

# EINSATZ VON LLM TOOLS IM SOFTWARE ENGINEERING PROZESS

BACHELORARBEIT  
ZUR ERLANGUNG DES AKADEMISCHEN GRADES  
BACHELOR OF ENGINEERING (B.ENG.)

Jonas Gundlach

Betreuerin:  
Prof. Dr. rer. nat. Andrea Baumann

Tag der Abgabe: XX.XX.XXXX

eingereicht bei  
Universität der Bundeswehr München  
Fakultät für Elektrotechnik und Technische Informatik



*Universität der Bundeswehr München*

**Institut für**  
**Software Engineering**

Neubiberg, Juli 2024



# Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, insbesondere keine anderen als die angegebenen Informationen.

Der Speicherung meiner Bachelorarbeit zum Zweck der Plagiatsprüfung stimme ich zu. Ich versichere, dass die elektronische Version mit der gedruckten Version inhaltlich übereinstimmt.

Neubiberg, den XX.XX.XXXX

---

Jonas Gundlach



## **Zusammenfassung**

TODO!!!!!!!!!!!!!!

Ein Abstract ist ein Text, der kurz und knapp die wichtigsten Punkte deiner Bachelorarbeit wiedergibt. Er soll die Fragestellung, Methoden, Quellen und Ergebnisse zusammenfassen, damit die Lesenden schnell die Relevanz der Bachelorarbeit für ihr Interessensgebiet beurteilen können.



# Inhaltsverzeichnis

<b>Erklärung</b>	<b>III</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Zielsetzung der Arbeit . . . . .	1
1.3 Struktur der Arbeit . . . . .	1
<b>2 Grundlagen</b>	<b>3</b>
2.1 LLM Tools . . . . .	3
2.1.1 ChatGPT . . . . .	4
2.1.2 Google Gemini . . . . .	6
2.1.3 Le Chat . . . . .	8
2.2 Software Engineering Prozess . . . . .	8
2.2.1 Besprechungsprotokoll . . . . .	10
2.2.2 Projekthandbuch . . . . .	11
2.2.3 Risikoliste . . . . .	12
2.2.4 Anforderungsspezifikation . . . . .	13
<b>3 Anwendung von LLM Tools im Software Engineering</b>	<b>15</b>
3.1 Besprechungsprotokoll . . . . .	15
3.2 Projekthandbuch . . . . .	16
3.3 Risikoliste . . . . .	16
3.4 Anforderungsspezifikation . . . . .	17
3.5 Architekturdokument . . . . .	18
3.6 Testspezifikation . . . . .	18
<b>4 Praxisergebnisse und Vergleich</b>	<b>21</b>
4.1 ?????? . . . . .	21
4.2 ChatGPT . . . . .	21
4.3 Gemini . . . . .	21
4.4 Le Chat . . . . .	21
4.5 Vergleich . . . . .	21
<b>5 Herausforderungen und Limitationen</b>	<b>23</b>
5.1 Technische Herausforderungen . . . . .	23
5.2 Limitationen von LLM Tools im Software Engineering Kontext . . . . .	23

<b>6 Zusammenfassung und Ausblick</b>	<b>25</b>
6.1 Zusammenfassung der Ergebnisse . . . . .	25
6.2 Implikationen für die Praxis . . . . .	25
6.3 Ausblick und zukünftige Entwicklungen . . . . .	25
<b>Abbildungsverzeichnis</b>	<b>V</b>
<b>Tabellenverzeichnis</b>	<b>VII</b>
<b>Quellcodeverzeichnis</b>	<b>IX</b>
<b>Stichwortverzeichnis</b>	<b>XI</b>
<b>Literaturverzeichnis</b>	<b>XIII</b>



# 1 Einleitung

TODO!!!!!!!!!!

## 1.1 Motivation

In einer Welt, die von immer komplexeren und dynamischeren Softwareanwendungen geprägt ist, wird die effektive Nutzung von Technologien zunehmend entscheidend für den Erfolg von Softwareprojekten. In diesem Zusammenhang gewinnen Werkzeuge des Natural Language Processing eine immer größere Bedeutung. Insbesondere die jüngsten Fortschritte im Bereich der Large Language Models (LLM) eröffnen neue Möglichkeiten für die Verbesserung des Software Engineering Prozesses.

Die Motivation hinter dieser Arbeit liegt in der Erkundung des Einsatzes von LLM-Tools im Software Engineering und der Analyse ihrer Auswirkungen auf verschiedene Phasen des Entwicklungszyklus. Die Frage, wie diese fortschrittlichen LLM-Tools genutzt werden können, um den Softwareentwicklungsprozess effizienter, präziser und insgesamt erfolgreicher zu gestalten, steht im Mittelpunkt dieser Untersuchung.

TODO!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

## 1.2 Zielsetzung der Arbeit

Todo!!!!!!!!!!!!!!!!!!!!!!

## 1.3 Struktur der Arbeit

Todo!!!!!!!!!!!!!!!!!!!!!!



## 2 Grundlagen

In diesem Kapitel wird erläutert, was LLM Tools sind, wie sie trainiert werden und wie sie funktionieren. Anschließend werden die einzelnen Tools ChatGPT, Google Gemini und Le Chat vorgestellt und deren Entwicklung beschrieben.

Im zweiten Teil des Kapitels wird der Software Engineering Prozess erklärt und erläutert, was dieser umfasst. Dabei wird als Softwareprozess ausschließlich das Wasserfallmodell betrachtet, da in dieser Arbeit nur mit diesem Modell gearbeitet wird.

### 2.1 LLM Tools

Unter LLM-Tools versteht man Sprachmodelle, die auf einem Large Language Model (LLM) basieren. Ein LLM ist ein Deep-Learning-Algorithmus, der mit sehr großen Datensätzen trainiert wird. Diese Modelle finden häufig Anwendung im Bereich des Natural Language Processing (NLP), wo sie verwendet werden, um Abfragen in natürlicher Sprache zu beantworten oder Ergebnisse zu liefern. LLMs können neue Inhalte verstehen, zusammenfassen, generieren und vorhersagen. Durch das Training sammeln LLMs Milliarden von Parametern, bei denen es sich um Variablen handelt, die im Modell angepasst werden, um neue Inhalte abzuleiten.

LLMs basieren auf einem Transformer-Modell, das Eingaben in Token umwandelt und dann gleichzeitig mathematische Gleichungen ausführt, um Beziehungen zwischen den Token zu ermitteln. Dadurch kann der Computer Muster erkennen, die auch ein Mensch wahrnehmen würde, wenn ihm die gleiche Frage gestellt wird. Zudem verwenden Transformer-Modelle Selbstaufmerksamkeitsmechanismen, die es dem Modell ermöglichen, schneller zu lernen als herkömmliche Modelle. Somit kann das Transformer-Modell verschiedene Teile der Sequenz oder den gesamten Kontext eines Satzes berücksichtigen, um Vorhersagen zu generieren.

Grundsätzlich bestehen LLMs aus vier neuronalen Netzwerkschichten: der wiederkehrenden Ebene, der Einbettungsebene, der Feedforward-Ebene und der Aufmerksamkeitsebene. Diese Schichten arbeiten zusammen, um den Eingabetext zu verarbeiten und Ausgabeinhalte zu generieren.

