AQI Automotive Quality Institute

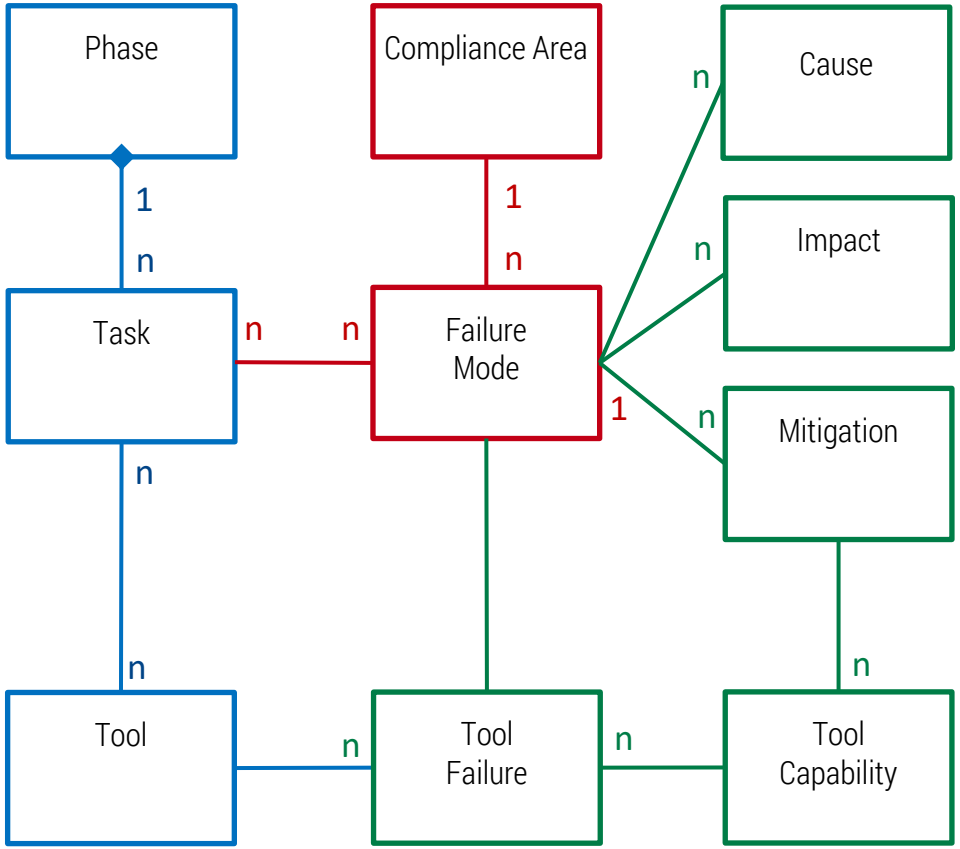
Q in KI: Anforderungen an Tools und Prozesse zur Absicherung

Anleitung zur Qualifizierung der Werkzeuge

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

Vorgehen im Projekt und verwendete Terminologie

Vom KI-Entwicklungszyklus zur Werkzeugqualifizierung



- Systematische Aufstellung der **Phasen (Phase)** und **Aufgaben (Task)** im KI-Entwicklungszyklus sowie beispielhafter **Werkzeuge (Tool).**
- Identifikation der **Abweichungen / Fehlerzustände (Failure Mode)** entlang der **Aufgaben** für jedes **Konformitätsgebiet (Compliance Area).**
- Identifikation von **Ursachen (Cause), Wirkungen (Impact)** und möglicher **Gegenmaßnahmen (Mitigation)** für jede **Abweichung (Failure Mode).**
- Identifikation von spezifischen **nicht-konformem Verhalten von Werkzeugen (Tool Failure)** und benötigter **Werkzeugfähigkeiten (ToolCapability)** für eine **Abweichung (Failure Mode).**

Phase (Phase): Phase aus dem KI-Entwicklungszyklus

Task (Aufgabe): Aufgabe während des KI-Entwicklungszyklus, die von Werkzeugen unterstützt wird.

Tool (Werkzeug): Werkzeug, dass eine Aufgabe unterstützt/ausführt.

Compliance Area (Konformitätsgebiet): Konformitätsgebiet (funktionale Sicherheit, Datenschutz, KI-Regulierung).

Failure Mode (Fehlerzustand): Abweichung, die zu Nichtkonformität führt.

Tool Failure (Werkzeugfehler): Nicht-konformes Verhalten von Werkzeugen als Ursache für eine Abweichung

Cause (Ursache): Ursachen für das Zustandekommen der Abweichung.

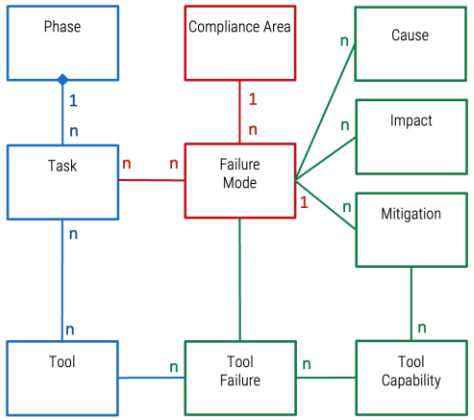
Impact (Auswirkung): Auswirkungen, die die Abweichung hat.

