AQI Automotive Quality Institute

Einsatz von KI zur maschinellen Programmierung

Anleitung zur Excel-basierten Risikokarte, um in der Praxis den Einsatz von LLMs in Softwareentwicklungsprozessen systematisch bewerten und abzusichern zu können

Dr. Björn Schünemann (bjoern.schuenemann@aqigmbh.de)
Dr. Jürgen Großmann (juergen.grossmann@fokus.fraunhofer.de)

Anleitung zur Excel-basierten Risikokarte

Übersicht & Zielstellung

AQI Automotive Quality Institute

Allgemein

- Risikokarte soll dabei helfen, eine Übersicht zu schaffen, welche Aspekte im Hinblick auf die Nutzung eines LLMs für eine Aufgabe in einem konkreten Kontext zu beachten sind.
- Zudem zeigt sie eine Reihe an möglichen Lösungs- und Präventionsansätzen sowie Vorschläge zu einem sicheren Umgang im Entwicklungsprozess.
- Durch die Nutzung der Risikokarte lassen sich außerdem notwendige Verhaltensweisen und Maßnahmen im Umgang mit dem LLM ableiten.

Anwendung der Risikokarte

Risiken validieren und überwachen

- **Sicherstellung der Verantwortlichkeit durch** klare Zuständigkeiten für verschiedene Automatisierungsschritte.
- **Qualitätssicherungsmaßnahmen** (Prüfmechanismen), die je nach Automatisierungsgrad und -autonomie in regelmäßigen Abständen durchgeführt werden.
- **Risikoabschätzung und -bewältigung:** durch Überprüfungen des Risikoindex und der entsprechenden Mitigationsmaßnahmen sicherstellen.

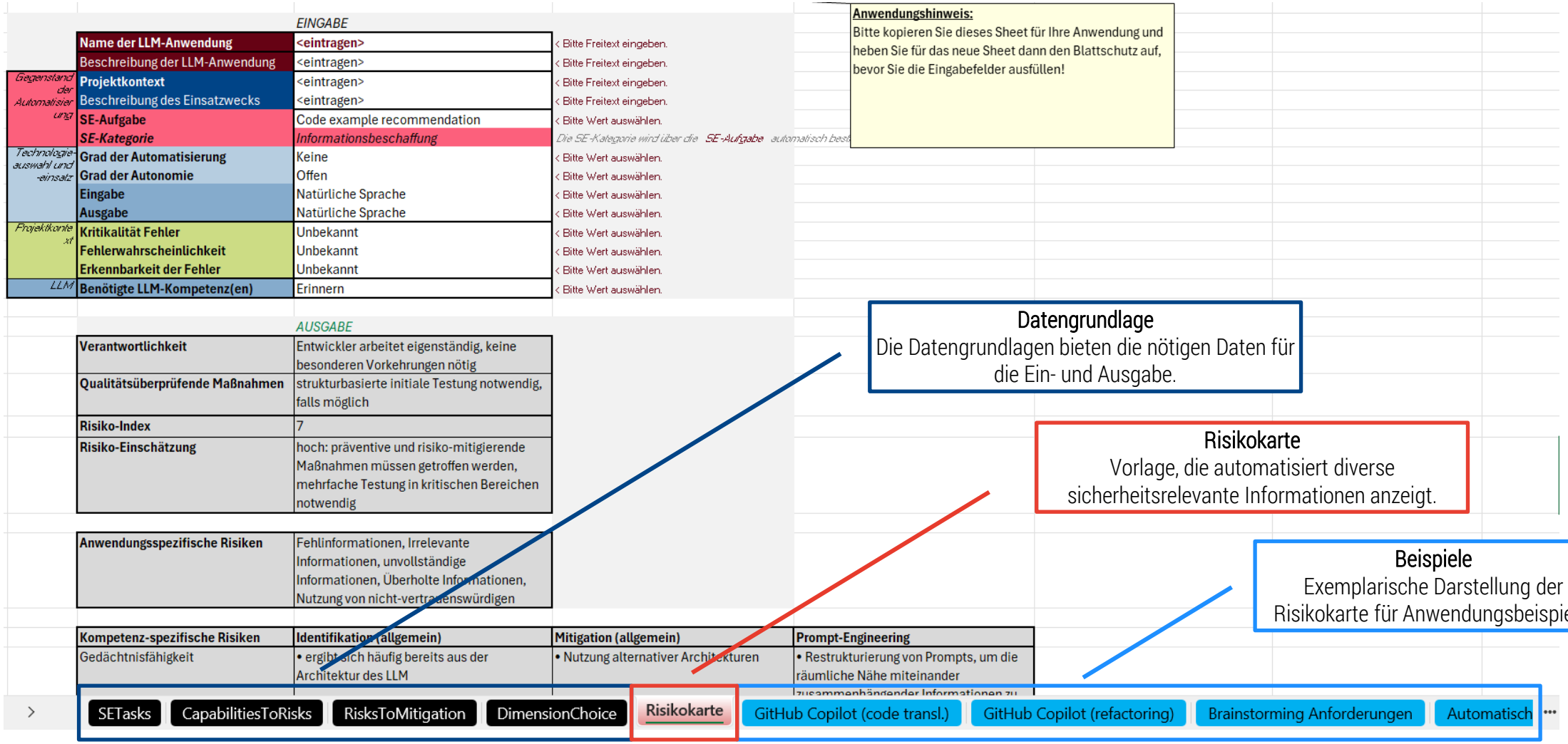
Übertragung auf Prozesse

→ Ziel ist es, Unternehmen und Entwickler bei der sicheren und effizienten Integration von LLMs in ihre Prozesse zu unterstützen und die potenziellen Risiken frühzeitig sichtbar zu machen, sodass geeignete Gegenmaßnahmen getroffen werden können.

Detaillierungsebenen der Risikokarte

Übersichtliche Darstellung der Arbeitsmappen

AQI | Automotive Quality Institute



EINGABE

	Name der LLM-Anwendung	<eintragen>	< Bitte Freitext eingeben.
	Beschreibung der LLM-Anwendung	<eintragen>	< Bitte Freitext eingeben.
Gegenstand der Automatisierung	**Projektkontext**	<eintragen>	< Bitte Freitext eingeben.
	Beschreibung des Einsatzwecks	<eintragen>	< Bitte Freitext eingeben.
	SE-Aufgabe	Code example recommendation	< Bitte Wert auswählen.
	SE-Kategorie	*Informationsbeschaffung*	*Die SE-Kategorie wird über die SE-Aufgabe automatisch best*
Technologieauswahl und -einsatz	**Grad der Automatisierung**	Keine	< Bitte Wert auswählen.
	Grad der Autonomie	Offen	< Bitte Wert auswählen.
	Eingabe	Natürliche Sprache	< Bitte Wert auswählen.
	Ausgabe	Natürliche Sprache	< Bitte Wert auswählen.
Projektkontext	**Kritikalität Fehler**	Unbekannt	< Bitte Wert auswählen.
	Fehlerwahrscheinlichkeit	Unbekannt	< Bitte Wert auswählen.
	Erkennbarkeit der Fehler	Unbekannt	< Bitte Wert auswählen.
LLM	**Benötigte LLM-Kompetenz(en)**	Erinnern	< Bitte Wert auswählen.

