

## ChatGPT in Softwareprojekten

Mit KI Codequalität, Anforderungen und  
Dokumentation verbessern

# DAS INHALTS- VERZEICHNIS

» Hier geht's  
direkt  
zum Buch

<b>Vorwort</b>	<b>9</b>
<b>1 Mit Sprachmodellen reden: das Prompting</b>	<b>19</b>
KI und Machine Learning: Was ist das? .....	19
Praxisbeispiel Machine Learning .....	20
Training ist das A und O .....	23
Was ist GPT? .....	25
Faktor 1: Menge der Trainingsdaten .....	25
Faktor 2: Kontext .....	26
Prompts: Was ist das? .....	27
Verschiedene Arten von Prompts .....	28
Informationen und Code beschreiben lassen .....	29
Informationen und Code anpassen lassen .....	30
Informationen und Code vergleichen und abgleichen lassen .....	30
Aufgaben und Anforderungen, die zu neuem Code führen .....	32
Personas .....	34
Beispiele für Personas in der Softwareentwicklung .....	35
Tipps zur Identifizierung und zum Einsatz von Personas .....	36
Tipps und Tricks für das Schreiben von Prompts .....	38
Klare und präzise Formulierung .....	38
Kontext explizit vorgeben .....	38
Schlüsselwörter verwenden .....	39
Mit Varianten experimentieren .....	39
Das Modellverhalten berücksichtigen .....	40

Praktiken für das Schreiben von Prompts	40
Zero-Shot-Prompting	41
One-/Few-Shot-Prompting	43
Chain-of-Thought-Prompting	45
Was kann alles schiefgehen?	46
Generieren von nicht funktionierendem Code	47
Falsche Dokumentation	48
Unbeabsichtigte Sicherheitslücken	48
Fehlende Performanceoptimierung	48
Wie kann ich Halluzinationen vermeiden?	49
<b>2 Bessere Softwarequalität durch strukturierte Software und lesbaren Code</b>	<b>51</b>
Qualitätsmerkmale von Softwareprojekten	51
Technische Aspekte der Softwarequalität	52
Softwarearchitektur	52
Flexible Architektur bietet langfristige Vorteile	53
Höhere Anfangsinvestition führt zu langfristigen	
Einsparungen	53
Empfehlungen von ChatGPT zum Architekturansatz	54
Codestruktur	55
Bewährte Architekturmuster für eine robuste Codestruktur	56
Investitionen für langfristige Einsparungen	57
ChatGPTs Empfehlungen für eine optimale Codestruktur	57
Lesbarkeit von Code	57
Lesbarer Code als Unterstützung für neue Teammitglieder	58
Lesbarer Code unterstützt die langfristige Wartung	58
Formatierung und konsistente Benennungen für verbessertes	
Debugging	58
Refactoring und Codemigration	59
Refactoring in langfristigen Softwareprojekten	59
Risiken beim Refactoring	59
Testprozeduren und -praktiken	60

<b>3</b>	<b>Die Codequalität mit ChatGPT verbessern</b>	<b>63</b>
	Code verbessern vs. Code generieren . . . . .	63
	Code lesbarer gestalten . . . . .	69
	Lesbarer Code als syntaktisches Werkzeug . . . . .	69
	Code besser strukturieren . . . . .	75
	Codestruktur als Architektur-Hilfsmittel . . . . .	75
	Aufteilung in logische Abschnitte . . . . .	76
	Selbsterklärender Code – auch ohne Kommentare . . . . .	81
	Verwendung sinnvoller Zwischenablagen . . . . .	85
	Übersichtliche Anordnung von Codeblöcken . . . . .	88
<b>4</b>	<b>Bessere Codequalität durch Anforderungen und Dokumentation</b>	<b>93</b>
	Aspekte guter Anforderungsdefinitionen . . . . .	93
	Kundenanforderungen verstehen . . . . .	95
	Projektmanagement und Anforderungsmanagement . . . . .	98
	Von der Anforderung zur Umsetzung im Entwicklungsteam . . . .	100
	Gute Dokumentationen erstellen . . . . .	102
	Dokumentation als Kommunikationswerkzeug . . . . .	102
	Unklarheiten in Formulierungen beseitigen . . . . .	103
	Die Sichtweise wechseln . . . . .	103
	Mit Use Cases und User Stories zu einer differenzierteren Sichtweise . . . . .	104
<b>5</b>	<b>Anforderungen mit ChatGPT verbessern</b>	<b>107</b>
	Anforderungen validieren . . . . .	107
	Die manuelle Validierung . . . . .	108
	Die Validierung durch ChatGPT . . . . .	109
	Anforderungen priorisieren . . . . .	114
	Konfliktpotenzial Priorität . . . . .	115
	Methoden der Priorisierung . . . . .	116
	Wie kann ChatGPT im Priorisierungsprozess helfen? . . . . .	117
	User Stories strukturieren . . . . .	122
	Die Essenz von User Stories . . . . .	122
	Mit ChatGPT User Stories generieren und strukturieren . . . . .	123

<b>6</b>	<b>Bessere Dokumentation und ein besseres Codeverständnis durch ChatGPT</b>	<b>135</b>
	Kriterien guter Dokumentation . . . . .	136
	In welchen Bereichen kommt Dokumentation zum Einsatz? . . .	138
	Dokumentation mit ChatGPT generieren . . . . .	139
	Datenmodelle und Zusammenhänge beschreiben . . . . .	141
	Algorithmen verständlich beschreiben . . . . .	150
	Codekommentare generieren und optimieren . . . . .	156
	Mit ChatGPT Codekommentare generieren . . . . .	156
	Kommentare für Datenmodelle generieren . . . . .	157
	Validierungen im Datenmodell . . . . .	163
	Programmabläufe beschreiben . . . . .	169
	Vom Code zur API und zurück . . . . .	178
<b>7</b>	<b>Sprachmodelle: Chancen und Herausforderungen für Softwareprojekte</b>	<b>191</b>
	Herausforderungen beim Einsatz von Sprachmodellen . . . . .	192
	Sicherheitsrisiken . . . . .	192
	Qualität und Zuverlässigkeit . . . . .	193
	Lizenz- und Urheberrechtsprobleme . . . . .	194
	Abhängigkeit und Kontrollverlust . . . . .	194
	Datenschutz und Compliance . . . . .	195
	Ausblick auf den Einsatz von KI in Softwareprojekten . . . . .	196
	<b>Index</b>	<b>199</b>