Mitigation (Minderung): Maßnahme zur Vermeidung der Abweichung.

Tool Capability (Werkzeugfähigkeiten): Benötigte Fähigkeiten eines Werkzeugs zur Vermeidung der Abweichung.

Systematische Ausarbeitung der Inhalte

Interaktive Excel-Tabelle (>80 Zeilen) als Demonstrator → Ergebnisse operabel machen



Inhalte der Excel-Tabelle wurden gemäß verwendeter Terminologie erarbeitet

Compliance Area	Phase	Task	Failure Mode	Impact	Severity	Causes	Mitigation	Occurrence	Detectability	RPN	Potential Tool Failure	Recommended Tool Capability
Functional Safety	Training & Validation	Deep Learning & ML Frameworks	Lack of Real-Time Processing	Trained models and supporting frameworks are unable to deliver low-latency, real-time inference performance required for critical applications such as autonomous systems.		Model architectures are too large or complex for target real-time environments. Frameworks do not leverage hardware acceleration efficiently (e.g., GPU, TPU, Edge devices). Pipeline overhead (e.g., preprocessing, postprocessing) adds excessive latency during inference.	Apply model optimization techniques such as pruning, quantization, or knowledge distillation. Ensure deployment-ready models are benchmarked and optimized for specific hardware accelerators. Optimize full pipeline latency by batching operations and reducing unnecessary transformations.				May fail to guarantee deterministic training execution and runtime validation, resulting in trained models that do not meet real-time safety-critical performance requirements.	Framework support for model optimization (e.g., TensorRT, ONNX Runtime optimizations). Support for hardware-specific runtime optimizations and deployment targeting (e.g., EdgeTPU, CUDA kernels). Framework support for pipeline fusion, minimal preprocessing APIs, and asynchronous inference.
Functional Safety	Training & Validation	Deep Learning & ML Frameworks	Inadequate Error Handling & Logging	Frameworks and models lack robust error detection, recovery, and structured logging capabilities, making root cause analysis and system recovery difficult.		Training and inference exceptions are not properly caught or logged. Losses, gradients, or other training dynamics are not systematically monitored for anomalies. Training and deployment pipelines lack comprehensive logging and metadata tracking.	Integrate structured exception handling and mandatory logging for all critical training and inference operations. Implement runtime monitors for numerical instabilities (e.g., NaNs, divergence) and alert on thresholds. Embed pipeline-wide metadata capture and systematic experiment logging into the training framework.				May fail to provide comprehensive error logging and runtime monitoring, resulting in trained models with limited traceability for failure analysis in safety-critical systems.	Framework support for structured error reporting, custom callbacks, and logging integration. Built-in hooks for anomaly detection during training and validation (gradient checkers, divergence detectors). ML lifecycle management tools (e.g., MLflow, Weights & Biases) integration for metadata and error tracking.
Functional Safety	Training & Validation	Deep Learning & ML Frameworks	Model Instability & Poor Generalization	Inconsistent training leading to unreliable AI behaviour in critical applications.		Lack of deterministic/reproducible training settings due to parallel computation and floating-point arithmetic. Limited built-in validation or monitoring. No built-in numerical stability mechanisms.	Implement deterministic training pipelines with seed control and precision configuration to ensure repeatability. Integrate continuous performance monitoring during training with automatic validation checkpoints. Use frameworks that incorporate gradient clipping, normalization, and regularization for improved numerical robustness.				May fail to enforce deterministic training procedures and runtime validation, resulting in trained models with unpredictable behavior and poor generalization in safety-critical applications.	Explicit deterministic training support with reproducibility settings and controlled parallelism. Built-in validation dashboards and metric tracking APIs. Numerical stability modules with configurable regularization options.

Risikobasierte Werkzeugqualifizierung (Details)

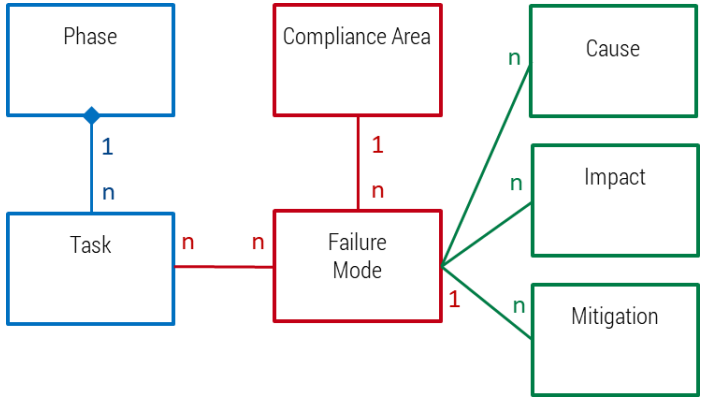
Zweistufiges Verfahren zur Ermittlung von aufgabenbezogenen Risiken und einer Qualifizierung von Werkzeugen entlang der Risiken