Anwendungshinweis:
Bitte kopieren Sie dieses Sheet für Ihre Anwendung und heben Sie für das neue Sheet dann den Blattschutz auf, bevor Sie die Eingabefelder ausfüllen!

AUSGABE

Verantwortlichkeit	Entwickler arbeitet eigenständig, keine besonderen Vorkehrungen nötig
Qualitätsüberprüfende Maßnahmen	strukturbasierte initiale Testung notwendig, falls möglich
Risiko-Index	7
Risiko-Einschätzung	hoch: präventive und risiko-mitigierende Maßnahmen müssen getroffen werden, mehrfache Testung in kritischen Bereichen notwendig

Anwendungsspezifische Risiken	Fehlinformationen, Irrelevante Informationen, unvollständige Informationen, Überholte Informationen, Nutzung von nicht-vertrauenswürdigen

Kompetenz-spezifische Risiken	**Identifikation (allgemein)**	**Mitigation (allgemein)**	**Prompt-Engineering**
Gedächtnisfähigkeit	• ergibt sich häufig bereits aus der Architektur des LLM	• Nutzung alternativer Architekturen	• Restrukturierung von Prompts, um die räumliche Nähe miteinander zusammenhängender Informationen zu

SETasks | CapabilitiesToRisks | RisksToMitigation | DimensionChoice | Risikokarte | GitHub Copilot (code transl.) | GitHub Copilot (refactoring) | Brainstorming Anforderungen | Automatisch

Datengrundlage
Die Datengrundlagen bieten die nötigen Daten für die Ein- und Ausgabe.

Risikokarte
Vorlage, die automatisiert diverse sicherheitsrelevante Informationen anzeigt.

Beispiele
Exemplarische Darstellung der Risikokarte für Anwendungsbeispiele.

Charakterisierung entlang der Taxonomie

Excel-basierte Anwendung liefert Risikoeinschätzung und -indizes

EINGABE

Name der LLM-Anwendung	**GitHub Copilot (code translation)**	< Bitte Freitext eingeben.
Beschreibung der LLM-Anwendung	Die Anwendung übersetzt Code von einer Programmiersprache in eine andere	< Bitte Freitext eingeben.
Projektkontext Beschreibung des Einsatzwecks	Entwicklung Seriensteuergerät Einsatz in der C-Kodierung eines Seriensteuergeräts für die Richtungsanzeige im Fahrzeug.	< Bitte Freitext eingeben. < Bitte Freitext eingeben.
SE-Aufgabe	Programming language translation	< Bitte Wert auswählen.
SE-Kategorie	*Inhaltserstellung*	*Die SE-Kategorie wird über die SE-Aufgabe automatisch bestimmt*
Grad der Automatisierung	Führt aus, wenn kein Einwand vorliegt	< Bitte Wert auswählen.
Grad der Autonomie	Unbestimmt	< Bitte Wert auswählen.
Eingabe	Programmcode	< Bitte Wert auswählen.
Ausgabe	Programmcode	< Bitte Wert auswählen.
Kritikalität Fehler	Hoch	< Bitte Wert auswählen.
Fehlerwahrscheinlichkeit	Hoch	< Bitte Wert auswählen.
Erkennbarkeit der Fehler	Schwer	< Bitte Wert auswählen.
Benötigte LLM-Kompetenz(en)	Anwenden	< Bitte Wert auswählen.

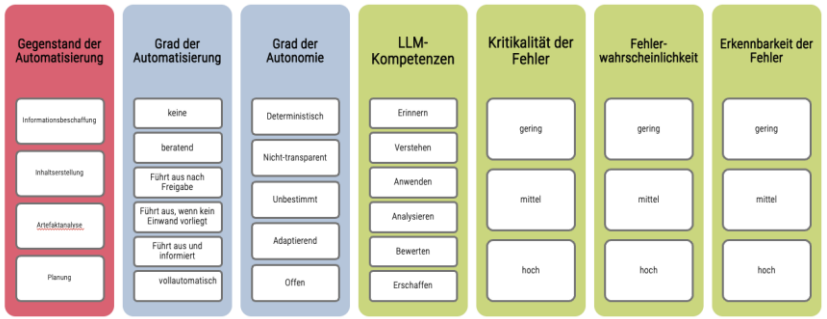
AUSGABE

Verantwortlichkeit	Verantwortung des Monitorings beim Entwickler, anfragenbasierte Überprüfung notwendig	
Qualitätsüberprüfende Maßnahmen	vollständige wiederholte Testung notwendig	
Risiko-Index	10	
Risiko-Einschätzung	hoch: präventive und risiko-mitigierende Maßnahmen müssen getroffen werden, mehrfache Testung in kritischen Bereichen notwendig	
Anwendungsspezifische Risiken	Halluzinationen, Schwachstellen, Teillösungen, Inkonsistente Lösungen, Nicht-Determinismus	
Kompetenz-spezifische Risiken	**Identifikation (allgemein)**	**Mitigation (allgemein)**
Gedächtnisfähigkeit	• ergibt sich häufig bereits aus der Architektur des LLM	• Nutzung alternativer Architekturen

Beispiel: GitHub Copilot

Die informativen Felder beschreiben den Anwendungskontext.

Auswahlfelder entlang der Dimensionen der Taxonomie



Anhand der Eingabe wird eine spezifische Auswahl der Verantwortlichkeiten, Risiken und Risikominderungsmaßnahmen zugeordnet.

Welche Schritte muss ich tun, um die Excel-basierte Risikokarte zu benutzen?

AQI Automotive Quality Institute

Fraunhofer FOKUS

Schritt 01

- 01.1 Risikokarte kopieren
- 01.2 Blattschutz aufheben
- 01.3 Neue Risikokarte umbenennen

