

# Anforderungen mit ChatGPT verbessern

Die detaillierte Ausarbeitung von Anforderungen ist für den Erfolg eines Softwareprojekts von großer Bedeutung. Im vorangegangenen Kapitel haben wir wichtige Aspekte von Anforderungsdefinitionen skizziert und bereits erwähnt, wie wir in diesem Bereich mit ChatGPT arbeiten können. In diesem Kapitel konzentrieren wir uns auf konkrete Anwendungsbeispiele und Fragen des Projektalltags.

ChatGPT kann beispielsweise genutzt werden, um Anforderungen zu verfeinern und zu strukturieren. Als leistungsfähiges Sprachmodell, das natürliche Sprache effektiv analysiert und generiert, kann es die Zusammenarbeit der verschiedenen Stakeholder im Entwicklungsprozess unterstützen. LLMs können eingesetzt werden, um Anforderungen zu validieren und zu priorisieren oder um User Stories zu strukturieren. ChatGPT & Co. können einen wichtigen Beitrag zur Optimierung des gesamten Anforderungsprozesses leisten.

Anhand konkreter Anwendungsbeispiele sehen wir uns in diesem Kapitel an, wie wir verschiedene Teams und Interessen besser koordinieren und Missverständnissen vorbeugen können. Wir gehen auf die Validierung und Priorisierung von Anforderungen ein und zeigen auf, wie User Stories effektiv strukturiert werden können. Fallbeispiele und Best Practices machen deutlich, wie ChatGPT im Entwickleralltag für das Erarbeiten der Anforderungsdefinitionen eingesetzt werden kann.

## Anforderungen validieren

Indem wir Anforderungen validieren, stellen wir sicher, dass die zu Beginn eines Softwareprojekts definierten Anforderungen den tatsächlichen Bedürfnissen und Erwartungen auf Kunden- und auf Auftraggeberseite entsprechen. Ein sorgfältig validierter Anforderungskatalog bildet die Grundlage für eine reibungslose und erfolgreiche Umsetzung von Softwareprojekten. Wir rekapitu-

lieren zunächst, wie Anforderungen validiert werden, und betrachten dann verschiedene Ansätze, mit denen wir sicherstellen können, dass die formulierten Anforderungen kohärent, vollständig und präzise sind.

## Die manuelle Validierung

Die manuelle Validierung von Anforderungen ist eine Herangehensweise, die den Erfahrungsschatz aller teilnehmenden Teammitglieder nutzt. Hierbei überprüfen erfahrene Fachleute, aber zum Teil auch die Benutzer der Software, im Detail die formulierten Anforderungen, um potenzielle Unstimmigkeiten, Widersprüche oder unklare Formulierungen zu identifizieren. Dieser Prozess erfordert nicht nur technisches Know-how, sondern auch sehr viel Intuition und ein tiefes Verständnis der Zusammenhänge. Die manuelle Validierung ermöglicht es, subtile Nuancen und spezifische Anforderungsdetails zu erfassen, die automatisierte Ansätze möglicherweise übersehen würden.

Gerade das Verständnis des Kontexts ist besonders wichtig, wenn es darum geht, Missverständnisse oder Diskrepanzen auszuräumen.

Grundsätzlich werden hierbei folgende Aspekte geprüft.

- **Konsistenzprüfung:**  
Teammitglieder prüfen auf inkonsistente oder widersprüchliche Anforderungen. Es geht darum, sicherzustellen, dass alle formulierten Anforderungen miteinander vereinbar sind und sich nicht widersprechen.
- **Vollständigkeitsüberprüfung:**  
Das Team stellt sicher, dass alle erforderlichen Informationen in den Anforderungen enthalten sind. Fehlende Details können auf potenzielle Lücken oder Missverständnisse hinweisen, die behoben werden müssen.
- **Verständlichkeitsanalyse:**  
Es wird bewertet, wie gut die formulierten Anforderungen von verschiedenen Stakeholdern – sowohl technischen als auch nicht technischen – verstanden werden. Klare Formulierungen sind entscheidend, um sicherzustellen, dass die Umsetzung den tatsächlichen Bedürfnissen entspricht.