Die wiederkehrende Ebene dient dazu, die Wörter des Eingabetextes der Reihe nach zu interpretieren und die Beziehungen zwischen den Wörtern in einem Satz zu erfassen. Die Einbettungsebene erfasst die semantische und syntaktische Bedeutung der Eingabe, sodass das Modell den Kontext verstehen kann. Die Feedforward-Schicht besteht aus mehreren vollständig verbundenen Schichten, die die Eingabeeinbettungen transformieren. Dadurch ermöglichen diese Schichten dem Modell, Abstraktionen auf höherer Ebene zu verstehen und somit die Absicht des Benutzers mit der Texteingabe zu erfassen. Die Aufmerksamkeitsebene ermöglicht es dem Sprachmodell, sich auf einzelne Teile des Eingabetextes

zu konzentrieren, die für die Aufgabe relevant sind, und dadurch die genauesten Ausgaben zu generieren. Damit ein großes Sprachmodell Texteingaben empfangen und eine Ausgabevorhersage generieren kann, muss es zunächst allgemein geschult und anschließend feinabgestimmt werden, um spezifische Aufgaben ausführen zu können. Für die Schulung werden riesige Datenmengen im Petabyte-Bereich benötigt. Das Training verläuft mehrstufig und beginnt in der Regel mit einem unbeaufsichtigten Lernansatz, bei dem das Modell mit unstrukturierten und unbeschrifteten Daten trainiert wird, da diese in größeren Mengen verfügbar sind. In dieser Phase leitet das Modell Beziehungen zwischen verschiedenen Wörtern und Konzepten ab.

Anschließend erfolgt die Schulung und Feinabstimmung durch eine Form des selbstüberwachten Lernens. Dabei wird eine Datenkennzeichnung durchgeführt, durch die das Modell verschiedene Konzepte besser und genauer identifizieren kann. Im nächsten Schritt durchläuft das LLM den transformatorischen neuronalen Netzwerkprozess im Rahmen des Deep Learning. Die Transformer-Modellarchitektur ermöglicht es dem LLM, mittels eines Selbstaufmerksamkeitsmechanismus Beziehungen und Verbindungen zwischen Wörtern und Konzepten zu erkennen, indem bestimmten Elementen (Token) Bewertungen zugewiesen werden, um die Beziehungen festzulegen.

Entscheidend über die Leistungsfähigkeit und die Qualität des Sprachmodells sind die Datensätze, welche zum trainieren benutzt werden. Im folgenden wird auf die drei unterschiedliche LLM-Tools eingegangen, welche in dieser Arbeit näher untersucht werden [1].

### 2.1.1 ChatGPT

ChatGPT wurde von OpenAI am 30. November 2022 veröffentlicht und entwickelte sich aufgrund seiner fortschrittlichen generativen KI-Fähigkeiten rasch zu einer internationalen Sensation. Wenn man ChatGPT fragt, ob es einen Infotext zu sich selbst erstellen kann, erhält man die Antwort in Prompt 2.1 Die Beta-Version der Plattform war zunächst kostenlos verfügbar, und OpenAI bot den Nutzern weiterhin eine kostenlose Basisversion an. Im Februar 2023 führte OpenAI ein kostenpflichtiges Abonnement ein, das unbegrenzten Zugriff auf ChatGPT sowie eine erweiterte Palette von Funktionen und Diensten ermöglichte.

Trotz der begeisterten Resonanz löste ChatGPT auch einige Kontroversen aus. Es lieferte teilweise fehlerhafte Antworten, insbesondere bei Aufgaben wie Schlussfolgerungen ziehen, Nuancen analysieren, Meinungen bewerten oder Vorhersagen treffen. Pädagogen weltweit äußerten Bedenken hinsichtlich des Einsatzes von ChatGPT im Bildungsbereich, auch an Universitäten wurde der Einsatz kritisch diskutiert. Trotz dieser Kontroversen schien ChatGPT bereit zu sein, die Erstellung von Inhalten im Internet zu revolutionieren. Webentwickler und Content-Ersteller nutzten ChatGPT zunehmend als Recherche- und Schreibhilfe für Website-Texte.

Im März 2023 brachte OpenAI eine neue Version heraus, bekannt als GPT-4. Ein bedeutender Unterschied zu früheren Versionen bestand darin, dass GPT-4 Eingaben sowohl in Text- als auch in Bildformaten akzeptierte. Dies ermöglichte dem Chatbot, Daten aus Diagrammen, Grafiken und Screenshots zu interpretieren. Zudem machte GPT-4 deutlich weniger Denkfehler und sachliche Fehler, auch "Halluzinationen" genannt, als das zuvor verwendete GPT-3.5-Modell. Am Ende des Jahres hatte sich ChatGPT weiterentwickelt und konnte beispielsweise Bilder analysieren, Sprachansagen verarbeiten

**Prompt: Erstell mir einen Infotext über dich für meine Bachelorarbeit**

ChatGPT ist ein fortschrittlicher KI-gestützter Sprachassistent, der auf der GPT-4-Architektur von OpenAI basiert. Entwickelt im Rahmen der Forschung und Entwicklung im Bereich Künstliche Intelligenz, ist ChatGPT in der Lage, natürliche Sprache zu verstehen und darauf zu antworten. Dies macht ihn zu einem vielseitigen Werkzeug für eine Vielzahl von Anwendungen, von der Beantwortung alltäglicher Fragen bis hin zur Unterstützung bei komplexeren Aufgaben wie dem Schreiben von Texten, der Generierung von Ideen und der Bereitstellung von Informationen.

Die GPT-4-Architektur, auf der ChatGPT basiert, verwendet tiefes maschinelles Lernen, um menschliche Sprache zu verarbeiten und zu erzeugen. Durch die Analyse großer Mengen an Textdaten kann das Modell Muster und Zusammenhänge erkennen, die es ihm ermöglichen, kontextuell relevante und kohärente Antworten zu geben. Diese Fähigkeiten machen ChatGPT zu einem nützlichen Instrument in Bereichen wie Bildung, Forschung, Kundenservice und mehr.

Ein weiterer bemerkenswerter Aspekt von ChatGPT ist seine Fähigkeit, sich an eine Vielzahl von Themen anzupassen und Informationen auf klare und präzise Weise zu vermitteln. Dies wird durch kontinuierliches Training und Verbesserung des Modells erreicht, wobei sowohl technische Fortschritte als auch Nutzerfeedback berücksichtigt werden.

Zusammengefasst ist ChatGPT ein leistungsstarker und vielseitiger KI-Assistent, der durch seine fortschrittliche Technologie und seine breite Anwendbarkeit einen wertvollen Beitrag in vielen Bereichen leistet.

Prompt 2.1: Infotext ChatGPT, Quelle: ChatGPT

und gesprochene Antworten geben.

Im Mai 2024 kündigte OpenAI ein aktualisiertes Modell namens ChatGPT-4o an, das schnellere Ergebnisse sowie verbesserte Text- und Audiofunktionen versprach. Dieses Modell umfasst auch ein neues Konversations-KI-System namens "Sky" [2].