AQI | Automotive Quality Institute

1. Ermittlung von Risiken in den Entwicklungsaufgaben (nach FMEA)

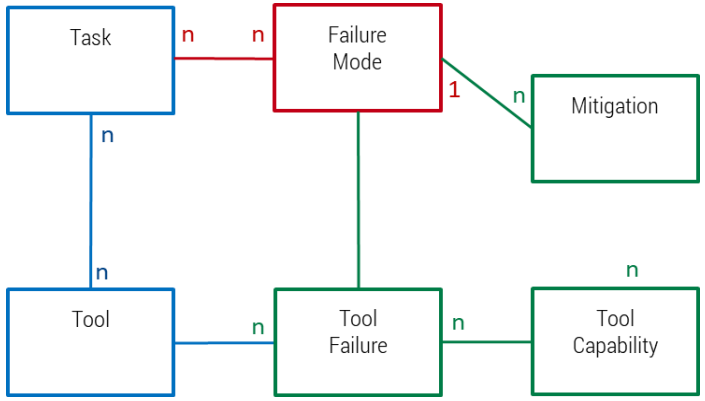
RPN = Severity * Occurrence * Detection

1.1 Auswahl der relevanten Aufgaben und Phasen	1.2 Auswahl der relevanten Konformitätsbereiche und Fehlermodi	1.3. Berechnung des RPN für ausgesuchte Fehlermodi in einer Aufgabe



2. Ermittlung des Qualifizierungsbedarfs für KI-Werkzeuge

2.1 Einschätzung über die Bedeutung des Werkzeugs für eine Abweichung (Tool Impact)	2.2 Bewertung der Erkennbarkeit des Werkzeugfehlers (Tool Error Detection) und Bestimmung des Tool Confidence Level (TCL) pro Konformitätsbereich	2.3 Abschließende Werkzeugbewertung und Handlungsempfehlung auf Basis der RPN und des TCL



Risikobasierte Werkzeugqualifizierung (Details)

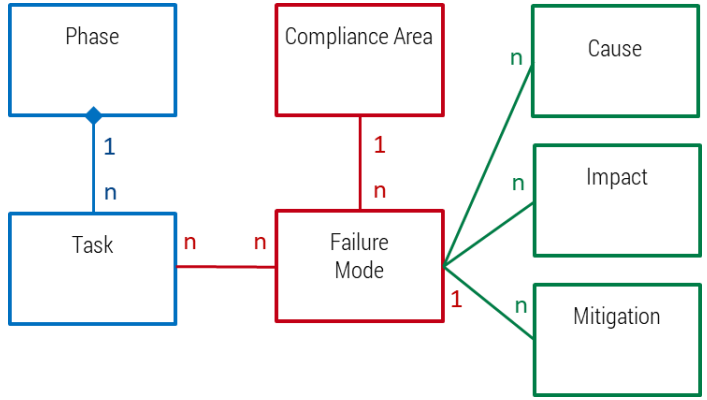
Zweistufiges Verfahren zur Ermittlung von aufgabenbezogenen Risiken und einer Qualifizierung von Werkzeugen entlang der Risiken

AQI | Automotive Quality Institute

1. Ermittlung von Risiken in den Entwicklungsaufgaben (nach FMEA)

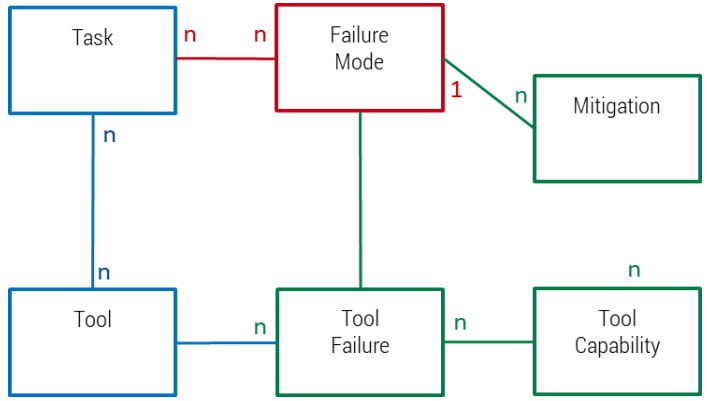
RPN = Severity * Occurrence * Detection

1.1 Auswahl der relevanten Aufgaben und Phasen	1.2 Auswahl der relevanten Konformitätsbereiche und Fehlermodi	1.3. Berechnung des RPN für ausgesuchte Fehlermodi in einer Aufgabe



