

Bewertungskriterien für den ersten Anwendungsfall

Bewertung

Anforderungen:

	Athene 70b	CodeLlama34b- Instruct	Gemma-2- 27b	Llama3.1 70b	Qwen2- 72b
Widersprüchlichkeit	3	2	3	4	1
Verständlichkeit	4	2	4	3	2
Richtigkeit	4	3	4	4	3
Überprüfbarkeit	4	2	2	4	3
SUMME (20 möglich)	15	9	13	15	9

Gesamtdokument:

	Athene 70b	CodeLlama34b- Instruct	Gemma-2- 27b	Llama3.1 70b	Qwen2- 72b
Konsistenz	3	2	3	2	2
Nicht-Redundanz	5	1	3	3	2
Vollständigkeit	4	1	4	5	3
Prägnanz	4	2	3	3	2
SUMME (20 möglich)	16	6	13	13	9

Anmerkungen

Athene-70b:

Actor-Relationship-Game:

Ziele der zu entwickelnden Software sind direkter und technologiegetriebener als in dem vorgegebenen Dokument aus dem DevBench-Datensatz. Features und Funktionen fast vollständig in dem Vergleich zu der Referenz. Es werden die Anforderungen vollständig definiert. Allerdings werden die Anforderungen nicht in ihrer Wichtigkeit und Kritikalität bewertet. Letzteres geschieht auch nicht in dem DevBench-Dokument. Nicht ganz vollständige Dependencies. Geht mit Erwähnung der SOLID Prinzipien über das vorgegebene Dokument hinaus. Nennt dahingegen aber nicht was dieses Akronym bedeutet. Letzteres schränkt die Verständlichkeit ein. Terms und Concepts nicht ganz vollständig.

Hone:

Auch hier wieder die Ziele direkter und technischer als in dem Vorgabedokument. Features und Funktionalitäten fast vollständig. Es fehlen allerdings die Utility-Funktionen, welche in dem Vorgabedokument gegeben sind. Supporting Data description ist viel kürzer und nicht annähernd so detailliert wie in der Referenz. Bei den technischen Constraints wird angeführt, dass auch Libraries wie Pytest verwendet werden dürfen, um spätere Tests zu implementieren. Allerdings ist all dies in

dem falschen Abschnitt definiert. Zusätzlich entsteht ein Widerspruch zu dem was in dem Dependencies Abschnitt definiert ist. Dort wird erwähnt, dass keine externen Bibliotheken verwendet werden sollen. Die Akzeptanzkriterien werden wiederum ausführlicher definiert. Somit sind die Akzeptanzkriterien ausführlicher als in der Referenz. In dem letzten Abschnitt wird der Begriff CSV nicht erklärt. Dahingegen der Begriff „JSON“ schon. Dies schwächt die Stringenz des Gesamtdokuments, da JSON genauso einer Erklärung würdig ist wie der Begriff CSV. Dahingegen wird der Begriff „Delimiter“ erklärt was bei dem Vorgabedokument nicht der Fall ist.

Hybrid-Images:

Intro fast genauso wie im Beispieldokument. Letzteres gilt auch für die Ziele. Es wird aber der wichtige Aspekt nicht genannt, dass dieses Projekt in Python implementiert werden soll. Letzteres wird bereits in der Einleitung erwähnt. Features und Funktionalitäten genauso wie in dem Vorgabedokument. Die Anforderungen sind viel detaillierter formuliert als in dem Vorgabedokument. Auch die Nutzung ist weitaus detaillierter als in der Referenz. Allerdings verstecken sich in dieser Beschreibung einige unechte Anforderungen wie zum Beispiel: „Provide an intuitive, interactive GUI for exploratory users, featuring“, in denen nicht definiert ist was intuitiv konkret bedeutet. Akzeptanzkriterien sind in der Beschreibung der Eingabedaten und in Bezug auf die Überprüfung dieser detaillierter als in dem Vorgabedokument. Jedoch werden keine der zu implementierenden Funktionalitäten genannt, sondern lediglich das Gesamtergebnis. Das ist für die Stringenz des Gesamtdokuments sehr schlecht, da somit kein Bezug auf die Anforderungen und den darin festgehaltenen Funktionalitäten genommen wird.