Schritt 02

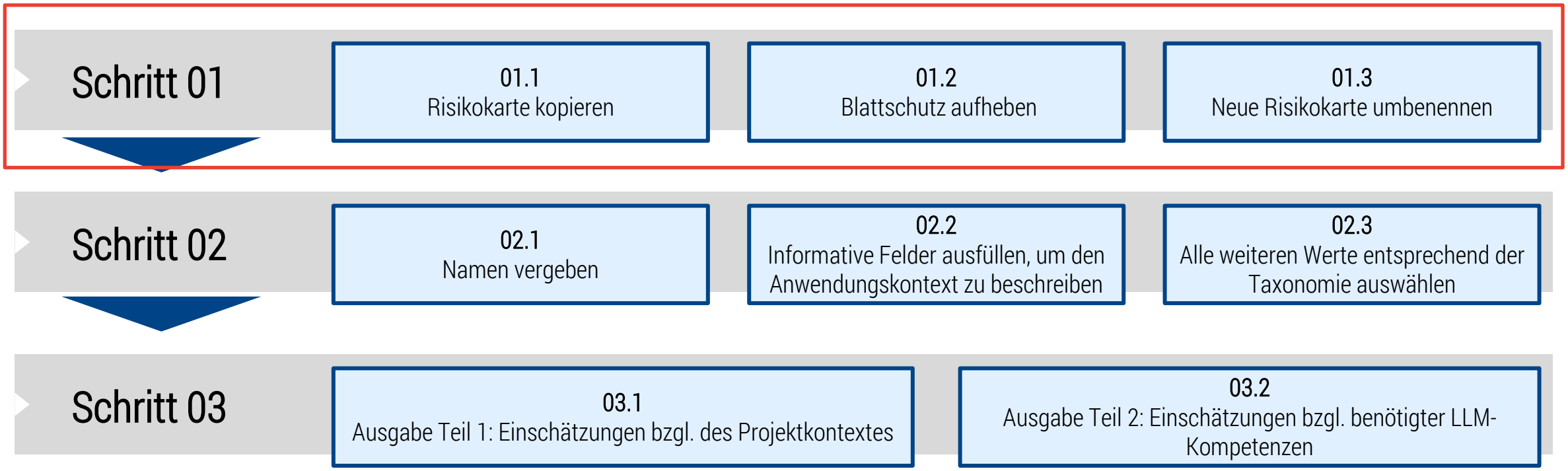
- 02.1 Namen vergeben
- 02.2 Informative Felder ausfüllen, um den Anwendungskontext zu beschreiben
- 02.3 Alle weiteren Werte entsprechend der Taxonomie auswählen

Schritt 03

- 03.1 Ausgabe Teil 1: Einschätzungen bzgl. des Projektkontextes
- 03.2 Ausgabe Teil 2: Einschätzungen bzgl. benötigter LLM-Kompetenzen

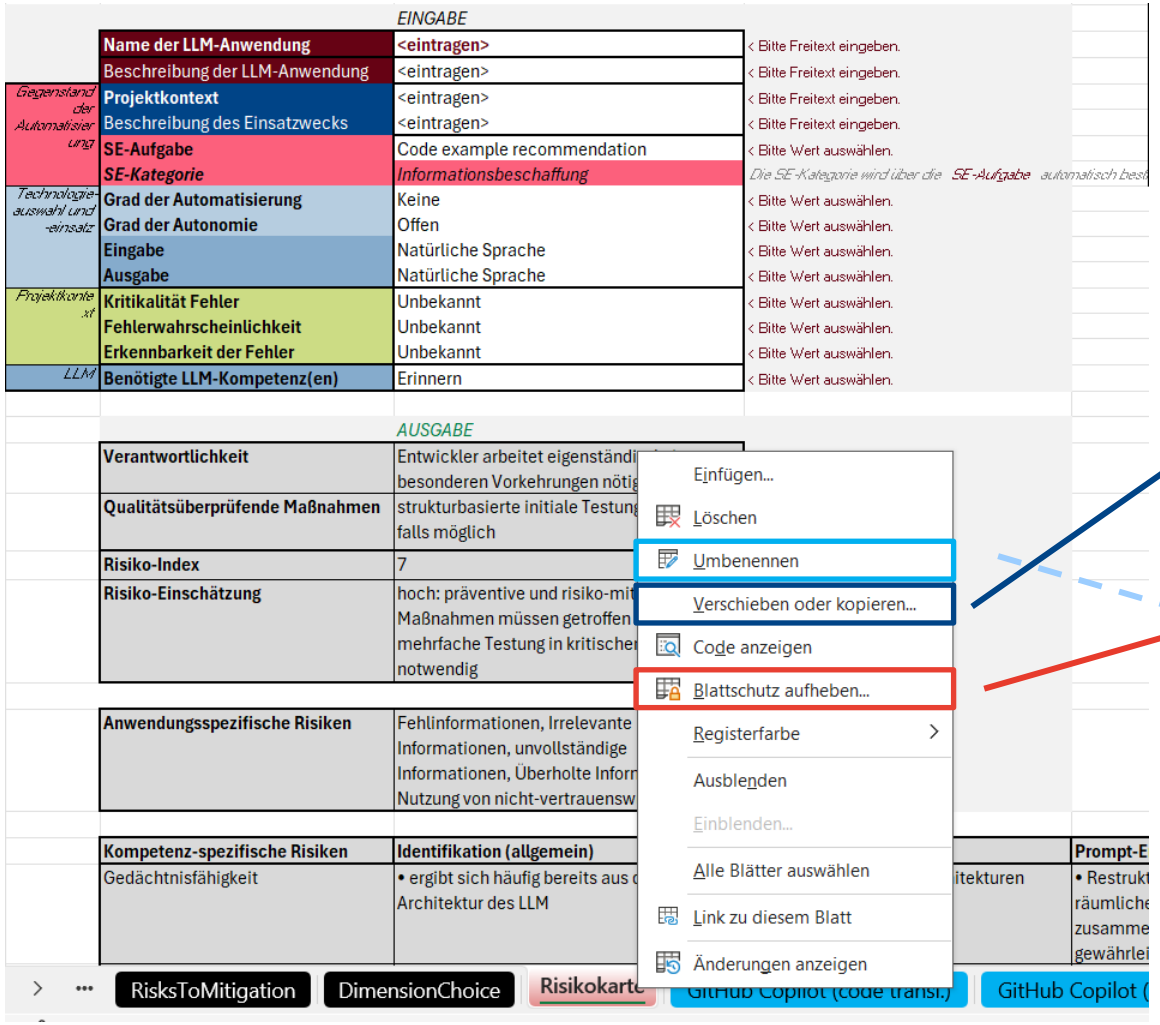
Detaillierungsebenen der Prozessbeschreibungen

Schritt 01: Start zur Nutzung der Risikokarte



Schritt 01

… Start zur Nutzung der Risikokarte



		EINGABE	
	Name der LLM-Anwendung	**<eintragen>**	< Bitte Freitext eingeben.
	Beschreibung der LLM-Anwendung	<eintragen>	< Bitte Freitext eingeben.
Gegenstand der Automatisierung	**Projektkontext**	<eintragen>	< Bitte Freitext eingeben.
	Beschreibung des Einsatzwecks	<eintragen>	< Bitte Freitext eingeben.
	SE-Aufgabe	Code example recommendation	< Bitte Wert auswählen.
	SE-Kategorie	*Informationsbeschaffung*	*Die SE-Kategorie wird über die SE-Aufgabe automatisch best*
Technologieauswahl und -einsatz	**Grad der Automatisierung**	Keine	< Bitte Wert auswählen.
	Grad der Autonomie	Offen	< Bitte Wert auswählen.
	Eingabe	Natürliche Sprache	< Bitte Wert auswählen.
	Ausgabe	Natürliche Sprache	< Bitte Wert auswählen.
Projektkontext	**Kritikalität Fehler**	Unbekannt	< Bitte Wert auswählen.
	Fehlerwahrscheinlichkeit	Unbekannt	< Bitte Wert auswählen.
	Erkennbarkeit der Fehler	Unbekannt	< Bitte Wert auswählen.
LLM	**Benötigte LLM-Kompetenz(en)**	Erinnern	< Bitte Wert auswählen.