### 2.1.2 Google Gemini

Im Dezember 2023 stellte Google eine generative Künstliche Intelligenz mit dem Namen Google Gemini vor. Sie wurde von Google DeepMind entwickelt und ist laut Aussage von Google das größte, leistungsstärkste und vielseitigste große Sprachmodell (LLM), das bisher von Google entwickelt und realisiert wurde. In der ursprünglichen Ankündigung wurde das KI-Sprachmodell unter dem Namen Gemini vorgestellt. Im Februar 2024 entschied Google, auch den bisher als Bard bekannten Chatbot in Gemini umzubenennen. Damit ist Gemini mittlerweile sowohl der Name des Google Chatbots als auch des zugrundeliegenden Sprachmodells.

Bei Nachfrage an Gemini, ob es einen Infotext über sich erstellen kann, erhält man die Antwort in Prompt 2.2 und Prompt 2.3.

Gemini 1.0 wurde in drei verschiedenen Größen optimiert: Gemini Ultra als das größte und leistungsstärkste Modell für hochkomplexe Aufgaben, Gemini Pro als das beste Modell zur Anwendung in einem breiten Aufgabenspektrum und Gemini Nano als das effizienteste Modell für Aufgaben, die direkt auf dem Gerät verarbeitet werden.

Die Gemini-Modelle sind von Grund auf multimodal konzipiert, was bedeutet, dass sie mit verschiedenen Arten von Informationen umgehen können. Gemini ist in der Lage, Texte, Programmcode, Bilder, Videos, Audioinformationen und deren Kombinationen zu analysieren, zu verstehen, zu kombinieren und zu generieren. Da Gemini auch mit Skizzen und handgeschriebenem Text umgehen kann, eignet es sich für den Einsatz in Fachbereichen wie zum Beispiel Mathematik oder Physik. Im Programmierbereich werden gängige Programmiersprachen wie Python, Java, C++ oder Go unterstützt. Gemini kann Programmcode analysieren, kommentieren und auch generieren. Dadurch steht Gemini in Konkurrenz zu anderen LLMs wie den GPT-Sprachmodellen von Open AI. Gemini soll jedoch die Konkurrenzprodukte in vielen Benchmarks deutlich übertreffen.

Gemini baut auf den ebenfalls von Google entwickelten Sprachmodellen LaMDA und PaLM 2 auf. Wie üblich für LLMs (Large Language Models) basiert auch Gemini auf einer Transformer-Architektur. Google hat die Architektur des Modells zur Skalierung der Trainingsfähigkeiten optimiert und mit einem Multi-Query-Aufmerksamkeitsmechanismus ausgestattet. Die unterstützte Kontextlänge beträgt 32.000 Tokens und in der Version 1.5 bis zu einer Million Tokens. Gemini wurde mit Daten verschiedener Modalitäten trainiert, und als Trainingsplattform verwendete Google Rechner-Cluster und Tensor Processing Units (TPUs) der neuesten Generation v4 und v5 als KI-Beschleuniger.

Google hat begonnen, die KI in zahlreiche eigene Produkte zu integrieren, um die Fähigkeiten der Websuche, des Chrome-Browsers, des Gmail-Services, verschiedener Entwicklungstools und anderer Produkte und Services zu verbessern und zu erweitern. Mitte Februar 2024 stellte Google Gemini 1.5 und 1.5 Pro vor, welche bis zu eine Million Tokens verarbeiten können. Diese neuen Versionen sollen die Benchmark-Ergebnisse der Vorgängerversion deutlich übertreffen [3].

**Prompt: Erstell mir einen Infotext über dich für meine Bachelorarbeit****Was ist Gemini?**

Gemini ist ein Large Language Model (LLM) von Google AI, das mit einem riesigen Datensatz aus Text und Code trainiert wurde. Es kann Text generieren, Sprachen übersetzen, verschiedene Arten von kreativen Inhalten schreiben und Ihre Fragen auf informative Weise beantworten. Gemini befindet sich noch in der Entwicklung, hat aber bereits gelernt, viele Arten von Aufgaben auszuführen, einschließlich:

- Befolgen Sie Anweisungen und erledigen Sie Aufgaben sorgfältig.
- Beantworten Sie Ihre Fragen umfassend und informativ, auch wenn sie offen, herausfordernd oder seltsam sind.
- Generieren Sie verschiedene kreative Textformate, wie Gedichte, Codes, Skripte, Musikstücke, E-Mails, Briefe usw. Ich werde mein Bestes tun, um alle Ihre Anforderungen zu erfüllen.

**Technische Details**

Gemini basiert auf der Transformer-Architektur und wird mit einem massiven Datensatz aus Text und Code trainiert. Es verwendet eine Vielzahl von Techniken, um seine Leistung zu verbessern, einschließlich:

- **Aufmerksamkeit:** Ermöglicht es Gemini, sich auf die relevantesten Teile einer Eingabe zu konzentrieren.
- **Beam Search:** Ermöglicht Gemini, die besten möglichen Ausgaben zu generieren.
- **Multitasking:** Ermöglicht Gemini, mehrere Aufgaben gleichzeitig zu lernen und auszuführen.

**Anwendungen**

Gemini kann für eine Vielzahl von Anwendungen eingesetzt werden, einschließlich:

- **Generierung von kreativen Inhalten:** Gemini kann verwendet werden, um Gedichte, Codes, Skripte, Musikstücke, E-Mails, Briefe usw. zu generieren.
- **Sprachübersetzung:** Gemini kann verwendet werden, um Text von einer Sprache in eine andere zu übersetzen.
- **Beantwortung von Fragen:** Gemini kann verwendet werden, um Ihre Fragen umfassend und informativ zu beantworten, auch wenn sie offen, herausfordernd oder seltsam sind.
- **Zusammenfassen von Texten:** Gemini kann verwendet werden, um lange Textpassagen zu zusammenfassen.
- **Schreiben verschiedener Arten von kreativen Textformaten:** Gemini kann verwendet werden, um verschiedene Arten von kreativen Textformaten zu schreiben, wie Gedichte, Codes, Skripte, Musikstücke, E-Mails, Briefe usw.

Prompt 2.2: Infotext Gemini 1, Quelle: Gemini

### Ethische Überlegungen

Es ist wichtig, Gemini verantwortungsvoll und ethisch zu nutzen. Zu den ethischen Überlegungen, die bei der Verwendung von Gemini zu beachten sind, gehören:

- **Vermeidung von Vorurteilen:** Gemini wird mit einem riesigen Datensatz aus Text und Code trainiert. Dieser Datensatz kann Vorurteile enthalten, die sich in den Ausgaben von Gemini widerspiegeln können. Es ist wichtig, sich dieser Vorurteile bewusst zu sein und Maßnahmen zu ergreifen, um sie zu mindern.
- **Vermeidung von Schadenszufügung:** Gemini kann verwendet werden, um schädlichen Text zu generieren, z. B. Hassreden oder Gewaltandrohungen. Es ist wichtig, Gemini nicht zu verwenden, um anderen Schaden zuzufügen.
- **Respekt vor der Privatsphäre:** Gemini kann verwendet werden, um personenbezogene Daten zu generieren. Es ist wichtig, Gemini nicht zu verwenden, um die Privatsphäre anderer zu verletzen.

### Fazit

Gemini ist ein leistungsstarkes Tool, das für eine Vielzahl von Anwendungen eingesetzt werden kann. Es ist jedoch wichtig, Gemini verantwortungsvoll und ethisch zu nutzen.