Login-Registration:

Einleitung etwas abgespeckter als in der Referenz. Ziele sind sehr viel detaillierter als dem Vorgabedokument. So werden bereits Themen wie Hashing und Salting angesprochen. Features und Funktionalitäten nicht wirklich so detailliert wie in dem Vorgabedokument. In den technischen Constraints wird kein Wort über den Support aktueller Browser verloren. Folglich ist dieser Abschnitt nicht vollständig. Ansonsten werden aber die Anwendungsfälle nahezu vollständig betrachtet. Die Lösungen von Athene und die Referenz ergänzen sich jeweils um einen Anwendungsfall. Die Anforderungen sind ausführlicher als in dem Vorgabedokument. Allerdings etwas unvollständig und widersprüchlich. Es wird von der Validierung des Benutzernamens gesprochen, jedoch nicht definiert nach welchen Kriterien dies geschehen soll. Dies macht die Anforderungen diesbezüglich nicht überprüfbar. Auch Begriffe wie „consistent“ oder „atomic“ werden in dem Kontext dieses Projekts nicht definiert was das Verständnis erschwert. Unter dem Abschnitt Design und UI wird dann bei den Anforderungen genannt, dass alle bekannten Browser unterstützt werden sollen. Die Dependencies werden genauer aufgelistet als bei der Referenz. In dem letzten Abschnitt werden Begriffe wie Salting und Hashing zwar aufgeführt. Die Erklärung gleicht jedoch einem Zirkelschluss, wodurch die aufgeführte Erklärung keinen Mehrwert bietet und für unwissende Stakeholder dennoch eine Recherche nötig sein wird.

Readtime:

Einleitung nennt die WPM-Metrik nicht, welche für dieses Projekt wichtig ist. In den Zielen wird dies aber wiederum genannt. Ziele gehen in dem von Athene generierten Dokument über den Horizont der Referenz hinaus. Es werden neben den Zielen auch Anforderungen für die zu implementierende

Software genannt. In Features und Funktionalitäten wird die Berechnung der WPM nicht genannt, welche entscheidend für dieses Projekt ist. Ansonsten ist das Modell Athene in dem Abschnitt Features und Funktionalitäten nicht so genau wie das Vorgabedokument. Dementgegen ist der Abschnitt „Support Data Description“ dafür aber viel prägnanter als in dem Vorgabedokument. Ferner gehen die dort enthaltenen Informationen über die des Vorgabedokuments hinaus. Technische Constraints nennen über das Vorgabedokument hinaus auch die Testbibliothek Pytest. Anforderungen sind ebenfalls umfangreicher als in dem Vorgabedokument, dafür aber im Detail nicht präzise genug. Benutzung ist dementgegen detaillierter. Akzeptanzkriterien sind zumeist auch detailliert und auch umfangreicher als in dem Vorgabedokument.

CodeLlama34b-Instruct:

Actor Relationship Game:

Einleitung zu umfangreich und dadurch als solche nicht wirklich zielführend. Der erste Satz bei den Zielen erscheint sinnvoll. Der zweite gleicht jedoch einer Anforderung und trägt nicht zu dem konkreteren Verständnis über das Ziel dieses Projektes bei. Bei den Features und Funktionen ist das Ergebnis von CodeLlama sehr prägnant. Die technischen Constraints sind weitestgehend ok. Jedoch wird das Verständnis durch die Verwendung von „com.google.gson“ und „org.junit.jupiter.api“ als Begriffe für Bibliotheken das Verständnis von Stakeholdern mit geringem technischen Vorwissen erschwert. Letzteres wird dadurch verstärkt, dass die zuvor genannten Bibliotheken in dem letzten Abschnitt nicht erklärt werden. Die Akzeptanzkriterien sind leider nicht vollständig. So fehlt zum einen bei den aufgeführten Kriterien die Granularität in der Beschreibung und des Weiteren fehlt zum Beispiel die Existenz einer build.gradle für die Bibliotheken gänzlich. Entgegen dem Vorgabedokument wird in dem Ergebnis von CodeLlama jede Entität erklärt. Dies erhöht das Verständnis und sorgt dafür, dass ein eindeutiges Verständnis über diese vorliegt.