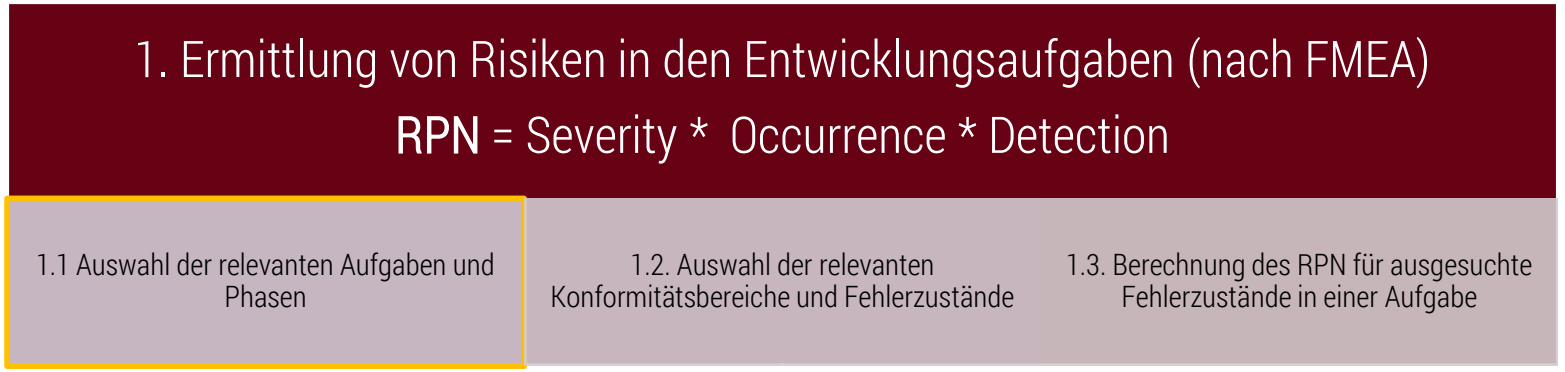
2. Ermittlung des Qualifizierungsbedarfs für KI-Werkzeuge

2.1 Einschätzung über die Bedeutung des Werkzeugs für eine Abweichung (Tool Impact)	2.2 Bewertung der Erkennbarkeit des Werkzeugfehlers (Tool Error Detection) und Bestimmung des Tool Confidence Level (TCL) pro Konformitätsbereich	2.3 Abschließende Werkzeugbewertung und Handlungsempfehlung auf Basis der RPN und des TCL



1. Ermittlung von Risiken in den Entwicklungsaufgaben

1.1 Auswahl der relevanten Aufgaben und Phasen



FMEA Excel: Auswahl der Aufgaben und Phasen

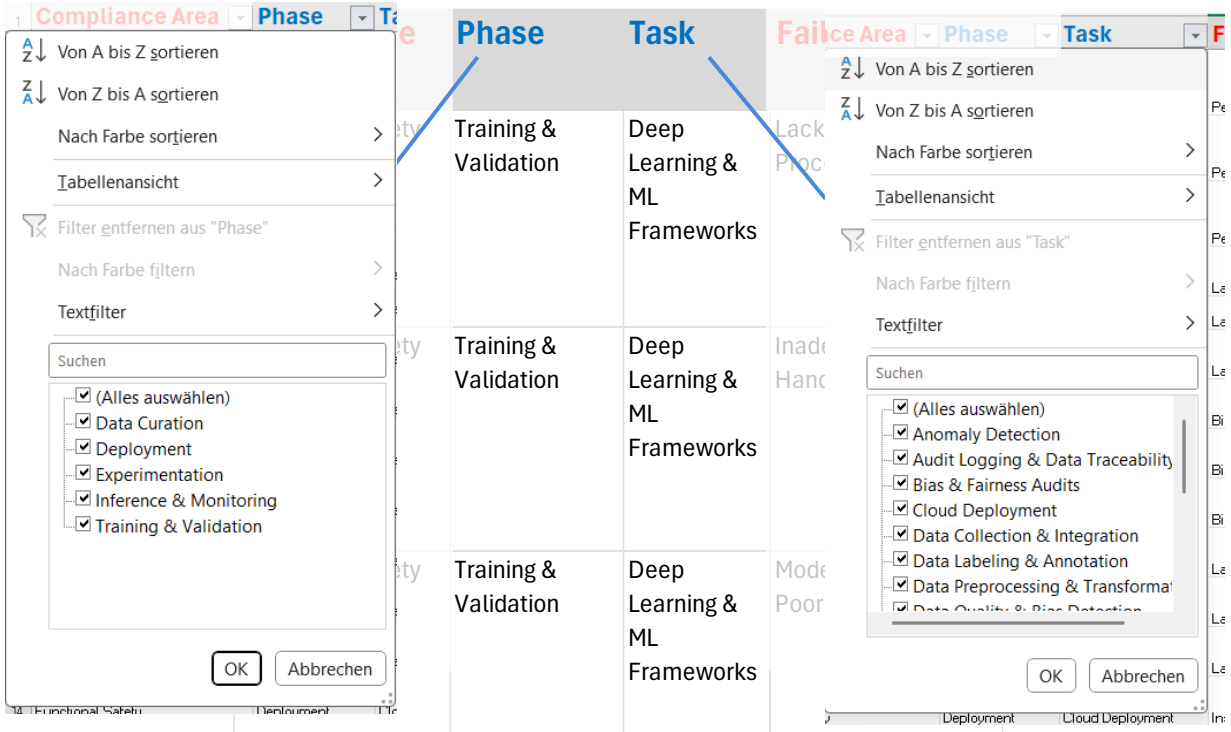
AQI | Automotive Quality Institute

Nicht jede ML-Aufgabe ist relevant

Ausgangslage: Ist diese ML-Aufgabe Teil des Entwicklungssystems?