	AUSGABE	
Verantwortlichkeit	Entwickler arbeitet eigenständi besonderen Vorkehrungen nötig	
Qualitätsüberprüfende Maßnahmen	strukturbasierte initiale Testung falls möglich	
Risiko-Index	7	
Risiko-Einschätzung	hoch: präventive und risiko-mit Maßnahmen müssen getroffen mehrfache Testung in kritischer notwendig	
Anwendungsspezifische Risiken	Fehlinformationen, Irrelevante Informationen, unvollständige Informationen, Überholte Inform Nutzung von nicht-vertrauensw	

Kompetenz-spezifische Risiken	**Identifikation (allgemein)**		**Prompt-E**
Gedächtnisfähigkeit	• ergibt sich häufig bereits aus d Architektur des LLM	itekturen	• Restrukt räumliche zusamme gewährlei

Einfügen...
Löschen
Umbenennen
Verschieben oder kopieren...
Code anzeigen
Blattschutz aufheben...
Registerfarbe
Ausblenden
Einblenden...
Alle Blätter auswählen
Link zu diesem Blatt
Änderungen anzeigen

RisksToMitigation | DimensionChoice | Risikokarte | GitHub Copilot (code transl.) | GitHub Copilot (

01.1
Risikokarte kopieren

01.2
Blattschutz aufheben

01.3
Neue Risikokarte umbenennen

Detaillierungsebenen der Prozessbeschreibungen

Schritt 02: Die Eingabemaske

AQI Automotive Quality Institute

Fraunhofer FOKUS

Schritt 01

- 01.1 Risikokarte kopieren
- 01.2 Blattschutz aufheben
- 01.3 Neue Risikokarte umbenennen

Schritt 02

- 02.1 Namen vergeben
- 02.2 Informative Felder ausfüllen, um den Anwendungskontext zu beschreiben
- 02.3 Alle weiteren Werte entsprechend der Taxonomie auswählen

Schritt 03

- 03.1 Ausgabe Teil 1: Einschätzungen bzgl. des Projektkontextes
- 03.2 Ausgabe Teil 2: Einschätzungen bzgl. benötigter LLM-Kompetenzen

Schritt 02

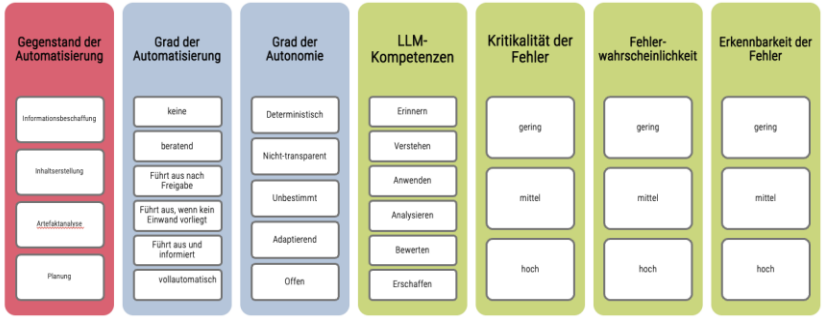
… die Eingabemaske

	EINGABE	
Name der LLM-Anwendung	**GitHub Copilot (code translation)**	< Bitte Freitext eingeben.
Beschreibung der LLM-Anwendung	Die Anwendung übersetzt Code von einer Programmiersprache in eine andere	< Bitte Freitext eingeben.
Projektkontext Beschreibung des Einsatzwecks	Entwicklung Seriensteuergerät Einsatz in der C-Kodierung eines Seriensteuergeräts für die Richtungsanzeige im Fahrzeug.	< Bitte Freitext eingeben. < Bitte Freitext eingeben.
SE-Aufgabe	Programming language translation	< Bitte Wert auswählen.
SE-Kategorie	*Inhaltserstellung*	*Die SE-Kategorie wird über die SE-Aufgabe automatisch bestimmt*
Grad der Automatisierung	Führt aus, wenn kein Einwand vorliegt	< Bitte Wert auswählen.
Grad der Autonomie	Unbestimmt	< Bitte Wert auswählen.
Eingabe	Programmcode	< Bitte Wert auswählen.
Ausgabe	Programmcode	< Bitte Wert auswählen.
Kritikalität Fehler	Hoch	< Bitte Wert auswählen.
Fehlerwahrscheinlichkeit	Hoch	< Bitte Wert auswählen.
Erkennbarkeit der Fehler	Schwer	< Bitte Wert auswählen.
Benötigte LLM-Kompetenz(en)	Anwenden	< Bitte Wert auswählen.

02.1
Beispiel: GitHub Copilot

02.2
Die informativen Felder beschreiben den Anwendungskontext.

02.3
Auswahlfelder entlang der Dimensionen der Taxonomie



Die Eingabemaske

Überblick & Beispiel

		EINGABE
	Name der LLM-Anwendung	**<eintragen>**
	Beschreibung der LLM-Anwendung	<eintragen>
Gegenstand der Automatisierung	**Projektkontext**	<eintragen>
	Beschreibung des Einsatzwecks	<eintragen>
	SE-Aufgabe	Code example recommendation
	SE-Kategorie	*Informationsbeschaffung*
Technologie-auswahl und -einsatz	**Grad der Automatisierung**	Keine
	Grad der Autonomie	Offen
	Eingabe	Natürliche Sprache
	Ausgabe	Natürliche Sprache
Projektkontext	**Kritikalität Fehler**	Unbekannt
	Fehlerwahrscheinlichkeit	Unbekannt
	Erkennbarkeit der Fehler	Unbekannt
LLM	**Benötigte LLM-Kompetenz(en)**	Erinnern

		EINGABE
	Name der LLM-Anwendung	**GitHub Copilot (code translation)**
	Beschreibung der LLM-Anwendung	Die Anwendung übersetzt Code von einer Programmiersprache in eine andere
Gegenstand der Automatisierung	**Projektkontext**	Entwicklung Seriensteuergerät
	Beschreibung des Einsatzwecks	Einsatz in der C-Kodierung eines Seriensteuergeräts für die Richungsanzeige im Fahrzeug.
	SE-Aufgabe	Programming language translation
	SE-Kategorie	*Inhaltserstellung*
Technologie-auswahl und -einsatz	**Grad der Automatisierung**	Führt aus, wenn kein Einwand vorliegt
	Grad der Autonomie	Adaptierend
	Eingabe	Programmcode
	Ausgabe	Programmcode
Projektkontext	**Kritikalität Fehler**	Hoch
	Fehlerwahrscheinlichkeit	Mittel
	Erkennbarkeit der Fehler	Mittel
LLM	**Benötigte LLM-Kompetenz(en)**	Anwenden

Die Eingabefelder entsprechen den Dimensionen aus der Taxonomie zur Charakterisierung einer Automatisierung.