Hone:

Zunächst muss die wenig lesbare Formatierung des erstellten Texts hervorgehoben werden.

Ein Ausschnitt aus dem Dokument:

```
Introduction: Hone is a program which converts CSV files into
nested JSON files. It is written in Python and utilizes
built-in libraries like csv, json, argparse or contextlib to
parse CSV files and generate respective nested JSON files. The
program should be able to deal with different delimiters which
might be used for CSV files. Furthermore the generated nested
JSON files need to have proper indentations and sorted keys for
better readability.
```

```
Goals: The main goal of Hone is to convert CSV files into
nested JSON files. In addition to that it should be possible to
set variables like custom delimiters for the delimiters of the
CSV file which the user refers to or a schema variable which
allows the user to set a template for the output of the nested
JSON file. The aforementioned variables have to be optional.
```

```
Features and Functionalities: Hone should be able to parse CSV
files and generate respective nested JSON files. It should also
be possible to deal with different delimiters which might be
used for CSV files. Furthermore the generated nested JSON files
need to have proper indentations and sorted keys for better
readability. In addition to that there have to be two required
variables with the CSV- and the JSON filepath.
```

```
Supporting Data Description: The program should be able to deal
with different delimiters which might be used for CSV files.
Furthermore the generated nested JSON files need to have proper
indentations and sorted keys for better readability. In
addition to that there have to be two required variables with
the CSV- and the JSON filepath.
```

Die Absätze wurden dabei nachträglich eingefügt. Inhaltlich führt leider auch die Einleitung viel zu weit. Es werden bereits verwendete Technologien und Anforderungen aufgelistet. Das darauffolgende Ziel wird hingegen gut beschrieben. Bei den Features und Funktionalitäten fehlen die Utility-Funktionen und die Interaktion mit dem Benutzer über das CLI. In der Datenbeschreibung wiederholt das Dokument von CodeLlama bereits zuvor Gesagtes. Es wird erneut angemerkt wie wichtig die Lesbarkeit der generierten JSON Dateien sei. Fast wortgleich klingt dabei die Formulierung in dem vorherigen Abschnitt zu den Features und Funktionen. Zusätzlich ist die Datenbeschreibung in dem

Vergleich zu dem Vorgabedokument unvollständig und zu ungenau. Die Anforderungen nutzen zwar Wörter wie „should“ oder „have to“, um die Anforderungen zu gewichten. Jedoch werden diese nicht nummeriert. Die Dependencies sind vollständig. Allerdings treten auch hier Wiederholungen auf. Auch bei den Akzeptanzkriterien treten Wiederholungen auf. Leider sind die Akzeptanzkriterien auch unvollständig. Die Begriffserklärungen bestehen nur wiederverwendeten Satzbausteinen aus den vorherigen Abschnitten.

Hybrid Images:

Erneut muss die Formatierung des Dokuments negativ hervorgehoben werden. Das Ergebnis von CodeLlama zu diesem Projekt ist nicht ausreichend. Es fehlen viele wichtige Kapitel, sodass ein dedizierter Vergleich zwischen dem Vorgabedokument und dem Ergebnis von CodeLlama nur wenig Sinn ergibt. Die Einleitung ist in diesem Fall kurz aber vollständig. Die Funktionalitäten werden in Stichpunkten ohne weiterführende Beschreibung aufgelistet. Somit sind diese unvollständig. Es werden zwei Anforderungen genannt von denen eine nicht in dem Vorgabedokument aufzufinden ist. Die Dependencies fehlen zwar als Kapitel, werden aber in dem Abschnitt zu den Akzeptanzkriterien mit aufgeführt. Die Akzeptanzkriterien sind dabei leider unvollständig. Der Abschnitt für Begriffserklärungen fehlt gänzlich. Dies schränkt das Verständnis für die verwendeten Algorithmen und Ansätze zu dem Generieren von hybriden Bildern sehr stark ein.

Login Registration:

Das Ergebnis von CodeLlama zu diesem Projekt ist kaum zu bewerten. Es wird lediglich die Eingabe paraphrasiert und inhaltlich wiederholt. Dabei ist dieser Text weiterhin ein Fließtext ohne die vorgegebene Struktur zu berücksichtigen.

