesagt werden, dass LLMs hauptsächlich zur Generierung menschenähnlichen Textes genutzt werden. Hierbei ist das Verständnis für Nuancen der Sprache aber eingeschränkt, die bei NLPs durch die Konzentration auf die Sprachanalyse vorhanden ist. LLMs können daher als eine Unterkategorie zu NLPs gesehen werden [14].

### 1.1.4 *Generative KI*

Generative KI ist eine künstliche Intelligenz, die es ermöglicht, Inhalte wie Text, Bilder, Videos, Audio oder Softwarecode aufgrund von Eingaben durch einen Nutzer zu generieren [7]. NLPs wie LLMs sind also Unterkategorien dieser, die sich auf das Generieren von Text spezialisieren.

## EVALUATION

---

In diesem Kapitel wird kurz dargestellt, wo die in der Arbeit referenzierten Quellen gesucht wurden. Danach werden die gefundenen Quellen zu den einzelnen Forschungsgegenständen dargestellt, um einen Überblick über diese zu erhalten.

### 2.1 SUCHE NACH LITERATUR

In dieser Sektion wird kurz dargestellt, wie die Literaturrecherche durchgeführt wurde. Um relevante Literatur zu finden, wurden IEEE, ACM und Google Scholar verwendet. Allerdings gibt es zu manchen Themen keine Arbeiten, die sich darauf beziehen. Um dennoch eine Aussage treffen zu können, wurden Artikel von Entwicklern für Entwickler als Quelle herangezogen.

### 2.2 INWIEFERN KANN DIE AUTOMATISCHE GENERIERUNG VON DOKUMENTATIONEN DIE ARBEIT VON ENTWICKLERN ERLEICHTERN

Die Arbeit mit dem Titel „Automated API Docs Generator using Generative AI“ befasst sich mit der Verbesserung der Erstellung von APIs (Application Programming Interfaces).

Diese sogenannten APIs sind grundsätzlich bestimmte Regeln und Protokolle, die es Anwendungen ermöglichen, untereinander zum Beispiel Daten und Funktionalitäten auszutauschen, ohne diese selbst entwickeln zu müssen. Dadurch, dass viele Dinge nicht selbst implementiert werden müssen, wird der Entwicklungsprozess positiv unterstützt, indem dieser simplifiziert und verschnellert wird [6].

Aber auch die Aufgabe, die API-Dokumentation up-to-date zu halten, stellt eine Aufgabe dar, die Entwickler von der eigentlichen Programmieraufgabe abhält. Die Arbeit beschäftigt sich nun mit der Aufgabe, den veralteten Prozess der Erstellung solcher Dokumentationen mit der Generativen KI zu verbessern. Dieser automatische Generator basiert auf Natural Language Processing und wird durch ein Large Language Model angewandt. Die Trainingsdaten stammen aus Dokumentationen von großen Unternehmen, um die KI auf Industrie-Standards zu trainieren. Wie sieht nun die Nutzung des Generators aus Nutzersicht aus? Hier wird dem Benutzer eine grafische Oberfläche geboten, in welcher er Details zu seiner API angeben kann. Im Anschluss gibt die KI eine JSON-Datei aus, die dann mithilfe von CSS, HTML und Javascript eine interaktive Dokumentation erstellt.

Die Arbeit zeigt, dass es möglich ist, den Prozess der API-Dokumentation effizienter zu gestalten und dabei Dokumentation auf einem Industrie-Standard

zu entwickeln. In einem Testfall des Papers brauchte das Erstellen der Dokumentation nur 36 Sekunden [3], was die Schnelligkeit nochmals aufzeigt.