Diese bewährten Methoden, die auf Basis von Listen stattfinden, können mit LLMs wie ChatGPT weitaus effizienter gestaltet werden. Das Team wird gewissermaßen durch einen weiteren »Player« unterstützt, der zusätzlich einen prüfenden Blick auf die Sachlage wirft.

# Die Validierung durch ChatGPT

Durch den Einsatz von ChatGPT können verschiedene Aspekte der Anforderungen automatisch überprüft werden. Diese automatisierte Validierung stellt eine effiziente Ergänzung der manuellen Validierung dar, indem sie bestimmte Aufgaben automatisiert und somit Zeit und Ressourcen spart.

Nehmen wir an, wir hätten für einen neuen Onlineshop eine Anforderungsliste erhalten. Diese lassen wir nun zunächst nicht von Menschen, sondern von ChatGPT prüfen.

Die für dieses Beispiel genutzte Liste umfasst folgende Punkte:

Tabelle 5-1: Anforderungsliste für neuen Onlineshop

Titel	Beschreibung	Komponente/Prozess
Übersichtliche Bestellzusammenfassung	Die Bestellbestätigungs-E-Mail sollte dem Kunden eine klare und gut strukturierte Zusammenfassung seiner Bestellung inklusive Artikel, Preisen und Lieferadresse bieten.	Backend (Bestellverarbeitung), E-Mail-Versand
Integration von Transaktionsdetails	Einfügen detaillierter Transaktionsinformationen wie Bestellnummer, Zahlungsdetails und Lieferstatus in die Bestellbestätigungs-E-Mail.	Backend (Bestellverarbeitung), E-Mail-Versand
Personalisierte Dankesnachricht	Hinzufügen einer persönlichen Dankesnachricht, um die Kundenbindung zu stärken und dem Kunden gegenüber Wertschätzung auszudrücken.	Backend (Bestellverarbeitung), E-Mail-Versand
Verlinkung zum Kundensupport	Einbinden von Links zu Kundensupport-Optionen für den Fall von Fragen oder Problemen bezüglich der Bestellung.	Backend (Bestellverarbeitung), E-Mail-Versand, Kundensupport
Möglichkeit zur Druckversion	Bereitstellung einer Option, die Bestellbestätigung als druckbare Version herunterzuladen oder auszudrucken.	Backend (Bestellverarbeitung), E-Mail-Versand
Echtzeit-Lieferstatusaktualisierung	Kontinuierliche Aktualisierung des Lieferstatus in Echtzeit, damit Kunden den Fortschritt ihrer Bestellung verfolgen können.	Backend (Lieferverfolgung)
Präzise Lieferzeitenangaben	Bereitstellung genauer Lieferzeiten und -termine, um Kunden eine genaue Vorstellung vom Zustellzeitpunkt zu geben.	Backend (Lieferverfolgung)
Automatisierte Statusaktualisierungen	Automatische Benachrichtigungen an Kunden über Statusänderungen wie Versandvorbereitung, Versand und Zustellung.	Backend (Lieferverfolgung), E-Mail-Versand



Tabelle 5-1: Anforderungsliste für neuen Onlineshop (Fortsetzung)

Titel	Beschreibung	Komponente/Prozess
Verknüpfung zur Sendungsverfolgung	Integration von Links zu externen Sendungsverfolgungsdiensten, damit Kunden ihre Bestellung direkt über Drittanbieterdienste verfolgen können.	Backend (Lieferverfolgung), Frontend (Kundenkonto)
Benachrichtigungen bei Verzögerungen	Benachrichtigungen an Kunden senden, wenn es zu Verzögerungen bei der Lieferung kommt, und klare Informationen über die Gründe für die Verzögerung bereitstellen.	Backend (Lieferverfolgung), E-Mail-Versand

Hier ein Beispiel, wie ChatGPT die oben dargestellte Liste validiert und die technische und fachliche Machbarkeit prüft.