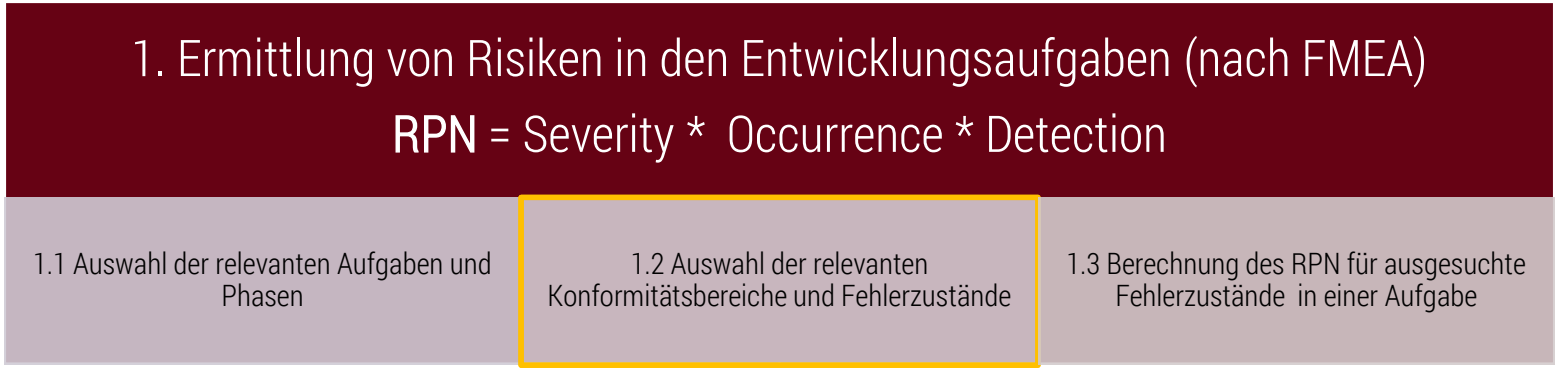
Auswahl	Relevante Phasen (Liste), relevante Tasks (Liste)
Skala	Boolean (Yes/No)
Zweck	Grenzt die Fehlermodi auf die gewünschten Phasen und Aufgaben ein.

Aktion:
Relevante ML-Aufgabe aus der Excel-Tabelle auswählen.



1. Ermittlung von Risiken in den Entwicklungsaufgaben

1.2 Auswahl der relevanten Konformitätsbereiche und Fehlerzustände



FMEA Excel: Auswahl der Fehlerzustände

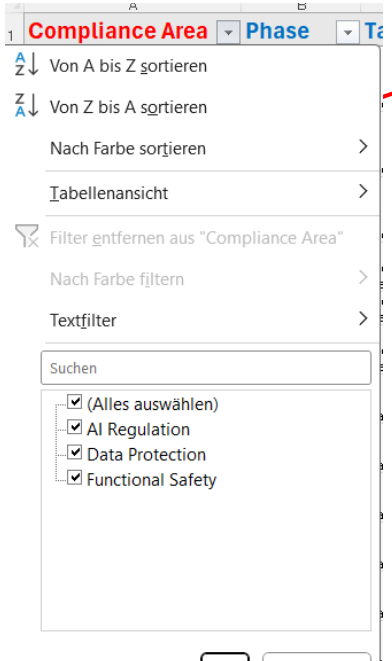
AQI | Automotive Quality Institute

Nicht jede ML-Aufgabe ist relevant

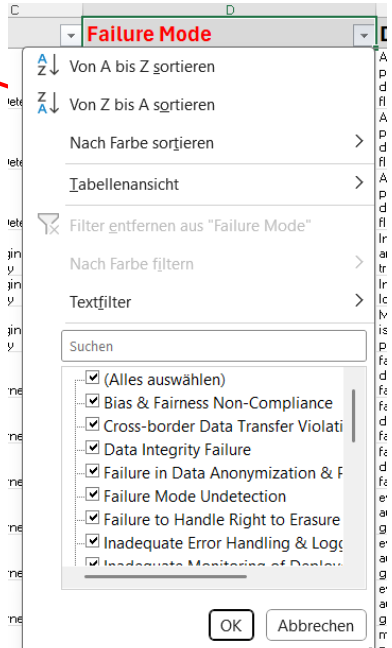
Ausgangslage: Ist diese ML-Aufgabe Teil des Entwicklungssystems?

Auswahl	Relevante Konformitätsbereiche (Liste), Fehlermodi (Liste)
Skala	Boolean (Yes/No)
Zweck	Grenzt die Fehlermodi auf die gewünschten Konformitätsbereiche ein.

Aktion:
Relevante Konformitäts-bereiche aus der Excel-Tabelle auswählen.