Prompt 2.3: Infotext Gemini 2, Quelle: Gemini

### 2.1.3 Le Chat

In Prompt 2.4 sieht man die Antwort die Le Chat gibt, wenn man ihn nach einem Infotext über sich fragt. Le Chat ist ein, von dem französischen KI-Startup Mistral AI veröffentlichter Chatbot. Mistral AI startete mit seinen frei verwendbaren Sprachmodellen, also auf der Grundlage von Open Source, erfolgreich durch. Nun hat Mistral AI sein bislang größtes Modell 'Mistral Large' veröffentlicht. Diesmal allerdings nicht auf der Basis von Open Source, sondern ausschließlich über die eigene Webseite und der KI-Infrastruktur Microsoft Azure. Es lassen sich allerdings API-Keys für Programmierschnittstellen erstellen, um z.B. Mistral Large über seinen eigenen Server laufen zu lassen und so für andere User auf der eigenen Homepage verfügbar zu machen. Mit Mistral Large wurde auch Le Chat veröffentlicht, welcher aktuell kostenfrei verwendet werden kann. Le Chat bietet derzeit noch sehr wenige Funktionen an. Es stehen lediglich Texteingabe und -ausgabe zur Verfügung. Die Datengrundlage reicht nur bis 2021, weshalb es auch hier, für die Jahre 2022 bis heute, zu der Problematik der Halluzination kommen kann. Grundsätzlich kann man zwischen drei Sprachmodellen auswählen: Large, Next und Small. Large bietet überlegene Denkfähigkeit, Next ist ein Prototyp-Modell für erhöhte Kürze und Small arbeitet schnell und kosteneffektiv [4].

## 2.2 Software Engineering Prozess

Software-Engineering ist eine technische Disziplin, die sich umfassend mit der Erstellung von Software beschäftigt, beginnend bei der Konzeption, über den Betrieb, bis hin zur Wartung. Die wesentlichen Aktivitäten im Software-Engineering umfassen die Softwarespezifikation, -entwicklung, -validierung und



**Prompt: Erstell mir einen Infotext über Le Chat für meine Bachelorarbeit Einleitung**

Le Chat ist ein fortschrittliches Sprachmodell, das auf künstlicher Intelligenz (KI) und maschinellem Lernen basiert. Entwickelt von einem Team aus Experten für KI und Sprachverarbeitung, hat Le Chat die Fähigkeit, menschliche Sprache zu verstehen, zu interpretieren und darauf zu reagieren, was es zu einem wertvollen Werkzeug in verschiedenen Branchen macht.

**Funktionsweise**

Le Chat nutzt eine Kombination aus Techniken des Natural Language Processing (NLP) und des Deep Learning. NLP ermöglicht es dem Modell, die Struktur und den Kontext der menschlichen Sprache zu verstehen, während Deep Learning-Algorithmen verwendet werden, um Muster in den Daten zu erkennen und die Genauigkeit der Vorhersagen zu verbessern.

**Anwendungsbereiche**

Le Chat kann in einer Vielzahl von Anwendungen eingesetzt werden. Einige der wichtigsten Anwendungsbereiche sind:

- **Kundenservice:** Le Chat kann als virtueller Assistent eingesetzt werden, um Kundenanfragen zu beantworten, Produktempfehlungen auszusprechen und den Kunden bei der Lösung von Problemen zu unterstützen.
- **Bildung:** Le Chat kann als Tutor-System eingesetzt werden, um Schülern und Studenten bei der Lösung von Aufgaben und beim Verständnis von komplexen Konzepten zu helfen.
- **Gesundheitswesen:** Le Chat kann als virtueller Gesundheitsassistent eingesetzt werden, um Patienten bei der Verwaltung ihrer Medikamente zu unterstützen, Gesundheitstipps zu geben und in Notfällen Erste Hilfe-Anweisungen bereitzustellen.

**Vorteile**

Le Chat bietet eine Reihe von Vorteilen gegenüber herkömmlichen Sprachmodellen. Einige der wichtigsten Vorteile sind:

- **Verbesserte Genauigkeit:** Le Chat nutzt fortschrittliche Deep Learning-Algorithmen, um die Genauigkeit der Vorhersagen zu verbessern und eine bessere Benutzererfahrung zu bieten.
- **Skalierbarkeit:** Le Chat kann leicht skaliert werden, um eine große Anzahl von Anfragen gleichzeitig zu bearbeiten, was es zu einer idealen Lösung für Unternehmen mit hohem Kundenaufkommen macht.
- **Kosteneffizienz:** Le Chat kann als kosteneffiziente Alternative zu menschlichen Kundenservice-Mitarbeitern eingesetzt werden, was zu erheblichen Einsparungen für Unternehmen führen kann.

**Fazit**

Le Chat ist ein innovatives Sprachmodell, das die Fähigkeit hat, die Art und Weise, wie wir mit Maschinen interagieren, zu revolutionieren. Mit seiner fortschrittlichen Technologie und seinen vielfältigen Anwendungsmöglichkeiten ist Le Chat ein wichtiger Meilenstein in der Entwicklung der KI und ein Werkzeug, das in Zukunft eine noch größere Rolle spielen wird.

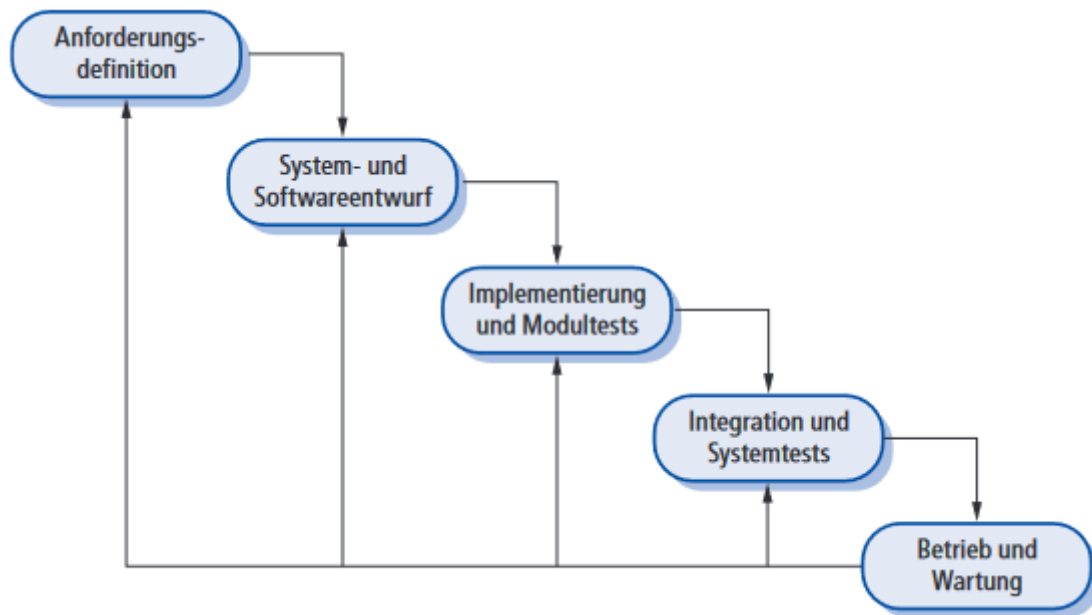


Abbildung 2.1: Wasserfallmodell, Quelle: [5]

-weiterentwicklung. Die Arbeitsweise im Software-Engineering variiert je nach den spezifischen Umständen. Für bestimmte Softwaretypen kann daher ein kreativeres, weniger formelles Vorgehen angemessen sein. Im Mittelpunkt des Software-Engineerings steht somit die Auswahl der am besten geeigneten Methode. Dieser systematische Ansatz wird oft als Softwareprozess bezeichnet. Ein Softwareprozess besteht aus einer Abfolge von Aktivitäten, die zur Erstellung eines Softwareprodukts führen. Die vier grundlegenden Aktivitäten des Software-Engineerings sind in jedem Softwareprozess enthalten. Während der Arbeit am Projekt werden die Ergebnisse in unterschiedlichen Dokumenten festgehalten. Im folgenden wird kurz auf die einzelnen Dokumente eingegangen und deren Aufbau und Inhalt beschrieben.