Readtime:

Auch in diesem Projekt ist das Ergebnis kaum zu bewerten. Leider wird ein kleiner Text generiert, welcher nicht der Gliederung aus der Eingabesequenz entspricht. Dieser Text wird zudem leider mehrmals wiederholt.

Gemma 2 27b:

Actor Relationship Game:

Einleitung ist vollständig und prägnant. Die Ziele gehen sogar über die des Vorgabedokuments hinaus. Enthält alle wichtigen Informationen aus dem Vorgabedokument und fasst diese in kürzerer Form zusammen. Die technischen Constraints werden besser vorgetragen als in dem Vorgabedokument. Dafür werden aber erneut „com.google.gson“ und „org.junit.jupiter.api“ genannt ohne diese Bibliotheken genauer zu erklären. Die Anforderungen gehen weit über denen des Vorgabedokuments hinaus. Es wird nicht nur zwischen funktionalen und nicht funktionalen Anforderungen unterschieden, sondern jeder Anforderung auch einen Bezeichner gegeben (z.B. „FR1“ und „NFR1“). Dies stärkt die Änderbarkeit der Anforderungsspezifikation. Jedoch sind einige nicht funktionale Anforderungen kaum überprüfbar. Zu diesem gehören Anforderungen wie zum Beispiel „The system shall be user-friendly and easy to navigate.“ oder „The system shall be responsive and provide timely results.“. Die Akzeptanzkriterien sind bis auf die Erwähnung von der „build.gradle“ Datei vollständig. Dependencies nahezu vollständig. In den Begriffserklärungen fehlen die zuvor angeführten Libraries, welche das Verständnis für Leser erschweren kann. Die vorhandenen Erklärungen sind jedoch leicht verständlich.

Hone:

Einleitung etwas länger aber dafür auch aufschlussreicher als in der Referenz. Ziele gehen über die des Vorgabedokuments hinaus und nennen beispielsweise auch die CLI-Interaktion mit dem Benutzer. Jedoch gehört dieser Punkt eher in das nachfolgende Kapitel. Ebendieses beinhaltet auch den zuvor genannten Aspekt mit der CLI-Funktionalität ist dabei nahezu vollständig. Es fehlt lediglich die Erwähnung eines Kontextmanagers. Die Datenbeschreibung ist dahingegen sehr viel kürzer und in dem Vergleich zu dem Vorgabedokument unvollständig. Die technischen Constraints sind vollständig. Es werden zusätzlich direkt die Bibliotheken mit denen das Projekt umgesetzt werden soll genannt. Bezüglich der Anforderungen geht das Dokument von Gemma zwar erneut weit über die von der Referenz erfassten Anforderungen hinaus. Allerdings ist keine der von Gemma generierten nicht funktionalen Anforderungen überprüfbar. Die nicht funktionalen Anforderungen sind somit unvollständig, da auch Begriffe wie „Usability“, „Performance“ und „Maintainability“ nicht an Bedingungen mit konkreten Zahlen, Rahmenwerken oder passenden Metriken verknüpft werden. Die Dependencies sind vollständig. Die Akzeptanzkriterien enthalten zwar alle relevanten Aspekte die auch in dem Vorgabedokument genannt werden. Allerdings ist das Kriterium für die CLI-Funktionalität nicht genau genug. Die Begriffserklärungen gehen über die des Vorgabedokuments hinaus.

Hybrid Images:

Einleitung ist vollständig. Gleiches gilt für die Ziele. Die Features und Funktionalitäten gehen über die des Vorgabedokuments hinaus und erwähnen Aspekte wie etwa das Anpassen von Parametern durch den Benutzer. Technische Anforderungen sind deckungsgleich mit dem Vorgabedokument. Die Anforderungen gehen erneut über die des Vorgabedokuments hinaus. Dabei sind aber die nicht funktionalen Anforderungen nicht überprüfbar. Für die Verwendung gilt ebenfalls, dass diese über die des Vorgabedokuments hinaus gehen. Bezüglich der Akzeptanzkriterien werden Begriffe wie „Performance“, „Usability“ und „Maintainability“ nicht an Bedingungen und konkreten Kriterien gebunden. Der Abschnitt zu der Begriffserklärung fehlt gänzlich. Die letzteren beiden Aspekte wirken sich negativ auf das Verständnis dieses Dokuments aus.