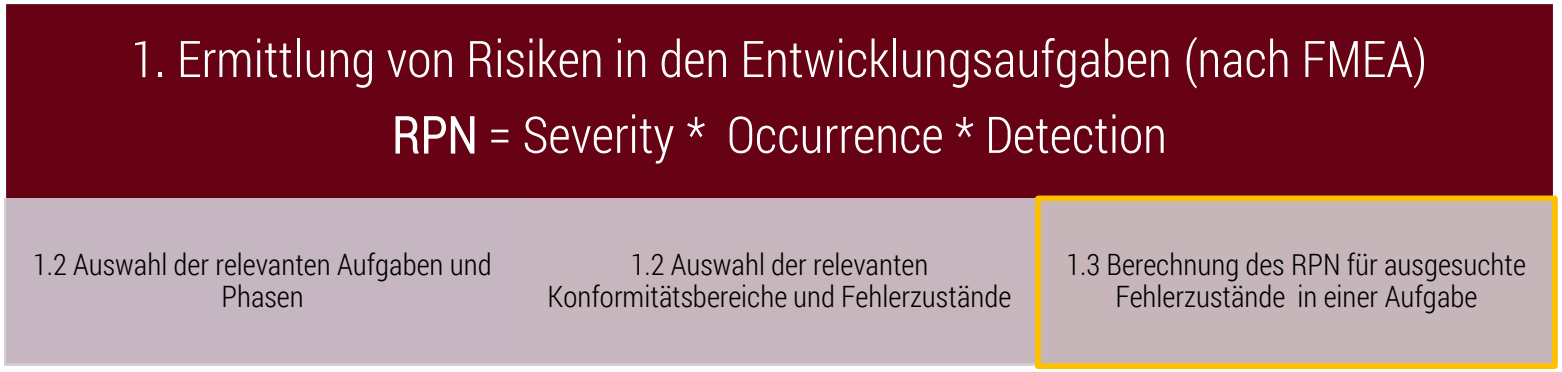


Compliance Area	Phase	Task	Failure Mode
Functional Safety	Training & Validation	Deep Learning & ML Frameworks	Lack of Real-Time Processing
Functional Safety	Training & Validation	Deep Learning & ML Frameworks	Inadequate Error Handling & Logging
Functional Safety	Training & Validation	Deep Learning & ML	Model Instability & Poor Generalization



1. Ermittlung von Risiken in den Entwicklungsaufgaben

1.3 Berechnung des RPN für ausgesuchte Fehlerzustände in einer Aufgabe



FMEA Excel: Bewertung der Fehlerzustände

Automotive Quality Institute

Risikoprioritätskennzahl

RPN = Severity * Occurrence * Detection

Aktion:
Berechnung der Risikoprioritätskennzahl **(RPN)** für jeden Fehlermodus.

Eingabe — Bewertung des Schweregrades — *#(Bewertung von 1 - 10)*

Eingabe — Bewertung der Auftretungswahrscheinlichkeit — *#(Bewertung von 1 - 10)*

Eingabe — Bewertung der Erkennbarkeit — *#(Bewertung von 1 - 10)*

Compliance Area	Phase	Task	Failure Mode	Impact	Severity	Causes	Mitigation	Occurrence	Detectability	RPN
Functional Safety	Training & Validation	Deep Learning & ML Frameworks	Lack of Real-Time Processing	Trained models and supporting frameworks are unable to deliver low-latency, real-time inference performance required for critical applications such as autonomous systems.		Model architectures are too large or complex for target real-time environments. Frameworks do not leverage hardware acceleration efficiently (e.g., GPU, TPU, Edge devices). Pipeline overhead (e.g., preprocessing, postprocessing) adds excessive latency during inference.	Apply model optimization techniques such as pruning, quantization, or knowledge distillation. Ensure deployment-ready models are benchmarked and optimized for specific hardware accelerators. Optimize full pipeline latency by batching operations and reducing unnecessary transformations.			
Functional Safety	Training & Validation	Deep Learning & ML Frameworks	Inadequate Error Handling & Logging	Frameworks and models lack robust error detection, recovery, and structured logging capabilities, making root cause analysis and system recovery difficult.		Training and inference exceptions are not properly caught or logged. Losses, gradients, or other training dynamics are not systematically monitored for anomalies. Training and deployment pipelines lack comprehensive logging and metadata tracking.	Integrate structured exception handling and mandatory logging for all critical training and inference operations. Implement runtime monitors for numerical instabilities (e.g., NaNs, divergence) and alert on thresholds. Embed pipeline-wide metadata capture and systematic experiment logging into the training framework.			
Functional Safety	Training & Validation	Deep Learning & ML Frameworks	Model Instability & Poor Generalization	Inconsistent training leading to unreliable AI behaviour in critical applications.		Lack of deterministic/reproducible training settings due to parallel computation and floating-point arithmetic. Limited built-in validation or monitoring. No built-in numerical stability mechanisms.	Implement deterministic training pipelines with seed control and precision configuration to ensure repeatability. Integrate continuous performance monitoring during training with automatic validation checkpoints. Use frameworks that incorporate gradient clipping, normalization, and regularization for improved numerical robustness.			