[5].

### 2.2.1 Besprechungsprotokoll

Alle Team- und Kundengespräche werden systematisch in Besprechungsprotokollen dokumentiert. Diese Protokolle enthalten Aufträge, Beschlüsse, Feststellungen und Informationen, die in Form von Texten, Bildern, Diagrammen usw. festgehalten werden. Aufträge müssen mit einem Fälligkeitsdatum und einem Verantwortlichen versehen sein, während Beschlüsse klar und unmissverständlich formuliert werden sollten. Im Rahmen der Anforderungsanalyse werden in Beschlüssen die Benutzeranforderungen erfasst. Feststellungen hingegen sind Beschlüsse, die keine Abstimmung benötigen, können aber ebenfalls Anforderungen an das System enthalten. Informationen bieten den Projektmitgliedern wichtige Hinweise. Dabei ist zu beachten, dass ein Besprechungsprotokoll keine Erlebniserzählung sein soll.

Der Aufbau eines Besprechungsprotokolls gestaltet sich wie folgt: Am Anfang stehen Datum, Thema, Verfasser, Teilnehmer und Verteiler. Anschließend wird eine Tabelle mit den Spalten Nummer, Art,

Beschreibung, Termin und Verantwortlicher erstellt. Die Nummer stellt eine eindeutige ID im Protokoll dar. Die Art spezifiziert, ob es sich um einen Auftrag (=A), einen Beschluss (=B), eine Feststellung (=F) oder eine Information (=I) handelt. Die Beschreibung fasst den Auftrag, den Beschluss, die Feststellung oder die Information kurz und präzise zusammen. Der Termin gibt an, bis wann der Auftrag erledigt sein muss. Der Verantwortliche benennt die Person, die den Auftrag ausführen muss, den Beschluss gefasst hat oder die Feststellung bzw. die Information geliefert hat.

### 2.2.2 Projekthandbuch

Das Projekthandbuch dient dazu, die Projektmitglieder umfassend über das Projekt zu informieren, einschließlich der Vorgeschichte, Vertragsbasis, Grundlagen, Ziele, Ergebnisse und mehr. Es bietet zudem eine Übersicht über die Projektorganisation, wie z.B. die Teamaufstellung, das Berichtswesen und die Dokumentenablage, sowie über wesentliche Projektabläufe wie Änderungsverfahren, Risikomanagement und Aufwandsverfolgung. Ziel des Projekthandbuchs ist es, neuen Teammitgliedern den Einstieg zu erleichtern und auf wichtige Dokumente hinzuweisen, die für die Projektabwicklung notwendig sind. Damit fungiert das Projekthandbuch als zentrale Anlaufstelle für das Projekt.

Das Dokument beginnt mit einer Einleitung, die die Rahmenbedingungen klärt. Dieser Abschnitt enthält den Zweck des Dokuments, Hinweise zur Redaktion, in denen festgelegt wird, wer für das Dokument verantwortlich ist, sowie Informationen zum Verteiler, also wer bei Änderungen informiert werden muss.

Im Anschluss folgt die Projektbeschreibung, die die Rahmenbedingungen des Projekts darlegt. Dieser Teil enthält einen Abschnitt zur Vorgeschichte, in dem beschrieben wird, wie es zu dem Projekt kam. Ein weiterer Abschnitt bietet eine inhaltliche Kurzdarstellung des Projekts. Danach werden in einem Abschnitt zur Vertragsbasis der zeitliche Projektrahmen, die Anzahl der Teammitglieder und der Umfang der Bearbeiter-Stunden festgelegt. Es folgt eine Beschreibung des Projektergebnisses, in der festgelegt wird, welche Produkte geliefert werden müssen. Ein weiterer Abschnitt zu sonstigen Besonderheiten beschreibt informelle Ziele.

Ein nachfolgendes Kapitel verweist auf die Risikoliste.

Daraufhin folgt ein Kapitel zur Projektorganisation, das die Teamstruktur und die Kommunikationswege beschreibt. Dieses Kapitel beginnt mit dem Teamaufbau, in dem die Aufgabenverteilung und die E-Mail-Adressen der einzelnen Teammitglieder vermerkt sind. Anschließend wird die Zusammenarbeit mit den Kunden definiert. Danach folgt ein Abschnitt über regelmäßige Pflichtbesprechungen für die Teammitglieder.

Das folgende Kapitel widmet sich der Planung und behandelt verschiedene Planungsaspekte. Der erste Abschnitt befasst sich mit dem Projektplan. Hier werden die festgelegten Meilensteine beschrieben, einschließlich der Startzeiten der einzelnen Aktivitäten, der parallelen Abläufe, der Abschlusszeiten der verschiedenen Tätigkeiten, der Entstehungszeitpunkte der Produkte und der möglichen Entscheidungs-

punkte.

Anschließend folgt ein Kapitel zur Dokumentation und Ablage, in dem beschrieben wird, wie und wo Produkte bzw. Dokumente abgelegt werden. Es beginnt mit einem Abschnitt zum Dokumenteneingang und -ausgang, in dem festgelegt wird, wie mit extern eingehenden Informationen umgegangen wird und wie ausgehende Informationen protokolliert werden. Im zweiten Abschnitt wird die Ablage und Archivierung geregelt, wobei erläutert wird, wie die Daten im Projektarchiv zu organisieren sind und in welchen Verzeichnissen die jeweiligen Daten abgelegt werden.

### 2.2.3 Risikoliste

Die Risikoliste ist ein zentrales Instrument, das dem gesamten Team einen umfassenden Überblick über bekannte Risiken und die entsprechenden Maßnahmen bietet. Der Projektleiter muss diese Risiken stets im Auge behalten, um eine fundierte Projektplanung sicherzustellen. Zusätzlich kann die Risikoliste dem Anforderungsanalysten bei der Identifikation von Anforderungen, dem Systemarchitekten bei der Wahl geeigneter technischer Lösungen und dem Testmanager bei der Entwicklung von Testfällen nützlich sein.

Der Inhalt der Risikoliste umfasst die Autoren des Dokuments, die Historie des Dokuments mit einer Zusammenfassung der Änderungen, die Liste der Risiken sowie eine Liste der möglichen Maßnahmen für jedes identifizierte Risiko. Die Historie enthält Spalten für Datum, Autor\*in und Änderungen, um später vorgenommene Modifikationen leichter nachvollziehbar zu machen.