Die Eingabemaske Schritte 1 – 6

Definition der LLM-Anwendung

	EINGABE	
Name der LLM-Anwendung	**<eintragen>**	1
Beschreibung der LLM-Anwendung	<eintragen>	2
Projektkontext	<eintragen>	3
Beschreibung des Einsatzwecks	<eintragen>	4
SE-Aufgabe	Code example recommendation	5
SE-Kategorie	*Informationsbeschaffung*	6
Grad der Automatisierung	Keine	7
Grad der Autonomie	Offen	8
Eingabe	Natürliche Sprache	9
Ausgabe	Natürliche Sprache	10
Kritikalität Fehler	Unbekannt	11
Fehlerwahrscheinlichkeit	Unbekannt	12
Erkennbarkeit der Fehler	Unbekannt	13
Benötigte LLM-Kompetenz(en)	Erinnern	14

Schritte 1 – 6

1. Anlegen eines Namens für die LLM-Anwendung
2. Kurze Beschreibung der Anwendung
3. Kurze Beschreibung des Projektkontextes
4. Kurze Beschreibung des Einsatzzwecks (welche Aufgabe die LLM-Anwendung übernehmen / wofür die Risikokarte erstellt wird)
5. Die konkrete SE-Aufgabe, wofür die LLM-Anwendung eingesetzt werden soll (alle mögliche Werte werden in der Dropdown-Auswahlliste bereitgestellt)
6. Kein Eingabefeld! – zeigt die Kategorie an, zu der die in Feld 5 ausgewählte SE-Aufgabe gehört

Charakterisierung von Automatisierung

Welche Faktoren beeinflussen die Qualität in der Automatisierung?

Gegenstand der Automatisierung	• Welche Tätigkeit wird automatisiert?
Grad der Automatisierung	• Wie ist die Arbeitsteilung Mensch/Maschine?
Grad der Autonomie	• Welche Freiheitsgrade hat die Maschine in der Automatisierung?
Kritikalität der durch Automatisierung induzierter Fehler	Wie kritisch sind die Fehler, die durch die Automatisierung entstehen können?
Wahrscheinlichkeit für durch die Automatisierung induzierte Fehler	• Wie wahrscheinlich ist das Auftreten neuer Fehler durch die Automatisierung?
Erkennbarkeit der durch die Automatisierung induzierten Fehler	• Wie einfach sind diese neuen Fehler zu erkennen?

Schema zur Charakterisierung von Automatisierung

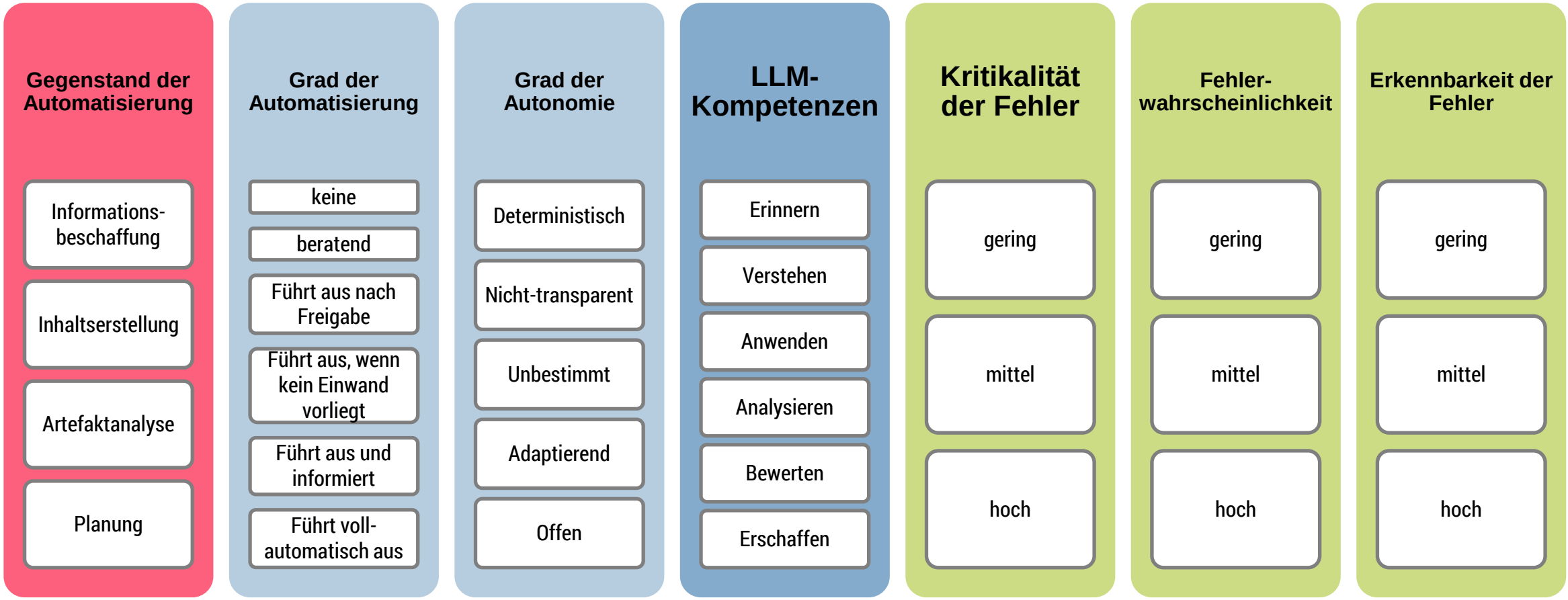
… und damit assoziierter Risiken

Automotive Quality Institute

Fraunhofer FOKUS

Risikofaktoren durch Technologieauswahl und -einsatz

Risikofaktoren aus dem Projektkontext



Die Eingabemaske Schritte 7 – 10

Charakterisierung der Risikofaktoren aufgrund Technologieauswahl & -einsatz

	EINGABE	
Name der LLM-Anwendung	**<eintragen>**	1
Beschreibung der LLM-Anwendung	<eintragen>	2
Projektkontext	<eintragen>	3
Beschreibung des Einsatzwecks	<eintragen>	4
SE-Aufgabe	Code example recommendation	5
SE-Kategorie	*Informationsbeschaffung*	6
Grad der Automatisierung	Keine	7
Grad der Autonomie	Offen	8
Eingabe	Natürliche Sprache	9
Ausgabe	Natürliche Sprache	10
Kritikalität Fehler	Unbekannt	11
Fehlerwahrscheinlichkeit	Unbekannt	12
Erkennbarkeit der Fehler	Unbekannt	13
Benötigte LLM-Kompetenz(en)	Erinnern	14

Schritte 7 – 10 (Dropdown-Auswahlliste)

7. Auswählen des Automatisierungsgrades aus der Taxonomie
8. Auswählen des Autonomiegrades aus der Taxonomie
9. Charakterisierung der Eingabe für die LLM-Anwendung (bspw. ob Freitext oder Programmcode der LLM-Anwendung als Eingabedaten dienen)
10. Charakterisierung der Ausgabe der LLM-Anwendung, d. h., in welchem Format die LLM-Anwendung antwortet

Die Eingabemaske Schritte 11 – 13 + 14

Charakterisierung der Risikofaktoren aufgrund des Projektkontexts + Charakterisierung der benötigten Fähigkeit der LLM-Anwendung