Das Paper mit dem Titel „Automatic Code Documentation Generation Using GPT-3“ beschäftigt sich hingegen mit dem Kommentieren von Code selbst und nicht mit der Generierung einer API-Dokumentation. Es geht also nicht um die Erstellung einer Schnittstelle, die andere Programme nutzen können, sondern um das Kommentieren des Codes selbst, um diesen für Entwickler potenziell besser lesbar zu machen. Hierzu wird in der Arbeit Codex genutzt, welches auf GPT-3 basiert. Die Daten selbst stammen aus CodeSearchNet, einem weit verbreiteten Datenset aus verschiedensten Programmiersprachen. Dieses wurde anschließend genutzt, um Code zu kommentieren und daraus qualitative und performancebezogene Schlüsse zu ziehen. In dieser Arbeit ist aber vor allem der qualitative Teil interessant zu betrachten. Hier wurden einmal der Flesch-Kincaid-Score und die TF-IDF-Methode genutzt. Ersterer beschäftigt sich mit der Quantität und der Lesbarkeit der Dokumentation. Hier ist zu sehen, dass die generierte Dokumentation mit dem Wert von 5,97 und einer durchschnittlichen Länge von 8 Wörtern sehr nahe an die tatsächliche Dokumentation mit einem Wert von 6,77 und einer durchschnittlichen Länge von 12 Wörtern herankommt. Bei der TF-IDF-Methode geht es um die qualitative Analyse des Inhalts. Auch hier ist der Wert der generierten Dokumentation nahe an der echten; diese erreicht nämlich einen Wert von 1,94 und die echte den Wert 2,28. Interessant zu beobachten ist auch, dass die KI in manchen Fällen extra Informationen bietet, die nicht vorhanden waren. Dies kann in der Abbildung 2.1 gesehen werden. Hier wird statt „decimal“ „base-10 number“ geschrieben, was für Entwickler leichter zu verstehen ist [10].

```

PHP .....
public static function base256ToDec($number)
{
    Validate::canCastAsString($number, 'number');
    $result = 0;
    $base = 1;
    for ($i = strlen($number) - 1; $i >= 0; $i--) {
        $result = bcadd($result, bcmul(ord($number[$i]), $base));
        $base = bcmul($base, 256);
    }
    return $result;
}
Actual: Convert base 256 number to decimal number.
Codex: Converts a base 256 number to a base 10 number.

```

Abbildung 2.1: Vergleich der Kommentare der echten und generierten Dokumentation

[10]

### 2.3 INWIEFERN KANN KI DABEI HELFEN, INTERAKTIVE DOKUMENTATIONEN ZU ERSTELLEN.

Die Arbeit mit dem Titel „Automated API Docs Generator using Generative AI“, welche bereits in der Sektion 2.2 erläutert wurde, zeigt, dass KI Output



erzeugen kann, welcher es ermöglicht, daraus leicht interaktive Dokumentationen zu erstellen. Dies geschieht in diesem Fall durch das Erzeugen einer JSON-Datei. Wobei der Schritt zum Erstellen der Datei in dieser Arbeit nicht automatisiert ist, aber durch ein Tool leicht durchzuführen ist. Anschließend kann durch den Output eine interaktive Seite erstellt werden [3].

Ähnliches wird auch in der Arbeit mit dem Titel „Improving open data web API documentation through interactivity and natural language generation“ aufgezeigt. In dieser Arbeit wird versucht, den Prozess der Erstellung von API-Dokumentationen mit KI zu erleichtern. Hier wird aber nicht, wie im ersten Paper gezeigt, eine Benutzeroberfläche genutzt, um Daten der API zu erlangen. Stattdessen wird die bereits vorhandene Dokumentation der API genutzt, um an die notwendigen Informationen zu kommen. Im Anschluss werden die Hauptkonzepte der API durch natürliche Sprache beschrieben und ein interaktives Webinterface dazu erstellt.

Des Weiteren wurde die generierte Dokumentation verschiedenen Tests mit Nutzern unterzogen. Interessant zu sehen ist, dass Nutzer mit der durch KI erstellten Version zufriedener waren und auch in der Anwendung weniger Fehler bei der Nutzung der API machten [5].

## 2.4 WELCHE BEDENKEN GIBT ES BEI DER NUTZUNG VON KI IM BEREICH DER DOKUMENTATION VON SOFTWARE?

In diesem Bereich war es schwer, wissenschaftliche Arbeiten spezifisch für die Dokumentation zu finden. Es wird sich daher auf Fachartikel bezogen. KI kann zwar dabei helfen, Entwickler zu entlasten und den Prozess zu erleichtern. Es gibt trotzdem, wie auch bei der manuellen Erstellung solcher Dokumente, die Bedenken, ob dem Erstellten zu vertrauen ist. In diesem Bereich wird es also wahrscheinlich weiterhin nötig sein, eine Qualitätssicherung durchzuführen, welche durch reale Personen ausgeführt werden muss.