Die Risikoliste selbst setzt sich aus mehreren Attributen zusammen. Die Identifikationsnummer ist eine eindeutige Nummer zur Identifizierung des Risikos, während die Risikobezeichnung eine kurze Beschreibung des Risikos bietet. Eine ausführliche Darstellung des Risikos findet sich in der Spalte für die Risikobeschreibung. Das Datum beschreibt, wann das Risiko identifiziert wurde, und der Autor gibt an, wer das Risiko gemeldet hat. In der Spalte für Auswirkungen werden die möglichen Folgen des Risikos beschrieben. Die Risikowahrscheinlichkeit ist eine Schätzung der Eintrittswahrscheinlichkeit des Risikos, und der Risikoschaden oder das Risikoausmaß beschreibt das potenzielle Schadensausmaß, einschließlich eventueller Vertragsstrafen. Das Risikomaß berechnet sich aus der Multiplikation von Risikowahrscheinlichkeit und Risikoschaden. Die Risikoklasse ist eine Priorisierung der Risiken, typischerweise in Kategorien wie tolerierbar, unerwünscht, kritisch oder katastrophal. Schließlich wird für jedes Risiko ein Status festgelegt, der zwischen aktiv, eingetragen und geschlossen unterscheiden kann. Die Maßnahmen zur Risikobewältigung umfassen ebenfalls mehrere Attribute. Zunächst wird der Typ der Maßnahme festgelegt, wobei zwischen Risiko verhindern, Risiko lindern oder minimieren, Risiko übertragen oder teilen und Risiko akzeptieren unterschieden werden kann. Eine detaillierte Beschreibung der Maßnahme folgt, und falls das Risiko den Status "geschlossen" erhält, wird hier die Begründung eingetragen. Wenn eine Maßnahme nicht sofort eingeleitet werden soll, wird das Ereignis, das die Maßnahme auslöst, in der Trigger-Spalte definiert. Die verantwortliche Person, die für die Umsetzung der Maßnahme zuständig ist, wird ebenfalls eingetragen. Der geplante Termin gibt an, bis wann die Maßnahme abgeschlossen sein soll, und der Ist-Termin beschreibt den voraussichtlichen Abschluss basierend auf aktueller Einschätzung. Der geplante Aufwand gibt die geschätzten Kosten der Maßnahme

an, während der Ist-Aufwand den aktuellen Aufwand widerspiegelt. Auch für die Maßnahme gibt es einen Status, bei dem man zwischen geplant, aktiv oder beendet wählen kann. Das letzte Attribut umfasst die Restrisikowahrscheinlichkeit, den Restrisikoschaden, das Restrisikomaß und die Restrisikoklasse. Dies sind die geschätzte Wahrscheinlichkeit, der geschätzte Schaden, das geschätzte Maß und die geschätzte Klasse des Restrisikos nach Durchführung der Maßnahmen.

### **2.2.4 Anforderungsspezifikation**

Die Anforderungsspezifikation beschreibt formell, was das zu entwickelnde System leisten muss. Sie beschreibt was das System leisten soll, aber nicht wie das System dies umsetzt. Das wie ist Inhalt des Architekturdokuments. In der Praxis sind Anforderungen und Systementwurf nicht trennbar. Die Anforderungsspezifikation ist oft Bestandteil des Vertrags und muss daher so vollständig und eindeutig wie möglich sein.

Die Anforderungsspezifikation hilft zum einen dem Kunden oder Endbenutzer zu beurteilen, ob das System das Gewünschte leisten wird und wie das System später bedient werden muss und zum anderen wird Sie vom Projektmanager zur Erstellung des Angebots für das System und zur Planung des detaillierten Systementwicklungsprozesses verwendet. Ebenfalls hilft es den späteren Administratoren zu entscheiden, wie das System administriert werden muss.



## 3 Anwendung von LLM Tools im Software Engineering

Dieses Kapitel untersucht ausführlich die potenziellen Anwendungsfälle von LLM Tools im Software Engineering. Zunächst wird eine umfassende Übersicht der verschiedenen Dokumente vorgestellt, die im Verlauf des Entwicklungsprozesses erstellt werden. Dabei wird jedes Dokument einzeln betrachtet und aufgezeigt, wie LLM Tools bei der Erstellung und Verwaltung dieser Dokumente eingesetzt werden können. Es wird detailliert beschrieben, in welchen Bereichen und auf welche Weise diese Tools die Entwickler unterstützen, um die Effizienz und Qualität des gesamten Entwicklungsprozesses zu verbessern.

### 3.1 Besprechungsprotokoll

Besprechungsprotokolle sind ein unverzichtbares Element im Software Engineering Prozess, da sie die Kommunikation und Dokumentation innerhalb eines Teams strukturieren und nachvollziehbar machen. Diese Protokolle erfassen die wesentlichen Punkte, die in Meetings besprochen werden, einschließlich Entscheidungen, Aufgabenverteilungen und nächste Schritte. Traditionell werden Besprechungsprotokolle manuell von einem oder mehreren Teammitgliedern erstellt, was zeitaufwendig und fehleranfällig sein kann. Der Einsatz von LLM Tools kann diesen Prozess erheblich optimieren.

LLM Tools wie ChatGPT, Google Gemini und Le Chat bieten fortschrittliche Möglichkeiten zur Unterstützung bei der Erstellung und Bearbeitung von Besprechungsprotokollen. Da die kostenlosen Versionen dieser Tools keine Audio-Dateien transkribieren können und lediglich mit Texteingaben arbeiten, entfällt die Möglichkeit einer Echtzeit-Transkription. Man könnte jedoch andere kostenfreie Sprach-zu-Text Tools nutzen, um das Meeting in Textform vorliegen zu haben. Auf diese Weise wären die LLM Tools in der Lage, die wichtigsten Punkte aus dem Text zu extrahieren und strukturiert zusammenzufassen.

Ein praktisches Beispiel für den Einsatz von LLM Tools ist die automatische Identifizierung von Aufgaben, die an bestimmte Teammitglieder verteilt wurden, und die direkte Einfügung dieser Informationen in das Protokoll. Dies erleichtert die Nachverfolgung und stellt sicher, dass keine wichtigen Punkte übersehen werden.

Eine weitere Möglichkeit für den Einsatz von LLM Tools ist die Durchsuchung historischer Besprechungsprotokolle, um relevante Informationen schnell zu finden. Dies ist besonders in großen Projekten nützlich, in denen viele Meetings stattfinden und eine Vielzahl von Informationen verwaltet werden

muss. Durch die Suchfunktion können Teammitglieder schnell auf vergangene Entscheidungen und Diskussionspunkte zugreifen.

## 3.2 Projekthandbuch

Das Projekthandbuch ist ein zentrales Dokument im Software Engineering Prozess, das alle wesentlichen Informationen und Richtlinien eines Projekts zusammenfasst. Es dient als Referenzwerk für das Projektteam und stellt sicher, dass alle Beteiligten ein gemeinsames Verständnis der Projektziele, -anforderungen und -prozesse haben. Das Handbuch umfasst typischerweise Abschnitte zu den Projektzielen, den beteiligten Personen, den Kommunikationswegen, den zu verwendenden Tools und Technologien sowie den Vorgehensweisen und Standards, die im Projekt eingehalten werden sollen.