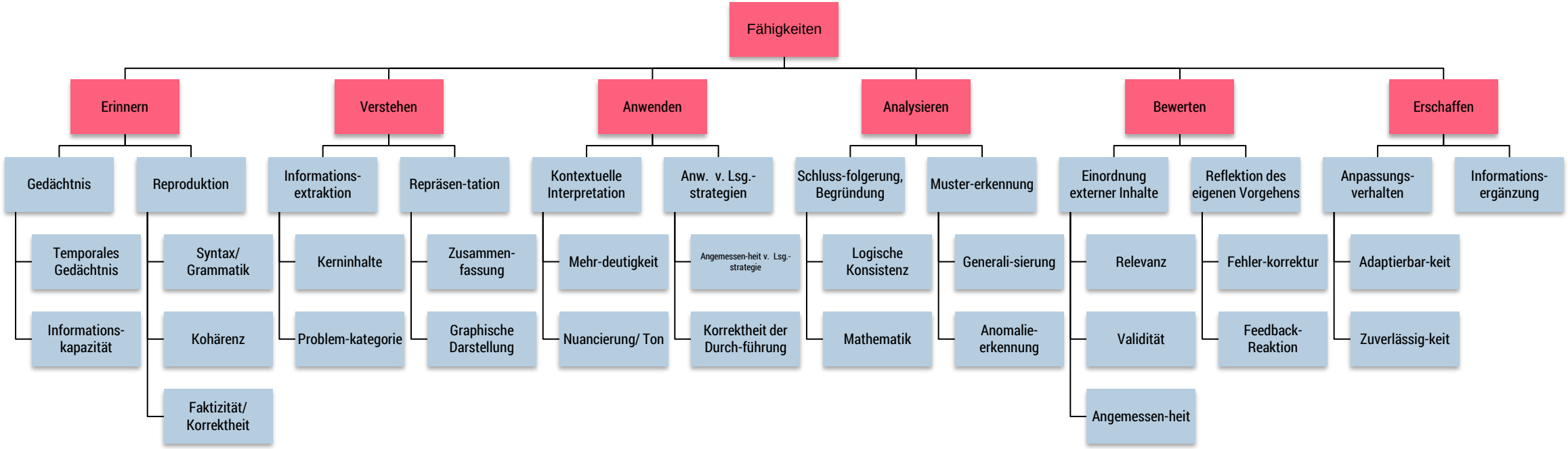
	EINGABE	
Name der LLM-Anwendung	**<eintragen>**	1
Beschreibung der LLM-Anwendung	<eintragen>	2
Projektkontext	<eintragen>	3
Beschreibung des Einsatzwecks	<eintragen>	4
SE-Aufgabe	Code example recommendation	5
SE-Kategorie	*Informationsbeschaffung*	6
Grad der Automatisierung	Keine	7
Grad der Autonomie	Offen	8
Eingabe	Natürliche Sprache	9
Ausgabe	Natürliche Sprache	10
Kritikalität Fehler	Unbekannt	11
Fehlerwahrscheinlichkeit	Unbekannt	12
Erkennbarkeit der Fehler	Unbekannt	13
Benötigte LLM-Kompetenz(en)	Erinnern	14

Schritte 11 – 13 + 14 (Dropdown-Auswahlliste)

11. Einschätzung der Kritikalität der durch Automatisierung induzierten Fehler
12. Einschätzung der Wahrscheinlichkeit für durch die Automatisierung induzierten Fehler
13. Einschätzung der Erkennbarkeit der durch die Automatisierung induzierten Fehler
14. Charakterisierung der benötigten Fähigkeit der LLM-Anwendung: Auswahl der Kategorie der LLM-spezifischen Kompetenzen gemäß der überarbeiteten Bloom-Hierarchie aus der Taxonomie

Bloom-Hierarchie

Taxonomie der LLM-spezifischen Kompetenzen eingeordnet in die Bloom-Hierarchie



Detaillierungsebenen der Prozessbeschreibungen

Schritt 02: Die Eingabemaske

AQI Automotive Quality Institute

Fraunhofer FOKUS

Schritt 01

- 01.1 Risikokarte kopieren
- 01.2 Blattschutz aufheben
- 01.3 Neue Risikokarte umbenennen

Schritt 02

- 02.1 Namen vergeben
- 02.2 Informative Felder ausfüllen, um den Anwendungskontext zu beschreiben
- 02.3 Alle weiteren Werte entsprechend der Taxonomie auswählen

Schritt 03

- 03.1 Ausgabe Teil 1: Einschätzungen bzgl. des Projektkontextes
- 03.2 Ausgabe Teil 2: Einschätzungen bzgl. benötigter LLM-Kompetenzen

Die Ausgabenfelder

Übersicht

AUSGABE

Verantwortlichkeit	Entwickler arbeitet eigenständig, keine besonderen Vorkehrungen nötig
Qualitätsüberprüfende Maßnahmen	strukturbasierte initiale Testung notwendig, falls möglich
Risiko-Index	7
Risiko-Einschätzung	hoch: präventive und risiko-mitigierende Maßnahmen müssen getroffen werden, mehrfache Testung in kritischen Bereichen notwendig

Anwendungsspezifische Risiken	Fehlinformationen, Irrelevante Informationen, unvollständige Informationen, Überholte Informationen, Nutzung von nicht-vertrauenswürdigen

Ausgabe, Teil 1
Risiken und Einschätzungen bzgl. des Projektkontextes

Ausgabe, Teil 2
Risiken und Einschätzungen bzgl. der benötigten LLM-Kompetenzen

Kompetenz-spezifische Risiken	Identifikation (allgemein)	Mitigation (allgemein)	Prompt-Engineering
Gedächtnisfähigkeit	• ergibt sich häufig bereits aus der Architektur des LLM	• Nutzung alternativer Architekturen	• Restrukturierung von Prompts, um die räumliche Nähe miteinander zusammenhängender Informationen zu gewährleisten
Korrektheit	• z.B. durch automatische	• Nutzung von Quellennachverfolgung	

Anhand der Eingabe wird eine spezifische Auswahl der zuvor diskutierten Risiken und möglichen Maßnahmen für eine Qualitätssicherung angezeigt.

Die Ausgabenfelder: Teil 1

Einschätzungen bezüglich des Projektkontextes
… Risiken zur Charakterisierung der Automatisierung

AUSGABE

Verantwortlichkeit	Verantwortung des Monitorings beim Entwickler, anfragenbasierte Überprüfung
Qualitätsüberprüfende Maßnahmen	regelmäßige Testung notwendig
Risiko-Index	6
Risiko-Einschätzung	mittel: risikominimierende Maßnahmen müssen in kritischen Bereichen nachgewiesen werden

Anwendungsspezifische Risiken	Halluzinationen, Schwachstellen, Teillösungen, Inkonsistente Lösungen, Nicht-Determinismus

Anwendungsspezifische Risiken

Mögliche Risiken bzgl. des Gegenstands der Automatisierung („SE-Kategorie“), abgeleitet aus der gewählten SE-Aufgabe

Verantwortlichkeit

Einschätzungen bzgl. des gewählten Grads der Automatisierung

Qualitätsprüfende Maßnahmen

Einschätzungen bzgl. des gewählten Grads der Autonomie

Risiko-Index

Wert ermittelt aus den angegebenen Werten für die Kritikalität, die Wahrscheinlichkeit und die Erkennbarkeit von Fehlern