Es treffen trotzdem auch generelle Bedenken zur generativen KI zu. Dazu zählen folgende Punkte:

- Inkorrekte Ausgaben
- Bias und Fairness
- Copyright-Verstöße
- Bedenken bezüglich der Umwelt

[4]

## FAZIT

---

In folgendem Kapitel werden die Ergebnisse des Papers zusammengefasst, um einen Überblick über diese zu erhalten. Des Weiteren werden Limitationen des Forschungsbereichs aufgezeigt und zuletzt wird ein Ausblick über mögliche weiterführende Forschung gegeben. Die einzelnen Unterpunkte werden bezüglich der einzelnen Forschungsfragen bearbeitet.

### 3.1 ERGEBNISSE

Folgend werden die Ergebnisse der einzelnen Forschungsfragen diskutiert und zusammengefasst. Fragenübergreifend kann gesagt werden, dass die Dokumentation von Software eine wichtige Aufgabe ist, die viele Vorteile bietet, jedoch meist durch die Art und Weise des Dokumentationsprozesses vernachlässigt wird. Dadurch entsteht ein großes Potenzial für KI, diesen Teil des Software-Design-Prozesses zu verbessern und Entwicklern unter die Arme zu greifen.

#### **Inwiefern kann die automatische Generierung von Dokumentationen die Arbeit von Entwicklern erleichtern?**

Durch die Recherche wird klar, dass es einige positive Entwicklungen in diesem Bereich gibt. Es ist bereits möglich, Dokumentationen im Code als auch für Schnittstellen erstellen zu lassen. Diese erstellten Artefakte sind auch bereits sehr nahe an von Menschen erstellten. Es ist anzumerken, dass die jeweiligen Arbeiten in einem weniger komplexen Rahmen hantieren als viele Produktivsysteme in Unternehmen. KI ist sehr gut darin, isoliert Code zu betrachten, ist aber nicht gut in der Lage, ein großes komplexes System zu verstehen, welches oft auch über mehrere Komponenten geht, die nicht selbst im Code vorhanden sind [12]. Betrachtet man beispielsweise ein Unternehmen im Finanzbereich, so haben diese möglicherweise eigene Systeme, die Daten von anderen großen Partnerfirmen über Schnittstellen erhalten. Eine KI ist immer nur so gut wie ihre verfügbaren Informationen und kann dadurch Fehler in solch komplexen Systemen machen, da sie nicht alle nötigen Informationen über Systeme von Dritten hat.

Es lässt sich also sagen, dass weiterführende Forschung in diesem Bereich vielversprechende Ergebnisse bieten kann und den Alltag eines Softwareentwicklers in Zukunft stark verbessern könnte.

### **Inwiefern kann KI dabei helfen, interaktive Dokumentationen zu erstellen?**

Die gefundene Literatur zeigt, dass es mithilfe von KI möglich ist, relativ leicht interaktive Dokumentationen zu erstellen. Es wird des Weiteren gezeigt, dass Nutzer mit der kreierte Dokumentation gut/besser arbeiten können. Es gibt jedoch trotzdem einige Limitationen. Die Information über die API muss irgendwie trotzdem vorhanden sein, sei es durch bereits existierende Dokumentation oder durch Spezifikation des Nutzers. Des Weiteren steht man vor den gleichen Schwierigkeiten, die in allen Bereichen der NLG (Natural Language Generation) auftreten. Dazu zählt beispielsweise das Vorhandensein von Metadaten, um sinnvolle Texte generieren zu können.

Weitere Forschung in der Zukunft könnte sein, mehr Optionen des Sprachstils bei der Generierung von Dokumentation anzubieten, um beispielsweise verschiedenste Sprachlevel abzudecken.

### **Welche Bedenken gibt es bei der Nutzung von KI im Bereich der Dokumentation von Software?**

Im Großen und Ganzen kann gesagt werden, dass KI vor allem im Bereich von komplexen Systemen noch nicht ohne Bedenken Dokumentation erzeugen kann, da diese keine Information über die außenstehenden Systeme hat. Des Weiteren gibt es die gleichen Bedenken wie auch bei menschlicher Dokumentation. Auch bei generierter KI wird es in Zukunft weiterhin eine Person geben, die die Qualität der Dokumentation überprüfen sollte.