Beispiel 5-1: Prompt zur Überprüfung der technischen und fachlichen Machbarkeit

```
# Rolle
Projektmanagement für eCommerce in einem Software-Projekt

# Aufgabe
Wir prüfen die Anforderungen auf technische Machbarkeit und fachliche Validität. Bitte liefere eine Liste mit Auffälligkeiten zurück, welche in der Liste nicht valide erscheinen.

# Projektinformation
eCommerce für Schuhe + Software-Unternehmen entwickeln einen neuen Online-Shop.

# Anforderungsliste|
=== Anmerkung: Hier wird die Tabelle 5-2: Anforderungsliste für neuen Onlineshop im Markdown-Format eingefügt.===
```

**Ausgabe von ChatGPT (Auszug)**

Nach der Überprüfung der Anforderungen habe ich einige potenzielle Auffälligkeiten identifiziert:

- 1. Integration von Transaktionsdetails:
  - *Auffälligkeit:* Die Übermittlung von detaillierten Transaktionsinformationen wie Zahlungsdetails per E-Mail könnte datenschutzrechtliche Bedenken aufwerfen. Stellen Sie sicher, dass alle Datenschutzrichtlinien und -vorschriften eingehalten werden.
- 2. Verlinkung zum Kundensupport:
  - *Auffälligkeit:* Es sollte klargestellt werden, welche Art von Kundensupport-Optionen verfügbar sind und wie schnell die Kunden eine Antwort erwarten können, um deren Erwartungen zu managen.

3. Echtzeit-Lieferstatusaktualisierung:
  - *Auffälligkeit*: Der Begriff "Echtzeit" könnte irreführend sein, da er eine sofortige Aktualisierung suggeriert. Klären Sie, wie oft und in welchen Intervallen der Lieferstatus tatsächlich aktualisiert wird.
4. Automatisierte Statusaktualisierungen:
  - *Auffälligkeit*: Es sollte darauf hingewiesen werden, dass automatische Benachrichtigungen an Kunden möglicherweise von deren Präferenzen abhängig sind. Einige Kunden bevorzugen möglicherweise weniger häufige Updates.
5. Verknüpfung zur Sendungsverfolgung:
  - *Auffälligkeit*: Die Integration von Links zu externen Sendungsverfolgungsdiensten könnte potenzielle Sicherheitsrisiken mit sich bringen. Stellen Sie sicher, dass diese Dienste vertrauenswürdig sind und den Datenschutzrichtlinien entsprechen.
6. Benachrichtigungen bei Verzögerungen:
  - *Auffälligkeit*: Klären Sie, wer die Verantwortung für Verzögerungsinformationen trägt. Falls möglich, sollten klare Angaben zu den Ursachen und zu voraussichtlichen Lösungszeiten gemacht werden.

Es ist wichtig, diese Punkte mit dem Entwicklungsteam und anderen relevanten Stakeholdern zu besprechen, um sicherzustellen, dass die Anforderungen sowohl technisch als auch fachlich gültig sind und den Erwartungen der Benutzer entsprechen.

## Tabellen und Listen in Prompts verwenden

Viele Sprachmodelle, darunter auch ChatGPT, unterstützen für komplexe Formatierungen *Markdown* ([markdownguide.org](https://www.markdownguide.org)). Markdown ist eine einfache und leicht verständliche Auszeichnungssprache, die insbesondere von Entwicklerinnen und Entwicklern für die Formatierung von Texten und Code verwendet wird. Um Text »fett« oder »kursiv« zu formatieren, umschließt man den auszuzeichnenden Text lediglich mit Sternchen oder Unterstrichen. Überschriften werden durch vorangestellte Hashtags ausgezeichnet, wobei ein Hashtag für die höchste Ebene steht.

Für Listen nutzt man entweder Bindestriche, Pluszeichen oder Sternchen, gefolgt von einem Leerzeichen. Nummerierte Listen werden durch Ergänzung der Ziffern erstellt. Wenn Sie Links einfügen möchten, setzen Sie den Text in eckige Klammern und den Link in runde Klammern.

Tabellen können Sie mit einfachen Zeichen erstellen. Verwenden Sie Pipe-Symbole (|) als Begrenzung zwischen den Spalten einer Tabelle. Wird eine



zweite Zeile mittels Trennlinien (Beispiel: |---|) definiert, so wird die erste Zeile als Überschrift der Tabelle genutzt. Durch Doppelpunkte (:) in der zweiten Zeile kann man die Ausrichtung der Spalten festlegen.

Markdown ermöglicht auch das Einbetten von Codeblöcken. Um Code als Code darzustellen, umschließt man ihn mit drei Backticks (```) und fügt optional die Programmiersprache hinzu.