Der Einsatz von LLM Tools kann durch die Fähigkeiten, wie die automatisierte Generierung, Aktualisierung und Verwaltung von Projektdokumentationen, an dieser Stelle gut unterstützen.

Ein Möglichkeit der Nutzung von LLM Tools ist, das Projekthandbuch kontinuierlich aktuell zu halten. Änderungen und Ergänzungen können automatisch erkannt und in das Handbuch eingepflegt werden. Beispielsweise können neue Anforderungen oder geänderte Spezifikationen, die in Besprechungen oder anderen Kommunikationskanälen besprochen wurden, direkt in das Projekthandbuch übernommen werden. Dies stellt sicher, dass das Handbuch stets den aktuellen Projektstand widerspiegelt und als verlässliche Quelle für alle Teammitglieder dient.

## 3.3 Risikoliste

Die Risikoliste identifiziert und bewertet potenzielle Risiken und legt Maßnahmen zu deren Minimierung fest. Eine sorgfältig erstellte Risikoliste trägt dazu bei, mögliche Probleme frühzeitig zu erkennen und geeignete Gegenmaßnahmen zu planen, um den Projekterfolg zu sichern. Traditionell wird die Risikoliste manuell von Projektmanagern und Teammitgliedern erstellt, was zeitaufwendig sein kann und oft eine systematische und kontinuierliche Überprüfung erfordert.

Durch den Einsatz von LLM Tools im Kontext der Risikoliste könnte dieser Prozess erheblich verbessert werden. LLM Tools wie ChatGPT, Google Gemini und Le Chat bieten fortschrittliche Möglichkeiten zur automatisierten Erstellung, Aktualisierung und Verwaltung von Risikolisten.

Ein bedeutender Vorteil der Nutzung von LLM Tools ist die Fähigkeit, große Mengen an Informationen zu analysieren und potenzielle Risiken zu identifizieren. Diese Tools können Daten aus verschiedenen Quellen wie Projektdokumentationen und Besprechungsprotokollen durchforsten, um mögliche Risiken zu erkennen und in die Risikoliste aufzunehmen. Dies würde nicht nur Zeit sparen, sondern würde auch die Genauigkeit und Vollständigkeit der Risikoliste erhöhen.



Darüber hinaus können LLM Tools dabei helfen, Risiken zu bewerten und zu priorisieren. Basierend auf historischen Daten und aktuellen Projektinformationen können diese Tools die Wahrscheinlichkeit und den möglichen Einfluss von Risiken einschätzen. Dies ermöglicht eine fundierte Entscheidungsfindung und hilft dem Projektteam, sich auf die wichtigsten Risiken zu konzentrieren.

Ein weiterer Vorteil ist die Fähigkeit der LLM Tools, kontinuierlich neue Informationen zu überwachen und die Risikoliste entsprechend zu aktualisieren. Wenn sich im Verlauf des Projekts neue Risiken ergeben oder bestehende Risiken ändern, können diese Änderungen automatisch erkannt und in die Risikoliste integriert werden. Dies stellt sicher, dass die Risikoliste stets aktuell ist und das Projektteam rechtzeitig auf neue Entwicklungen reagieren kann.

LLM Tools können zudem bei der Entwicklung von Maßnahmen zur Risikominimierung unterstützen. Basierend auf den identifizierten Risiken und den Erfahrungen aus früheren Projekten können diese Tools Vorschläge für geeignete Gegenmaßnahmen machen. Dies erleichtert es dem Projektteam, proaktiv zu handeln und potenzielle Probleme frühzeitig anzugehen.

## 3.4 Anforderungsspezifikation

Die Anforderungsspezifikation definiert die funktionalen und nicht-funktionalen Anforderungen an einem Softwareprojekt. Es bildet die Basis für das Design, die Implementierung und das Testen der Software. Eine genaue und umfassende Anforderungsspezifikation ist daher unerlässlich, um Missverständnisse zu vermeiden und sicherzustellen, dass das Endprodukt den Erwartungen der Stakeholder entspricht. Die Bestandteile der Anforderungsspezifikation wurden bereits im ?? beschrieben und erläutert.

Ein bedeutender Vorteil der Nutzung von LLM-Tools liegt in ihrer Fähigkeit, natürliche Sprache zu verstehen und in strukturierte Anforderungen zu übersetzen. Dies kann genutzt werden, um das fachliche Datenmodell, die Entitätsdiagramme für die Anwendungsfälle und die Dialoge zu erstellen. Dadurch kann man die Zeit sparen, die ansonsten für die manuelle Erstellung dieser Diagramme und Skizzen erforderlich wäre, und muss diese lediglich in Textform dokumentieren. Einfachere und weniger komplexe Diagramme könnten sogar ohne ausführliche Beschreibung erstellt werden.

Für das Anwendungsfalldiagramm kann das umfassende Wissen der LLM-Tools genutzt werden. Diese Tools können Vorschläge für Anwendungsfälle für die einzelnen Rollen im System unterbreiten, was zu einem Zeitersparnis und einer Erhöhung der Genauigkeit und Vollständigkeit des Anwendungsfalldiagramms führt. Zudem können die Rollen in Beziehung zueinander gesetzt werden.

Darüber hinaus können LLM-Tools dabei helfen, Unklarheiten und Widersprüche in den Anforderungen zu erkennen. Durch die Analyse der erfassten Anforderungen können diese Tools potenzielle Konflikte identifizieren und auf Inkonsistenzen hinweisen. Dies ermöglicht es dem Projektteam, frühzeitig Korrekturen vorzunehmen und die Qualität der Anforderungsspezifikation zu verbessern.

## 3.5 Architekturdokument

Das Architekturdokument beschreibt die grundlegende Struktur und die Designentscheidungen eines Softwareprojekts. Es dient als detaillierte Vorlage für die Entwicklung und stellt sicher, dass alle Teammitglieder ein einheitliches Verständnis der Softwarearchitektur haben. Das Dokument umfasst in der Regel Informationen über die Hauptkomponenten, deren Interaktionen, die verwendeten Technologien und die Designprinzipien, die im Projekt verfolgt werden. Auch hier wurden die Bestandteile bereits im ?? aufgezählt und erklärt.

LLM Tools können bei der Erstellung von Diagrammen und anderen visuellen Darstellungen der Softwarearchitektur helfen. Durch die automatische Generierung von Architekturdigrammen auf Basis der erfassten Informationen tragen diese Tools dazu bei, die Struktur und Interaktionen der verschiedenen Komponenten klar und übersichtlich darzustellen. Dies erleichtert das Verständnis der Architektur und unterstützt die Kommunikation innerhalb des Teams.

Ein praktisches Beispiel für den Einsatz von LLM Tools ist die automatische Erstellung von Architekturdokumenten basierend auf den initialen Anforderungen und Designentscheidungen. Diese Tools dokumentieren die grundlegenden Komponenten und deren Interaktionen sowie die verwendeten Technologien und Designprinzipien. Dadurch wird eine solide Grundlage für die weitere Entwicklung geschaffen und sichergestellt, dass alle Teammitglieder ein einheitliches Verständnis der Architektur haben.