FMEA Excel: Bewertung der Fehlerzustände

AQI Automotive Quality Institute

Bedeutung des Schweregrades (Severity)

RPN = Severity * Occurrence * Detection

Eingabe	Bewertung des Schweregrades
Skala	1 – 3: 4 – 6; 7 - 10

Aktion:
Bewertung des Schweregrades für die Failure-Modes durchführen, die in der Excel-Tabelle den ausgewählten ML-Aufgaben zugeordnet sind.

Compliance Area	Phase	Task	Failure Mode	Impact	Severity
Functional Safety	Training & Validation	Deep Learning & ML Frameworks	Lack of Real-Time Processing	Trained models and supporting frameworks are unable to deliver low-latency, real-time inference performance required for critical applications such as autonomous systems.	2
Functional Safety	Training & Validation	Deep Learning & ML Frameworks	Inadequate Error Handling & Logging	Frameworks and models lack robust error detection, recovery, and structured logging capabilities, making root cause analysis and system recovery difficult.	6
Functional Safety	Training & Validation	Deep Learning & ML Frameworks	Model Instability & Poor Generalization	Inconsistent training leading to unreliable AI behaviour in critical applications.	3

FMEA Excel: Bewertung der Fehlerzustände

Auftrittswahrscheinlichkeit der Fehlerursache (Occurence)

RPN = Severity * Occurrence * Detection

Eingabe	Bewertung der Auftrittswahrscheinlichkeit
Skala	1 – 3: 4 – 6; 7 - 10

Aktion:
Bewertung der Auftrittswahrscheinlichkeit für die Failure-Modes durchführen, die in der Excel-Tabelle den ausgewählten ML-Aufgaben zugeordnet sind.

Causes	Mitigations	Occurrence	Detectability	RPN
Model architectures are too large or complex for target real-time environments. Frameworks do not leverage hardware acceleration efficiently (e.g., GPU, TPU, Edge devices). Pipeline overhead (e.g., preprocessing, postprocessing) adds excessive latency during inference.	Apply model optimization techniques such as pruning, quantization, or knowledge distillation. Ensure deployment-ready models are benchmarked and optimized for specific hardware accelerators. Optimize full pipeline latency by batching operations and reducing unnecessary transformations.	2		
Training and inference exceptions are not properly caught or logged. Losses, gradients, or other training dynamics are not systematically monitored for anomalies. Training and deployment pipelines lack comprehensive logging and metadata tracking.	Integrate structured exception handling and mandatory logging for all critical training and inference operations. Implement runtime monitors for numerical instabilities (e.g., NaNs, divergence) and alert on thresholds. Embed pipeline-wide metadata capture and systematic experiment logging into the training framework.	6		
Lack of deterministic/reproducible training settings due to parallel computation and floating-point arithmetic. Limited built-in validation or monitoring. No built-in numerical stability mechanisms.	Implement deterministic training pipelines with seed control and precision configuration to ensure repeatability. Integrate continuous performance monitoring during training with automatic validation checkpoints. Use frameworks that incorporate gradient clipping, normalization, and regularization for improved numerical robustness.	3		

FMEA Excel: Bewertung der Fehlerzustände

Automotive Quality Institute

Entdeckungswahrscheinlichkeit des Fehlers oder seiner Ursache (Detectability)

RPN = Severity * Occurrence * Detection

Eingabe	Bewertung der Erkennbarkeit
Skala	1 – 3: 4 – 6; 7 - 10

Aktion:
Bewertung der Erkennbarkeit für die Failure-Modes durchführen, die in der Excel-Tabelle den ausgewählten ML-Aufgaben zugeordnet sind.

Causes	Mitigations	Occurrence	Detectability	RPN
Model architectures are too large or complex for target real-time environments. Frameworks do not leverage hardware acceleration efficiently (e.g., GPU, TPU, Edge devices). Pipeline overhead (e.g., preprocessing, postprocessing) adds excessive latency during inference.	Apply model optimization techniques such as pruning, quantization, or knowledge distillation. Ensure deployment-ready models are benchmarked and optimized for specific hardware accelerators. Optimize full pipeline latency by batching operations and reducing unnecessary transformations.	2	5	
Training and inference exceptions are not properly caught or logged. Losses, gradients, or other training dynamics are not systematically monitored for anomalies. Training and deployment pipelines lack comprehensive logging and metadata tracking.	Integrate structured exception handling and mandatory logging for all critical training and inference operations. Implement runtime monitors for numerical instabilities (e.g., NaNs, divergence) and alert on thresholds. Embed pipeline-wide metadata capture and systematic experiment logging into the training framework.	6	6	
Lack of deterministic/reproducible training settings due to parallel computation and floating-point arithmetic. Limited built-in validation or monitoring. No built-in numerical stability mechanisms.	Implement deterministic training pipelines with seed control and precision configuration to ensure repeatability. Integrate continuous performance monitoring during training with automatic validation checkpoints. Use frameworks that incorporate gradient clipping, normalization, and regularization for improved numerical robustness.	3	3	

FMEA Excel: Bewertung der Fehlerzustände

AQI | Automotive Quality Institute

Risikoprioritätskennzahl

RPN = Severity * Occurrence * Detection

Aktion:
Berechnung des Risikoprioritätskennzahl **(RPN)** für jeden Fehlermodus.