Login registration:

Einleitung ist vollständig. In den Zielen werden bereits einige Anforderungen genannt. Dies sorgt im weiteren Verlauf für Redundanzen. Die Features und Funktionalitäten werden ausreichend beschrieben. Letzteres findet zwar nicht in dem Detailgrad des Vorgabedokuments statt, dies tut dem Verständnis jedoch keinen Abbruch. Die technischen Constraints gehen über die des Vorgabedokuments hinaus. Use Cases sind vollständig. Die Anforderungen werden erneut in funktionale und nicht funktionale Anforderungen untergliedert. Bei den funktionalen Anforderungen wird von „unique usernames and passwords“ gesprochen aber nicht definiert inwiefern „unique“ Auswirkungen auf die Registrierung von Benutzern mit gleichem Nutzernamen hat. Dies mindert das Verständnis und die Vollständigkeit des Dokuments. Auch bei den nicht funktionalen Anforderungen sind drei der vier Anforderungen nicht überprüfbar. Die Datenanforderungen gehen dann wiederum weit über die des Vorgabedokuments hinaus. Es wird mit „User data will be stored in a suitable database (details to be determined).“, sogar darauf geachtet, dass unvollständige Anforderungen weiter zu detaillieren sind. Letzteres wird sogar getreu der IEEE Standards 830-1991 vollzogen. Bei den Akzeptanzkriterien sind jedoch wieder einige Probleme zu finden. „The system meets the specified performance and security requirements.“, heißt es in dem Dokument, ohne, dass diese jemals konkret definiert wurden. Somit werden die Stringenz, das Verständnis und die Widerspruchsfreiheit eingeschränkt. Bei den Begriffserklärungen erfolgen dahingegen sinnvolle Ausführungen zu „Hashing“ oder auch „Babel6.x“. Letzteres steigert dann wiederum das Verständnis. Das letzte Kapitel geht weit über den Inhalt des Vorgabedokuments hinaus.

Readtime:

Einleitung vollständig. Ziele sind in diesem Fall durch Gemma umschweifend formuliert worden. Weniger Informationen aber mehr Text. So fehlt beispielsweise die wichtige Metrik WPM, in der schlussendlich die Messung der Lesedauer erfolgen soll. Die Features und Funktionen sind nahezu vollständig. Es werden entgegen dem Vorgabedokument Beispiele für inkompatible Formate genannt, jedoch fehlt die Nennung der Default WPM-Rate. Die Datenbeschreibung ist im Vergleich zu dem Vorgabedokument unzureichend. Die technischen Constraints sind vollständig. Bei den Anforderungen folgt die Aufteilung in funktionale und nicht funktionale Anforderungen. Bei den letzteren ist nun zu beobachten, dass der Begriff „Performance“ an die Bedingung „e.g., less than 5 seconds for typical document lengths“ geknüpft wird. Dies ist leider eine Seltenheit und stellt ein gutes Beispiel dafür dar, wie mit solch unklar definierten Begriffen in derartigen Dokumenten umgegangen werden sollte. Es kann aber selbst in diesem positiven Beispiel die Frage gestellt werden, ab welcher Länge ein Dokument eine typische Länge aufweist. Ebenso werden Begriffe wie „Reliability“ und „Usability“ dahingegen nicht so genau definiert. Dies führt erneut zu dem Problem der Überprüfbarkeit. Die Akzeptanzkriterien sind dahingegen vollständig. Interessant ist, dass neben den bereits bekannten Funktionalitäten nun auch ein Akzeptanzkriterium für Unit-Tests genannt wird.