Zu guter Letzt ist anzumerken, dass die Bedenken von generativer KI auch im Anwendungsfall von generierter Dokumentation relevant sind.

Weiterführende Forschung sollte dennoch im Kontext der Dokumentation durchgeführt werden, um auch hier wissenschaftliche Aussagen besser treffen zu können. Momentan gibt es dazu nur fachliche Artikel.

## LITERATUR

---

- [1] BITECC. *Was ist der Software-Lebenszyklus (SDLC)?* <https://bitecc.de/was-ist-der-software-lebenszyklus-sdlc/>. Zugriff am 29. Juni 2024. 2023.
- [2] Bitkom. *Umsatz mit Software in Deutschland in den Jahren 2007 bis 2023 (in Milliarden Euro)*. <https://de.statista.com/statistik/daten/studie/189894/umfrage/marktvolumen-im-bereich-software-in-deutschland-seit-2007/>. Zugriff am 29. Juni 2024. 2023.
- [3] Prakhar Dhyani, Shubhang Nautiyal, Aditya Negi, Shikhar Dhyani und Preeti Chaudhary. "Automated API Docs Generator using Generative AI". In: *2024 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*. 2024, S. 1–6. DOI: [10.1109/SCEECS61402.2024.10482119](https://doi.org/10.1109/SCEECS61402.2024.10482119).
- [4] Stefan Feuerriegel, Jochen Hartmann, Christian Janiesch und Patrick Zschech. "Generative AI". In: *Business & Information Systems Engineering* 66.1 (2024), S. 111–126. DOI: [10.1007/s12599-023-00834-7](https://doi.org/10.1007/s12599-023-00834-7). URL: <https://doi.org/10.1007/s12599-023-00834-7>.
- [5] César González-Mora, Cristina Barros, Irene Garrigós, Jose Zubcoff, Elena Lloret und Jose-Norberto Mazón. "Improving open data web API documentation through interactivity and natural language generation". In: *Computer Standards Interfaces* 83 (2023), S. 103657. ISSN: 0920-5489. DOI: <https://doi.org/10.1016/j.csi.2022.103657>. URL: <https://www.sciencedirect.com/science/article/pii/S0920548922000344>.
- [6] IBM. *What Is an API (Application Programming Interface)?* <https://www.ibm.com/topics/api>. Zugriff am 29. Juni 2024. 2024.
- [7] IBM. *What is Generative AI?* <https://www.ibm.com/topics/generative-ai>. Zugriff am 1. Juli 2024. 2024.
- [8] IBM. *What is NLP (Natural Language Processing)?* <https://www.ibm.com/topics/natural-language-processing>. Zugriff am 14. Juli 2024. 2024.
- [9] IBM. *What are Large Language Models (LLMs)?* <https://www.ibm.com/topics/large-language-models>. Zugriff am 14. Juli 2024. o.D.
- [10] Junaed Younus Khan und Gias Uddin. "Automatic Code Documentation Generation Using GPT-3". In: *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*. ASE '22. Rochester, MI, USA: Association for Computing Machinery, 2023. ISBN: 9781450394758. DOI: [10.1145/3551349.3559548](https://doi.org/10.1145/3551349.3559548). URL: <https://doi.org/10.1145/3551349.3559548>.

- [11] Timothy C. Lethbridge, Janice Singer und Andrew Forward. "How software engineers use documentation: the state of the practice". In: *IEEE Software* 20.6 (2003), S. 35–39. DOI: [10.1109/ms.2003.1241364](https://doi.org/10.1109/ms.2003.1241364).
- [12] O. Rosenbaum. *Will ChatGPT and generative AI replace internal code documentation?* <https://leaddev.com/tech/will-chatgpt-and-generative-ai-replace-internal-code-documentation>. Zugriff am 1. Juli 2024. n.d.
- [13] Statista. *Popular uses of AI in development workflow worldwide 2023*. <https://www.statista.com/statistics/1401409/popular-ai-uses-in-development-workflow-globally/>. Zugriff am 29. Juni 2024. 2024.
- [14] S. Vaniukov. *NLP vs LLM: A Comprehensive Guide to Understanding Key Differences*. <https://medium.com/@vaniukov.s/nlp-vs-llm-a-comprehensive-guide-to-understanding-key-differences-0358f6571910>. Zugriff am 1. Juli 2024. 2024.