Mit diesen einfachen Markdown-Elementen ist es möglich, Texte übersichtlich zu strukturieren, zu formatieren und sogar einfache Tabellen für die Darstellung von Daten in ChatGPT anzulegen. Es ist eine effektive und zeitsparende Methode, um gut lesbare Inhalte zu erstellen. Wenn wir Inhalte in Form einer Markdown-Tabelle präsentieren wollen, können wir sogar bereits existierende Tabellen (z. B. Excel-Tabellen) mithilfe von Konvertern, die schnell im Internet zu finden sind, in Markdown konvertieren, aber auch wieder ins Ausgangsformat zurückführen.

Hinweis: Mit dem Release von OpenAI GPT-4o ist es ebenso möglich, über Datei-Uploads mit Bildern bzw. Screenshots zu arbeiten. In den folgenden Beispielen werden wir dennoch auf die konkrete Nutzung als Text bzw. Markdown zurückgreifen, da dies allgemein für alle Sprachmodelle geeignet ist. Ebenso ergibt sich damit die Möglichkeit, mehrere Tabellen und Informationen in einem Prompt zu nutzen und diesen Informationen auch einen weiteren Kontext (z. B. durch eine Beschreibung dessen, was in den Daten zu finden ist) mitzugeben.

Unser Prompt ist hier nach dem folgenden Schema aufgebaut.

- **Rolle:**  
Wir beschreiben die Aufgabe/Persona, die das Sprachmodell erfüllen bzw. einnehmen soll.
- **Aufgabe:**  
Wir beschreiben die zu suchenden Merkmale und wie die Ausgabe erfolgen soll.
- **Projektinformation:**  
Wir geben dem LLM zusätzlichen Kontext wie z. B. eine kurze Projektbeschreibung, um den inhaltlichen Rahmen deutlicher zu machen.
- **Liste an Anforderungsdefinitionen:**  
Wir übergeben dem LLM die Liste an Anforderungen als Markdown-Dokument.

Das Ergebnis, das wir erhalten, ist bemerkenswert und weist auf Auffälligkeiten hin, die vom Projektteam initial so vielleicht nicht berücksichtigt worden wären. Sowohl rechtliche Themen wie Datenschutz werden aufgebracht als auch das Managen der Erwartungshaltung des Kunden. Aspekte mit einem gewissen Interpretationsspielraum oder mögliche Missverständnisse werden ebenfalls deutlich vom LLM herausgestellt.

Der nächste logische Schritt bestünde natürlich darin, die Anmerkungen, die ChatGPT gemacht hat, in das Ursprungsdokument einzuarbeiten. Reflexartig würde man diese Überarbeitungen möglicherweise manuell im Text vornehmen. Im nächsten Schritt wollen wir die Überarbeitung aber von unserem neuen Teammitglied ChatGPT vornehmen lassen.

Hierzu formulieren wir einen weiteren Prompt. Für die Überarbeitung ist es wichtig, dass ChatGPT in die Lage versetzt wird, diese Änderungen durchzuführen. Das muss im Prompt-Text konkret formuliert werden, ansonsten würde ChatGPT die Anmerkungen ganz einfach in einer neuen Spalte hinzufügen.

*Beispiel 5-2: Prompt zur Überarbeitung von Anforderungen*

Arbeite die Auffälligkeiten eigenverantwortlich in die vorhandene Tabelle ein. Passe dafür die Beschreibung an.

*Tabelle 5-2: Ausgabe von ChatGPT (Auszug)*

Titel	Beschreibung	Komponente/Prozess
Übersichtliche Bestellzusammenfassung	Die Bestellbestätigungs-E-Mail sollte dem Kunden eine klare und gut strukturierte Zusammenfassung seiner Bestellung inklusive Artikel, Preisen und Lieferadresse bieten.	Backend (Bestellverarbeitung), E-Mail-Versand
Integration von Transaktionsdetails	Einfügen detaillierter Transaktionsinformationen wie Bestellnummer, Zahlungsdetails und Lieferstatus in die Bestellbestätigungs-E-Mail. Beachten Sie dabei Datenschutzrichtlinien.	Backend (Bestellverarbeitung), E-Mail-Versand
Personalisierte Dankesnachricht	Hinzufügen einer persönlichen Dankesnachricht, um die Kundenbindung zu stärken und dem Kunden gegenüber Wertschätzung auszudrücken.	Backend (Bestellverarbeitung), E-Mail-Versand
Verlinkung zum Kundensupport	Einbindung von Links zu Kundensupport-Optionen für den Fall von Fragen oder Problemen bezüglich der Bestellung. Klären Sie die verfügbaren Support-Optionen und die erwartete Antwortzeit.	Backend (Bestellverarbeitung), E-Mail-Versand, Kundensupport
Möglichkeit zur Druckversion	Bereitstellung einer Option, die Bestellbestätigung als druckbare Version herunterzuladen oder auszudrucken.	Backend (Bestellverarbeitung), E-Mail-Versand