Llama3.1 70b:

Actor Relationship Game:

Einleitung vollständig. Ziele ebenso. Features und Funktionalitäten sind ebenso deckungsgleich mit dem Vorgabedokument. Die technischen Constraints gehen über das Vorgabedokument hinaus und nennen ebenso Bibliotheken für das Testen von Software. In diesem Falle werden die Bibliotheken, auch wenn nur in Klammern dahinter, wenigstens mit einem Schlagwort konkretisiert. Dies hilft bei dem Verständnis. Anforderungen sind erneut in funktionale und nicht funktionale Anforderungen aufgeteilt. In diesem Fall gilt zu beachten, dass unter den nicht funktionalen Anforderungen die SOLID-Prinzipien richtig erklärt werden. Die Akzeptanzkriterien sind nicht so umfangreich wie in dem Vorgabedokument. Das Akzeptanzkriterium: „Given two actors, the shortest path computation and visualization are accurate and efficient.“, ist zudem nicht überprüfbar. Die Abhängigkeiten sind vollständig. Bei der Begriffserklärung fehlen potenziell weiterführende Erklärungen zudem „ActorGraph“. Letzteres kann das Verständnis einschränken.

Hone:

Einleitung vollständig und für einige Stakeholder potenziell zu technisch. Die Ziele sind auch vollständig. Features und Funktionalitäten sind vollständig. Es wird sogar darauf eingegangen, ob bestimmte durch den Benutzer definierte Variablen zwingend notwendig oder optional für die Nutzung von Hone sind. Die Datenbeschreibung ist nicht vollständig. Die technischen Einschränkungen sind hingegen vollständig. Die Anforderungen sind vollständig und gehen sogar auf die notwendigen und optionalen Parameter ein. Es erfolgt jedoch keine Aufteilung in funktionale und nicht funktionale Anforderungen, was nicht zwingend negativ ist, da die gegebenen Anforderungen zumindest alle überprüfbar sind. Auch die Akzeptanzkriterien sind vollständig und überprüfbar. Die Begriffserklärungen gehen über den Rahmen des Vorgabedokuments hinaus und erklären sogar Begriffe wie „Command Line Interface“, „Custom delimiter“ oder „External dependencies“. Dies erhöht die Verständlichkeit für technisch weniger versierte Stakeholder ungemein.

Hybrid Images:

Einleitung kaum technisch aber dennoch vollständig. Es werden die Methodiken für die Fusion von Bildern nicht genannt. Letzteres ist aber in der Einleitung nicht von großer Relevanz. Features und Funktionalitäten gehen weit über das Vorgabedokument hinaus und führen auch die Implementierung einer GUI mit Bibliotheken wie Tkinter oder PyQt an. Entsprechend sind auch die technischen Anforderungen um die zuvor genannten Bibliotheken ergänzt. Auch die Erklärung der Nutzung ist ausführlicher als in dem Vorgabedokument. Jedoch wird kein Mockup mitgeliefert. Dies ist aufgrund der fehlenden Fähigkeit des LLMs Bilder zu generieren aber auch nicht erwartbar gewesen. Eines der Akzeptanzkriterien ist nicht überprüfbar und aufgrund der fehlenden Konkretisierung von „(...)user-friendly and easy to understand(...)“ nicht vollständig. Leider fehlen auch die nötigen Parameter „filters, sigma, size“ für die Verarbeitung.

Login registration:

Einleitung ist zwar vollständig. Es wird aber von „This software requirement specification outlines the functional and non-functional requirements“ gesprochen, obwohl die Anforderungen im weiteren Verlauf des Dokuments nicht in funktionale und nicht funktionale Anforderungen gegliedert werden. Ebenso wird von einer „user authentication [page]“ gesprochen, obwohl das Programm „Login-registration“ heißt. Letztere beide Aspekte schränken die Verständlichkeit und Stringenz des Dokuments stark ein. Dies setzt sich leider auch in den Zielen fort. So wird in diesem über die „login [page]“ gesprochen wobei es zuvor noch „user authentication [page]“ hieß. In Features und Funktionen heißt es dann wieder „user authentication [page]“ wobei ansonsten die Ausführungen sehr gut und prägnant sind. Die technischen Anforderungen gehen weit über die des Vorgabedokuments hinaus und führen Cloud-Lösungen für ein mögliches Deployment an. Letzteres erscheint aber zu weit gegriffen, zumal es sich um ein relativ kleines Beispielprojekt handelt. In den Use-Cases wird wieder von „login“ und „login page“ gesprochen. Anforderungen sind entgegen der Einleitung nicht in funktionale und nicht funktionale Anforderungen aufgeteilt. Eine der Anforderungen ist aufgrund fehlender Konkretisierung von „intuitive, user-friendly, and flexible“ nicht überprüfbar. Gleiches gilt für Akzeptanzkriterien. Dependencies sind vollständig. Begriffsbeschreibungen erklären auch Bibliotheken wie „Password.js“ was zu einem besseren Verständnis beiträgt.