Risiko-Index & Risiko-Einschätzung

- 1 bis 3: „gering“: Standard-Testung notwendig
- 4 bis 6: „mittel“: risikominimierende Maßnahmen nachweisen
- 7 bis 10: „hoch“: präventive / risiko-mitigierende Maßnahmen treffen

Die Ausgabenfelder: Teil 2

Einschätzungen bezüglich der benötigten LLM-Kompetenzen … Risiken und Mitigationsmöglichkeiten

Kompetenz-spezifische Risiken	Identifikation (allgemein)	Mitigation (allgemein)	Prompt-Engineering	Identifikation (codespezifisch)	Mitigation (codespezifisch)
Gedächtnisfähigkeit	• ergibt sich häufig bereits aus der Architektur des LLM	• Nutzung alternativer Architekturen	• Restrukturierung von Prompts, um die räumliche Nähe miteinander zusammenhängender Informationen zu gewährleisten		
Korrektheit	• z.B. durch automatische Evidenzgewinnung und Evaluation auf Benchmarks; ausführliche Übersicht zu Halluzinationen in (Huang et al. 2023)	• Nutzung von Quellennachverfolgung (siehe Urheberrecht) • z.B. durch gezielte Datenerweiterung, Modelleditierung, automatisierte Prüfung der logischen und kontextuellen Konsistenz; ausführliche Übersicht zu Halluzinationen in (Huang et al. 2023)		• Metrik für Syntax- & Semantiktests: Pass@k (Chen et al. 2021) auf Datensätzen wie HumanEval oder MBPP; CodeBLEU (Ren et al. 2020). Quellen: https://deepgram.com/learn/humaneval-llm-benchmark, https://github.com/google-research/google-research/tree/master/mbpp • Fehleridentifikation, durch menschlichen kognitiven Bias inspiriert (Jones und Steinhardt 2022)	• Frameworks zur Selbstkorrektur, bspw. CRITIC (Gou et al. 2024) • Syntaktische oder semantische Korrektheit kann teilweise durch Eigenschaften der Sprache gegeben sein, zum Beispiel beim Einsatz von Codeblöcken wie bei der von Microsoft entwickelten Programmiersprache ODSL für Office-Produkte (Gandhi et al. 2023)
Haftung	Während sich die Rechtsprechung im Kontext von LLMs noch in der Entwicklung befindet, haftet nach aktueller Rechtslage der LLM-Nutzer für mithilfe von LLM produzierte Inhalte (Einschätzung einer Kanzlei). In Bezug auf Datenschutz ist noch unklar, inwieweit OpenAI sich der Haftung entziehen kann, da Programme wie Chat-GPT keine DSGVO-Konformität einhalten können. Quelle: https://www.kanzlei.law/medien-und-wirtschaftsrecht/wenn-ki-auf-recht-trifft-die-rechtlichen-herausforderungen-der-chatgpt-nutzung/	—			
Informationssicherheit Eingabe	• Prüfung: Wo landen bei Prompts eingegebene Daten?	• Nutzung von firmeneigenen Instanzen auf internen Servern			
Reproduzierbarkeit	• Vielfachtestung	• Einfrieren von LLMs			
Robustheit	• Adversarial Attacks (ähnlich zur Verifikation herkömmlicher neuronaler Netze)	—		• Statische Prüfung, beispielsweise auf ASTs	—
Quellenangaben	• Evaluation der automatischen Attribution durch AttrEval-Simulation und AttrEval-GenSearch (Yue et al. 2024)	• manche LLMs ermöglichen eine Bereitstellung von Quellenangaben, bspw. Elicit und Consensus (beschränkt auf wissenschaftliche Paper) • Ansatz zu korrekten Quellenangaben bereits beim Pretraining: Source-Aware-Training (Khalifa et al. 2024)			

Kompetenz-spezifische Risiken

Ermittelte LLM-Fähigkeit mit erhöhtem Risiko

Identifikation (allgemein)

Relevante Aspekte und Einschätzungen

Mitigation (allgemein)

Mögliche risiko-mitgierende Maßnahmen

Prompt-Engineering

Mögliche risiko-mitgierende Maßnahmen bzgl. Anfragen und Kontext

< optional > Identifikation (codespezifisch)

Relevante Aspekte und Einschätzungen bzgl. Code

Prompt-Engineering (codespezifisch)

Mögliche risiko-mitgierende Maßnahmen bzgl. Code

AQI Automotive Quality Institute

Fraunhofer FOKUS

Beispiel „Code Refactoring“

Beispiel „Code Refactoring“

1. Ausfüllen der informativen Felder Name, Beschreibung, Projektkontext und Beschreibung des Einsatzzwecks.

		EINGABE
	Name der LLM-Anwendung	**GitHub Copilot (code refactoring)**
	Beschreibung der LLM-Anwendung	Die Anwendung analysiert den Code und macht auf konkrete Anforderung hin Verbesserungsvorschläge.
Gegenstand der Automatisierung	**Projektkontext** **Beschreibung des Einsatzwecks**	Entwicklung Seriensteuergerät Einsatz in der C/C++-Kodierung eines Seriensteuergeräts für die Richtungsanzeige im Fahrzeug.
	SE-Aufgabe	
	SE-Kategorie	*#NV*
Technologie-auswahl und -einsatz	**Grad der Automatisierung** **Grad der Autonomie**	
	Eingabe **Ausgabe**	
Projektkontext	**Kritikalität Fehler** **Fehlerwahrscheinlichkeit** **Erkennbarkeit der Fehler**	
LLM	**Benötigte LLM-Kompetenz(en)**	

Diese Felder dienen dazu, den Kontext für die folgenden Angaben zu dokumentieren. Die Eingaben beeinflussen die vom Programm generierten Risikoeinschätzungen nicht.

Beispiel: Wir setzen GitHub-Copilot für das Refactoring von C/C++ Code bei der Programmierung der Software für ein Seriensteuergerät ein.

Beispiel „Code Refactoring“

2. Zuordnung der Tätigkeit zu einer vordefinierten SE-Aufgaben und Ableitung der SE-Kategorie

		EINGABE
	Name der LLM-Anwendung	**GitHub Copilot (code refactoring)**
	Beschreibung der LLM-Anwendung	Die Anwendung analysiert den Code und macht auf konkrete Anforderung hin Verbesserungsvorschläge.
Gegenstand der Automatisierung	**Projektkontext** Beschreibung des Einsatzwecks	Entwicklung Seriensteuergerät Einsatz in der C/C++-Kodierung eines Seriensteuergeräts für die Richtungsanzeige im Fahrzeug.
	SE-Aufgabe	Code refactoring
	SE-Kategorie	*Inhaltserstellung*
Technologie-auswahl und -einsatz	**Grad der Automatisierung** **Grad der Autonomie**	
	Eingabe **Ausgabe**	
Projektkontext	**Kritikalität Fehler**	
Anwendungsspezifische Risiken	Halluzinationen, Schwachstellen, Teillösungen, inkonsistente Lösungen, Nicht-Determinismus	

Die geplante Tätigkeit wird einer vordefinierten SE-Aufgabe und damit automatisch einer SE-Kategorie zugeordnet. Die SE-Aufgaben werden als Liste in einem Dropdown-Menü zur Verfügung gestellt.