Basierend auf der Systemübersicht aus dem Abschnitt 3.4 kann die technische Infrastruktur erstellt und die aktuellsten Versionen der Hardware spezifiziert werden. Zudem könnten LLM-Tools die Hardwarekomponenten erläutern und die erforderlichen Hardwarevoraussetzungen definieren. Das Komponentendiagramm und die Sequenzdiagramme lassen sich ähnlich wie die Diagramme aus dem Abschnitt 3.4 beschreiben und erstellen. Beim Komponentendiagramm können die Komponenten und Schnittstellen erklärt und definiert werden. Für die Sequenzdiagramme der einzelnen Anwendungsfälle könnten die benötigten Funktionen innerhalb der Komponenten inklusive der Übergabeparameter definiert und der Ablauf beschrieben werden.

## 3.6 Testspezifikation

Die Testspezifikation ist ein essenzielles Dokument im Software Engineering Prozess, das die Teststrategie und die spezifischen Testfälle für ein Softwareprojekt definiert. Sie stellt sicher, dass alle funktionalen und nicht-funktionalen Anforderungen umfassend getestet werden, um die Qualität und Zuverlässigkeit der Software zu gewährleisten. Was die Testspezifikation umfasst, wurde bereits im ?? und ?? aufgelistet.

Ein wesentlicher Vorteil der Nutzung von LLM Tools ist ihre Fähigkeit, Testfälle automatisch aus den Anforderungsspezifikationen zu generieren. Diese Tools können die Anforderungen analysieren und

darauf basierend Testfälle erstellen, die sicherstellen, dass alle Aspekte der Software umfassend getestet werden. Dies spart nicht nur Zeit, sondern erhöht auch die Vollständigkeit und Genauigkeit der Testspezifikation.

Darüber hinaus können LLM Tools dabei helfen, Testskripte für automatisierte Tests zu erstellen. Basierend auf den definierten Testfällen können diese Tools Testskripte generieren, die in verschiedenen Testumgebungen ausgeführt werden können. Dies würde die Automatisierung von Testprozessen erleichtern und zur Effizienzsteigerung im Testzyklus beitragen.

Ein weiterer Vorteil ist die Unterstützung bei der Verwaltung und Aktualisierung der Testspezifikation. LLM Tools können Änderungen in den Anforderungen oder im Code automatisch erkennen und die entsprechenden Testfälle und Testskripte aktualisieren. Dies stellt sicher, dass die Testspezifikation stets den aktuellen Stand der Softwareentwicklung widerspiegelt und alle neuen oder geänderten Funktionen angemessen getestet werden.

LLM Tools können auch dabei helfen, die Testabdeckung zu analysieren und zu verbessern. Durch die Überprüfung der vorhandenen Testfälle und die Identifizierung von Lücken können diese Tools Vorschläge zur Ergänzung der Testspezifikation machen, um eine umfassende Testabdeckung zu gewährleisten. Dies trägt dazu bei, potenzielle Fehler frühzeitig zu erkennen und die Qualität der Software zu erhöhen.

Ein praktisches Beispiel für den Einsatz von LLM Tools ist die automatische Erstellung von Testberichten. Nach der Ausführung der Tests können diese Tools die Testergebnisse analysieren und detaillierte Testberichte erstellen, die die Testabdeckung, die gefundenen Fehler und die insgesamt erreichte Qualität der Software dokumentieren. Dies erleichtert die Nachverfolgung von Fehlern und die Kommunikation der Testergebnisse an alle Stakeholder.



## **4 Praxisergebnisse und Vergleich**

TODO!!!!!!!

### **4.1 ??????**

### **4.2 ChatGPT**

### **4.3 Gemini**

### **4.4 Le Chat**

### **4.5 Vergleich**



## **5 Herausforderungen und Limitationen**

TODO!!!!!!!

### **5.1 Technische Herausforderungen**

TODO!!!!!!!!!!!!!!

### **5.2 Limitationen von LLM Tools im Software Engineering Kontext**





## **6 Zusammenfassung und Ausblick**

TODO!!!!!!!!!!

### **6.1 Zusammenfassung der Ergebnisse**

TODO!!!!!!!!!!!!

### **6.2 Implikationen für die Praxis**

Alles Produkte von Unternehmen, welche Gewinnorientiert sind. Daher Diebstahl von Infos und Herangehensweisen der Projekte wenn man diese mit dem Tool zusammenfassen lässt. Grade im militärischen Bereich.

TODO!!!!!!!!!!!!

### **6.3 Ausblick und zukünftige Entwicklungen**

TODO!!!!!!!!!!!!



# Anhang







# Abbildungsverzeichnis

2.1 Wasserfallmodell, Quelle: [5] . . . . .	10
---	----





# **Tabellenverzeichnis**



# Quellcodeverzeichnis



# Stichwortverzeichnis

????, 21

Anforderungsspezifikation, 13, 17

Anwendung von LLM Tools im Software Engineering, 15

Architekturdokument, 18

Ausblick und zukünftige Entwicklungen, 25

Besprechungsprotokoll, 10, 15

ChatGPT, 4, 21

Einleitung, 1

Gemini, 21

Google Gemini, 6

Grundlagen, 3

Herausforderungen und Limitationen, 23

Implikationen für die Praxis, 25

Le Chat, 8, 21

Limitationen von LLM Tools im Software Engineering Kontext, 23

LLM Tools, 3

Motivation, 1

Praxisergebnisse und Vergleich, 21

Projekthandbuch, 11, 16

Risikoliste, 12, 16

Struktur der Arbeit, 1

Technische Herausforderungen, 23

Testspezifikation, 18

Vergleich, 21

Zielsetzung der Arbeit, 1

Zusammenfassung der Ergebnisse, 25

Zusammenfassung und Ausblick, 25



# Literaturverzeichnis

- [1] Milfad Sadar. *Large Language Models - Grundlagen KI-getriebener Kommunikation*. Zugriff am 27. Mai 2024. 2023. URL: <https://weissenberg-group.de/was-ist-ein-large-language-model/#:~:text=Zum%20IPA%20Whitepaper-,Wie%20funktioniert%20ein%20LLM%3F,zwischen%20den%20Token%20zu%20ermitteln> (siehe Seite 4).
- [2] Jim Greene. *ChatGPT(software)*. Zugriff am 28. Mai 2024. 2024. URL: <https://eds.p.ebscohost.com/eds/detail/detail?vid=3&sid=1983b266-2f53-4090-a0f2-1aef00f51d1f%40redis&bdata=Jmxhbm9ZGUmc2l0ZT1lZHMtYGl2ZQ%3d%3d#AN=163030996&db=ers> (siehe Seite 6).
- [3] Dipl.-Ing. (FH) Stefan Luber. *Was ist Google Gemini?* Zugriff am 29. Mai 2024. 2024. URL: <https://www.bigdata-insider.de/was-ist-google-gemini-a-f855ef105c02f4533c18e913d4643007/> (siehe Seite 6).
- [4] Matthias Kindt.  
*Mistral AI: Französisches KI-Startup stellt Chatbot "Le Chat" und Kooperation mit Microsoft Azure vor.* Zugriff am 16. Mai 2024. 2024. URL: <https://www.unidigital.news/mistral-ai-franzoesisches-ki-startup-stellt-chatbot-le-chat-und-kooperation-mit-microsoft-azure-vor/> (siehe Seite 8).
- [5] Ian Sommerville. *Software Engineering*. 9. Auflage. Harlow, England: Addison-Wesley, 2010. ISBN: 978-0-13-703515-1 (siehe Seite 10).