Readtime:

Einleitung vollständig. Es wird aber nicht die WPM-Einheit genannt. Ziele bis auf WPM-Einheit vollständig. Features und Funktionalitäten gehen über die des Vorgabedokuments hinaus. So wird auch das Zählen von Sätzen und Buchstaben als Funktionalität angegeben. Auch die Kategorisierung der Komplexität von dem Eingabetext in niedrig, mittel und hoch wird als weitere Funktionalität angegeben. Gleiches gilt für die Interpunktion und durchschnittliche Satzlänge in Bezug auf die Beurteilung der Komplexität. Datenbeschreibung ist prägnant. Technische Constraints auch vollständig. Anforderungen vollständig. Benutzung sehr viel detaillierter und mit konkreteren Beispielen als in dem Vorgabedokument. Akzeptanzkriterien alle überprüfbar und mit Erwähnung des Bestehens aller Unit- und Akzeptanztests auch umfangreicher als das Vorgabedokument. Eine Randnotiz am Ende des Dokuments weist auf potenzielle Unvollständigkeit des Dokuments hin.

Qwen2 72B:

Actor relationship Game:

Einleitung sehr monoton geschrieben. Es wird immer wieder der gleiche Satzbau verwendet. Gleicht auch weniger einer Einleitung und mehr einer Auflistung von zu erfüllenden Anforderungen. Die Punkte die in den Zielen aufgeführt werden gehen zu weit über das eigentliche Ziel hinaus. Es wird

somit nicht abstrahiert, sondern eher erneut eine Auflistung von Anforderungen vorgenommen. Des Weiteren ergibt der fünfte Punkt „Utilize Java, Gradle, and other specified libraries for development.“, wenig Sinn. Es handelt sich bei dem zuvor genannten Punkt weniger um ein Ziel und viel mehr um eine Möglichkeit das eigentliche Ziel umzusetzen. Die von Qwen2 erzeugte Liste an Features und Funktionen erscheint unvollständig. Zudem werden einige Punkte zu ungenau beschrieben. Als Beispiel kann der Punkt „Efficiently process large amounts of data from the TMDB API without crashing or slowing down significantly“, dienen. Es wird nicht genau definiert ab wann die Verarbeitung der Daten als „zu langsam“ gilt. In dem Abschnitt zu den technischen Constraints sind alle relevanten Punkte aufgeführt. Jedoch nicht in dem Detailgrad wie in dem Vorgabedokument. Die Anforderungen sind unvollständig. So werden lediglich die zu implementierenden Instanzen detailliert. Einige Punkte aus den vorherigen Abschnitten, welche als Anforderung dienen sollten werden hier nicht aufgeführt. In den Dependencies fehlt auch die Nennung von Bibliotheken um JSON Dateien zu verarbeiten. In den Begriffserklärungen werden nur wenig sinnvolle Begriffe wie der Name des Programms selbst oder die TMDB API erklärt. Beide Begriffe wurden bereits in dem Rahmen der vorherigen Abschnitte erläutert. Zuletzt gilt es anzumerken, dass von nun an Qwen2 in dem erzeugten Dokument die Anforderungsspezifikation sehr häufig wiederholt.