Beispiel: Wir suchen als SE-Aufgabe „Code Refactoring“ aus. Die SE-Kategorie „Inhaltserstellung“ wird automatisch zugeordnet. Auf Basis der Zuordnung werden „Anwendungsspezifische Risiken“ dargestellt.

Beispiel „Code Refactoring“

3. Festlegung von „Grad der Automatisierung“ und „Grad der Autonomie“

		EINGABE
	Name der LLM-Anwendung	**GitHub Copilot (code refactoring)**
	Beschreibung der LLM-Anwendung	Die Anwendung analysiert den Code und macht auf konkrete Anforderung hin Verbesserungsvorschläge.
Gegenstand der Automatisierung	**Projektkontext** Beschreibung des Einsatzwecks	Entwicklung Seriensteuergerät Einsatz in der C/C++-Kodierung eines Seriensteuergeräts für die Richtungsanzeige im Fahrzeug.
	SE-Aufgabe	Code refactoring
	SE-Kategorie	*Inhaltserstellung*
Technologie-auswahl und -einsatz	**Grad der Automatisierung**	Führt aus nach Freigabe
	Grad der Autonomie	Adaptierend
	Eingabe	
	Verantwortlichkeit	Verantwortung der Genehmigung beim Entwickler, risikobewusste Inhaltsüberprüfung notwendig
	Qualitätsüberprüfende Maßnahmen	regelmäßige Testung notwendig

Die geplante Aufgabe wird hinsichtlich ihres Grades der Automatisierung und ihres Grades der Autonomie eingeordnet. Beide Angaben erfolgen über ein Dropdown-Menü.

Beispiel: Wir gehen davon aus, dass der Copilot so konfiguriert ist, dass Änderungen „adaptierend“ vorgenommen werden, aber der Entwickler bei Änderungen um Freigabe gebeten wird. Als Ergebnis erhalten wir „Verantwortlichkeit“ und „Qualitätsüberprüfende Maßnahmen“.

Beispiel „Code Refactoring“

Festlegung der benötigten Kompetenz der KI sowie der Eingabe- und Ausgabeformate in der Interaktion mit der KI

Kompetenz-spezifische Risiken	Identifikation (allgemein)	Mitigation (allgemein)	Prompt-Engineering	Identifikation (codespezifisch)	Mitigation (codespe
Korrektheit	• z.B. durch automatische Evidenzgewinnung und Evaluation auf Benchmarks; ausführliche Übersicht zu Halluzinationen in (Huang, Yu et al. 2023)	• Nutzung von Quellennachverfolgung (siehe Urheberrecht) • z.B. durch gezielte Datenerweiterung, Modelleditierung, automatisierte Prüfung der logischen und kontextuellen Konsistenz; ausführliche Übersicht zu Halluzinationen in (Huang, Yu et al. 2023)		• Metrik für Syntax- & Semantiktests: Pass@k (Chen, Tworek et al. 2021) auf Datensätzen wie HumanEval oder MBPP; CodeBLEU (Ren et al. 2020). Quellen: https://deepgram.com/learn/humaneval-llm-benchmark, https://github.com/google-research/google-research/tree/master/mbpp • Fehleridentifikation, durch menschlichen kognitiven Bias inspiriert (Jones & Steinhardt 2022)	• Frameworks zur Se bspw. CRITIC (Gou e • Syntaktische oder s Korrektheit kann teil Eigenschaften der Sp gegeben sein, zum B Einsatz von Codeblö der von Microsoft en Programmiersprache Office-Produkte (Gar 2023)

Gegenstand der Automatisierung	**Projektkontext** Beschreibung des Einsatzwecks	Entwicklung Seriensteuergerät Einsatz in der C/C++-Kodierung eines Seriensteuergeräts für die Richtungsanzeige im Fahrzeug.
	SE-Aufgabe	Code refactoring
	SE-Kategorie	*Inhaltserstellung*
Technologie-auswahl und -einsatz	**Grad der Automatisierung**	Führt aus nach Freigabe
	Grad der Autonomie	Adaptierend
	Eingabe	Code + natürliche Sprache
	Ausgabe	Programmcode
Projektkontext	**Kritikalität Fehler** **Fehlerwahrscheinlichkeit** **Erkennbarkeit der Fehler**	
LLM	**Benötigte LLM-Kompetenz(en)**	Analysieren

Für die geplante Aufgabe wird festgelegt, welche LLM-Kompetenz benötigt wird und über welche Formate mit der KI kommuniziert wird..

Beispiel: Für das Code Refactoring nehmen wir an, dass der Copilot den bestehenden Code „Analysieren“ muss und Eingaben in Form von “Code und natürlicher Sprache“ erhält. In natürlicher Sprache wird das Ziel des Refactorings beschrieben, während der Code den Ausgangscode darstellt, der verbessert werden soll. Als Ausgabe liefert der Copilot den Programmcode zurück. Im Ergebnis liefert die Tabelle kompetenzspezifische Risiken aus. Da Programmcode involviert ist, wird die Tabelle durch „codespezifische“ Anmerkungen ergänzt.

Beispiel „Code Refactoring“

Risikoeinschätzung

		EINGABE
	Name der LLM-Anwendung	**GitHub Copilot (code refactoring)**
	Beschreibung der LLM-Anwendung	Die Anwendung analysiert den Code und macht auf konkrete Anforderung hin Verbesserungsvorschläge.
Gegenstand der Automatisierung	**Projektkontext** Beschreibung des Einsatzwecks	Entwicklung Seriensteuergerät Einsatz in der C/C++-Kodierung eines Seriensteuergeräts für die Richtungsanzeige im Fahrzeug
Risiko-Index		8
Risiko-Einschätzung		hoch: präventive und risiko-mitigierende Maßnahmen müssen getroffen werden, mehrfache Testung in kritischen Bereichen notwendig
	Ausgabe	Programmcode
Projektkontext	**Kritikalität Fehler**	Hoch
	Fehlerwahrscheinlichkeit	Mittel
	Erkennbarkeit der Fehler	Schwer
LLM	**Benötigte LLM-Kompetenz(en)**	Analysieren

Für die geplante Aufgabe wird festgelegt, welche Risiken vorliegen. Dies erfolgt über die Einschätzung der Kritikalität, Wahrscheinlichkeit sowie Erkennbarkeit der Fehler.
Für alle Bereiche lassen sich qualitative Risikoeinschätzungen angeben.

Beispiel: Für das Code Refactoring gehen wir von einer hohen Kritikalität der Fehler aus, da es sich um sicherheitskritische Software handelt.
Darüber hinaus schätzen wir die Wahrscheinlichkeit der Fehler auf „Mittel“ ein (wir haben bisher wenig Erfahrungen gesammelt) und ihre Erkennbarkeit auf „Gering“.
Als Ergebnis erhalten wir einen aggregierten Risiko-Index von 8.