→

Tabelle 5-2: Ausgabe von ChatGPT (Auszug) (Fortsetzung)

Titel	Beschreibung	Komponente/Prozess
Echtzeit-Lieferstatus-aktualisierung	Kontinuierliche Aktualisierung des Lieferstatus in Echtzeit, wobei zu klären ist, wie oft und in welchen Intervallen die Aktualisierung erfolgt.	Backend (Lieferverfolgung)
Präzise Lieferzeitenangaben	Bereitstellung genauer Lieferzeiten und -termine, um Kunden eine genaue Vorstellung vom Zustellzeitpunkt zu geben.	Backend (Lieferverfolgung)
Automatisierte Statusaktualisierungen	Automatische Benachrichtigungen an Kunden über Statusänderungen wie Versandvorbereitung, Versand und Zustellung. Klären Sie, ob Kunden ihre Präferenzen für die Häufigkeit solcher Updates angeben können.	Backend (Lieferverfolgung), E-Mail-Versand
Verknüpfung zur Sendungsverfolgung	Integration von Links zu externen Sendungsverfolgungsdiensten, damit Kunden ihre Bestellung direkt über Drittanbieterdienste verfolgen können. Stellen Sie sicher, dass die Dienste den Datenschutzrichtlinien entsprechen.	Backend (Lieferverfolgung), Frontend (Kundenkonto)
Benachrichtigungen bei Verzögerungen	Benachrichtigungen an Kunden senden, wenn es zu Verzögerungen in der Lieferung kommt, und klare Informationen über die Gründe für die Verzögerung bereitstellen. Stellen Sie fest, wer für die Bereitstellung dieser Informationen verantwortlich ist.	Backend (Lieferverfolgung), E-Mail-Versand

Wir sehen, dass ChatGPTs Anmerkungen jetzt direkt in die Beschreibungstexte eingeflossen sind und so zu einem konkreteren Anforderungsbild geführt haben. Mit relativ geringem Aufwand konnten Sie Ihren ursprünglichen Anforderungskatalog überprüfen und verbessern. Das Team gewinnt hierdurch größere Klarheit, und die Abstimmungsprozesse werden reibungsloser ablaufen.

## Anforderungen priorisieren

Die Priorisierung von Anforderungen ist ein entscheidender Schritt im Entwicklungsprozess, der sicherstellt, dass die benötigten Ressourcen effektiv genutzt werden, um den größtmöglichen Wert für das Projekt zu generieren. In den meisten größeren Projekten arbeitet man mit dem Konzept eines MVP (*Minimum Viable Product*), d.h., dass für den angenommenen Anwendungsfall möglichst schnell eine erste funktionstüchtige Lösung entwickelt wird, die im weiteren Entwicklungsprozess immer weiter verfeinert werden kann.



Wir sehen uns in diesem Abschnitt verschiedene Methoden der Priorisierung an und beschreiben, wie ChatGPT in diese Prozesse integriert werden kann. Als Ausgangspunkt dient uns eine bereits existierende Anforderungsliste, die wir durch die verschiedenen Aspekte der Priorisierung führen werden.

## Konfliktpotenzial Priorität

Wer kennt sie nicht, die Prioritätenlisten, auf denen gefühlt 75 % der Aufgaben mit »Prio A« gekennzeichnet sind? Die Priorisierung von Anforderungen ist nicht ohne Herausforderungen, insbesondere wenn es um das Abwägen unterschiedlicher geschäftlicher, fachlicher und technischer Aspekte geht. Das Konfliktpotenzial in diesem Prozess besteht naturgemäß in den unterschiedlichen Interessen und Perspektiven der beteiligten Stakeholder. Geschäftsziele können mit fachlichen Anforderungen kollidieren. Und technische Machbarkeit steht möglicherweise in Konflikt mit Funktionen, die sich Stakeholder wünschen.