Hone:

Die Einleitung ist vollständig aber weniger prägnant als in dem Vorgabedokument. Gleiches gilt für die Ziele. Die Features und Funktionalitäten sind fast vollständig. Es fehlt jedoch die Funktionalität, einen Dateipfad für das Speichern der entstandenen JSON Datei über die CLI angeben zu können. Die Datenbeschreibung ist im Vergleich zu dem Vorgabedokument unvollständig. Die technischen Constraints sind widersprüchlich. So wird in dem Punkt „users can choose to add other libraries if they want to use additional features like logging or more complex data manipulation.“ angeführt, dass Benutzer das Programm um weitere Bibliotheken für zusätzliche Funktionalitäten erweitern könnte. Jedoch heißt es in dem vorherigen Satz noch „No external dependencies are required for the core functionality of the program“. Somit ist jede Erweiterung, welche über das ursprünglich definierte Ziel hinaus geht eher als gänzlich neues Projekt und nicht als Erweiterung des bestehenden Projekts. Ferner erscheint es auch wenig Sinnvoll, dass der Benutzer das Programm erweitert. Dies gehört schließlich eher zu dem Aufgabenbereich der Entwickler. Die Anforderungen sind bis auf die Angabe eines Speicherpfads vollständig. Die Akzeptanzkriterien sind vollständig und überprüfbar. In den Begriffserklärungen werden bereits zuvor erklärte Begriffe erneut definiert. Hinzu kommt, dass in der Definition von dem Begriff „CLI“ Qwen2 anfängt zu halluzinieren und eine Kommunikation zwischen einem Menschen und dem LLM selbst als Text einfügt.

Hybrid Images:

Zunächst ist die nur schwer lesbare Formatierung des Dokuments negativ anzumerken. Es wurden keine Absätze gemacht und das gesamte Dokument als Fließtext durchgeschrieben. Die Einleitung ist dabei vollständig. Gleiches gilt für die Ziele. Die Features und Funktionalitäten sind dahingegen wieder unvollständig. Es fehlt die vollständige Nennung der zu implementierenden Filter. Die technischen Anforderungen sind fast vollständig. Es fehlt die Konkretisierung des Begriffs „responsive design“. Die Akzeptanzkriterien sind nicht genau genug und lassen die Überprüfung der konkreten Parameter missen. Ein Abschnitt für Begriffserklärungen existiert nicht, da Qwen2 bei der Generierung der Akzeptanzkriterien erneut anfängt zu halluzinieren. Dabei wird, wie auch schon in der Anforderungsspezifikation zu Hone, eine Kommunikation zwischen einem Menschen und dem LLM selbst als Text einfügt.

Login Registration:

Einleitung weniger technisch als in dem Vorgabedokument. Vollständig aber nicht so prägnant. Ziele sind vollständig. Die Features und Funktionalitäten sind nicht so umfangreich wie in dem Vorgabedokument aber dennoch ausreichend. Technische Anforderungen sind vollständig und prägnanter als in dem Vorgabedokument. Die betrachteten Anwendungsfälle gehen über die des Vorgabedokuments hinaus. Die Anforderungen des Vorgabedokuments und des Ergebnisses von Qwen2 ergänzen sich gegenseitig. So nennen beide Dokumente zwei Anforderungen, welche in dem jeweils anderen Dokument nicht vertreten sind. Die Akzeptanzkriterien sind vollständig und überprüfbar. Auch die Dependencies sind vollständig. Bei den Begriffserklärungen folgen dann wiederum obsoletere Erklärungen zu Begriffen wie „Registration“ oder „Delete Button“. Zuletzt fügt das Modell die Eingabesequenz zu der generierten Anforderungsspezifikation hinzu.

Readtime:

Einleitung ist vollständig. Ziele gehen bis auf die WPM-Rate über die des Vorgabedokuments hinaus. Ebenso fehlt die Nennung der Default WPM-Rate in dem Abschnitt zu den Features und Funktionen. Ansonsten vollständig. Die Datenbeschreibung ist in dem Vergleich zu dem Vorgabedokument deutlich kürzer. Es fehlen eingehende Beispiele für das Vorgehen mit den Dateien verschiedener Formate. Die technischen Constraints sind dann wiederum vollständig. Gleiches gilt auch für die Anforderungen, welche allesamt überprüfbar sind. Bei den Akzeptanzkriterien fehlt die Nennung des Umgangs mit unterschiedlichen WPM-Raten. Abgesehen davon sind die Akzeptanzkriterien vollständig und überprüfbar. Qwen2 beginnt dann jedoch wieder mit dem Halluzinieren und fügt eine Konversation über die Entwicklung einer neuen Programmiersprache an.