Die Einführung von Sprachmodellen in den Priorisierungsprozess kann hier wertvollen Input bieten. Das Modell kann möglicherweise Nuancen und subtile Zusammenhänge zwischen verschiedenen Anforderungen erkennen, die von menschlichen Entscheidungsträgern gegebenenfalls übersehen werden könnten. Der Prozess der Priorisierung kann zu Spannungen führen, insbesondere wenn darum geht, verschiedene Kriterien zu gewichten. Es kann deshalb hilfreich sein, die Einblicke einer »nicht im Projektumfeld involvierten Instanz« (ChatGPT) mit in die Diskussion einzubeziehen. Auch wenn die Entscheidungen bei der Priorisierung natürlich von den Projektverantwortlichen zu treffen sind, kann es die Diskussionsbereitschaft der Beteiligten erhöhen, den vermeintlich objektiven Blick des Sprachmodells auf die Projektsituation zu nutzen.

Grundsätzlich ist es während des Priorisierungsprozesses von entscheidender Bedeutung, klare Kommunikationskanäle zu etablieren und sicherzustellen, dass alle relevanten Stakeholder in den Entscheidungsprozess eingebunden sind. Selbstverständlich ist auch dann, wenn ein LLM unterstützend genutzt wird, eine Priorisierung aus z.B. geschäftlichen oder geschäftspolitischen Gründen zwingend erforderlich. Wer kennt nicht das besondere Lieblings-Feature eines Stakeholders, das berücksichtigt werden muss, um das Projekt gegenüber der Geschäftsführung verkaufen zu können?

Die Herausforderungen der Priorisierung verdeutlichen die Notwendigkeit einer ausgewogenen und möglichst umfassenden Herangehensweise. Sowohl menschliche Expertise als auch künstliche Intelligenz, vertreten durch ChatGPT oder andere LLMs, sollten zusammenwirken, um die bestmöglichen Entscheidungen zu treffen.

# Methoden der Priorisierung

Die Priorisierung von Anforderungen kann – je nach Kontext und Projektanforderungen – mithilfe unterschiedlicher Methoden erfolgen. Dabei ist es entscheidend, die geschäftlichen, fachlichen und technischen Abhängigkeiten zu berücksichtigen, um eine umfassende und ganzheitliche Bewertung zu ermöglichen.

## Geschäftliche Priorisierung

Die geschäftliche Priorisierung orientiert sich an den strategischen Zielen des Unternehmens. Hierbei werden Anforderungen danach bewertet, wie stark sie zur Erreichung der Unternehmensziele beitragen. Eine Methode ist die *MoS-CoW-Priorisierung* (*Must-haves, Should-haves, Could-haves, Won't-haves*), bei der Anforderungen in Kategorien eingeteilt werden, um ihre Bedeutung für die Zielerreichung einzuschätzen und zu verdeutlichen.

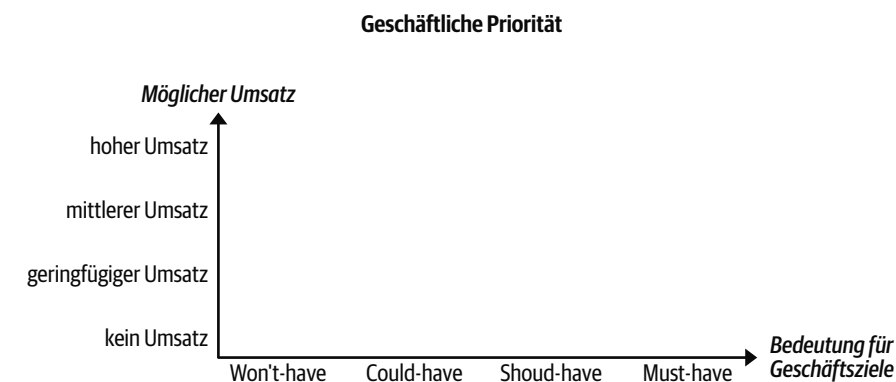


Abbildung 5-1: Geschäftliche Überlegungen bei der Priorisierung von Anforderungen – den richtigen Grad zwischen finanziellem und geschäftlichem (Produkt-) Nutzen finden

## Fachliche Priorisierung

Die fachliche Priorisierung berücksichtigt die Bedürfnisse der Endnutzer und Stakeholder, wie z.B. die Projektleitung des Kunden etc. Anforderungen werden hier danach bewertet, wie stark sie die Zufriedenheit der Benutzerinnen und Benutzer beeinflussen. Eine Möglichkeit der Priorisierung ist die *Kano-Analyse*, bei der Anforderungen in Basis-, Leistungs- und Begeisterungsfaktoren eingeteilt werden, um ihre Auswirkungen auf die Benutzerzufriedenheit zu bewerten.

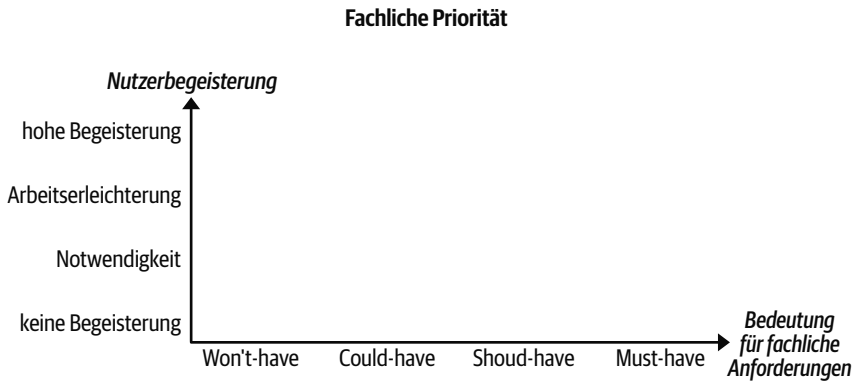


Abbildung 5-2: Fachliche Notwendigkeit und Nutzerbegeisterung bei der fachlichen Priorisierung abwägen

## Technische Priorisierung

Die technische Priorisierung bezieht sich auf die Machbarkeit und Umsetzbarkeit der Anforderungen. Hierbei werden technologische Aspekte, die Verfügbarkeit von Ressourcen und der Entwicklungsaufwand berücksichtigt. Eine Methode der technischen Priorisierung ist die Bewertung nach Aufwand und Nutzen, bei der die Kosten und der Nutzen jeder Anforderung abgewogen werden.

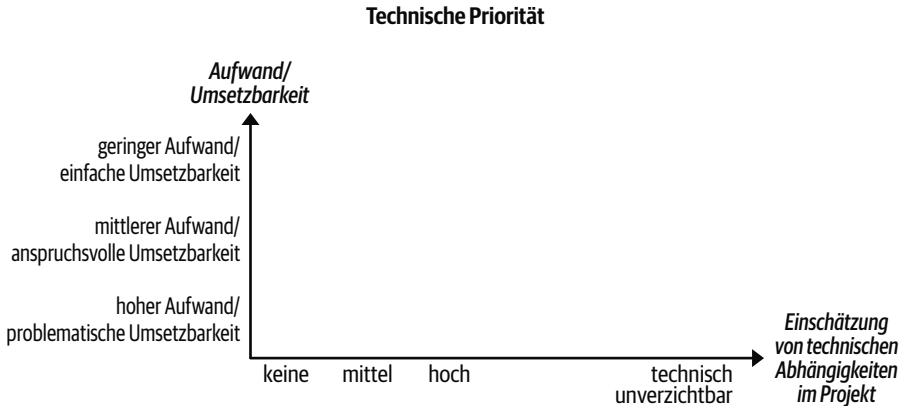


Abbildung 5-3: Was kann möglichst einfach und was muss technisch umgesetzt werden?

## Wie kann ChatGPT im Priorisierungsprozess helfen?

Die Integration von ChatGPT in Priorisierungsprozesse hilft, den Entscheidungsprozess zu optimieren und wertvolle in natürlicher Sprache formulierte Einblicke zu gewinnen. Auch an diesem Punkt kann ChatGPT genutzt werden, um Abhängigkeiten zwischen verschiedenen Anforderungen zu identifizieren.

<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>