Skala 1 – 29; 30 – 299; 300 -1000

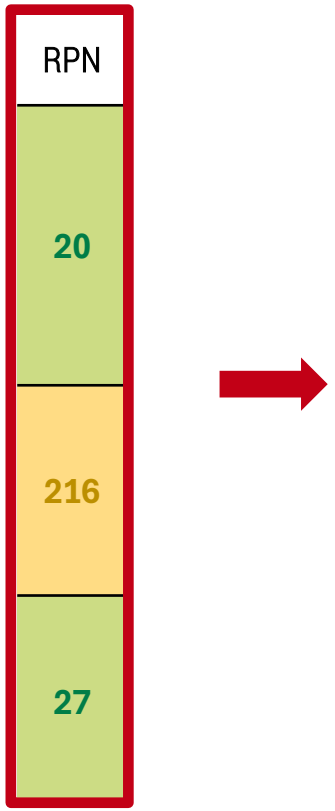
Compliance Area	Phase	Task	Failure Mode	Impact	Severity	Causes	Mitigations	Occurrence	Detectability	RPN
Functional Safety	Training & Validation	Deep Learning & ML Frameworks	Lack of Real-Time Processing	Trained models and supporting frameworks are unable to deliver low-latency, real-time inference performance required for critical applications such as autonomous systems.	2	Model architectures are too large or complex for target real-time environments. Frameworks do not leverage hardware acceleration efficiently (e.g., GPU, TPU, Edge devices). Pipeline overhead (e.g., preprocessing, postprocessing) adds excessive latency during inference.	Apply model optimization techniques such as pruning, quantization, or knowledge distillation. Ensure deployment-ready models are benchmarked and optimized for specific hardware accelerators. Optimize full pipeline latency by batching operations and reducing unnecessary transformations.	2	5	20
Functional Safety	Training & Validation	Deep Learning & ML Frameworks	Inadequate Error Handling & Logging	Frameworks and models lack robust error detection, recovery, and structured logging capabilities, making root cause analysis and system recovery difficult.	6	Training and inference exceptions are not properly caught or logged. Losses, gradients, or other training dynamics are not systematically monitored for anomalies. Training and deployment pipelines lack comprehensive logging and metadata tracking.	Integrate structured exception handling and mandatory logging for all critical training and inference operations. Implement runtime monitors for numerical instabilities (e.g., NaNs, divergence) and alert on thresholds. Embed pipeline-wide metadata capture and systematic experiment logging into the training framework.	6	6	216
Functional Safety	Training & Validation	Deep Learning & ML Frameworks	Model Instability & Poor Generalization	Inconsistent training leading to unreliable AI behaviour in critical applications.	3	Lack of deterministic/reproducible training settings due to parallel computation and floating-point arithmetic. Limited built-in validation or monitoring. No built-in numerical stability mechanisms.	Implement deterministic training pipelines with seed control and precision configuration to ensure repeatability. Integrate continuous performance monitoring during training with automatic validation checkpoints. Use frameworks that incorporate gradient clipping, normalization, and regularization for improved numerical robustness.	3	3	27

FMEA Excel: Bewertung der Fehlerzustände

AQI Automotive Quality Institute

Risikoprioritätskennzahl

RPN = Severity * Occurrence * Detection



Auswertung der Risikoprioritätskennzahl (RPN):

- Niedrige RPN-Werte (RPN 1–30) weisen auf eine geringe Kritikalität hin und erfordern keine Maßnahmen.
- Mittlere RPN-Werte (RPN 31–299) weisen auf eine mittlere Kritikalität hin und erfordern daher Maßnahmen.
- Hohe Werte (RPN 300–1000) weisen auf eine hohe Kritikalität hin und erfordern umfangreichere Maßnahmen.

→ Die Ermittlung von Risiken (RPN) in den Entwicklungsaufgaben (nach FMEA) fokussiert sich auf die Ermittlung **je Use-Case**

Risikobasierte Werkzeugqualifizierung (Details)

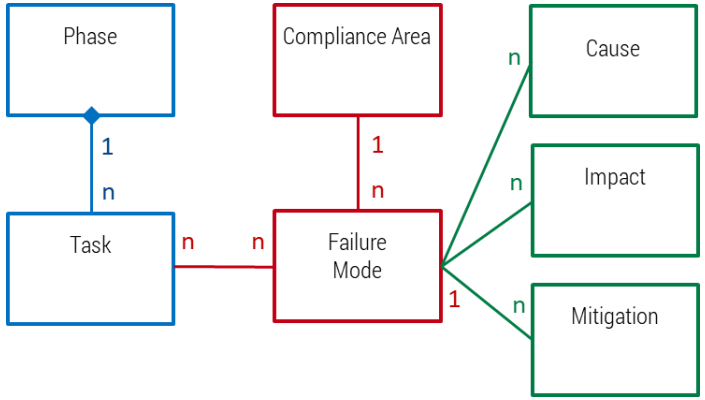
Zweistufiges Verfahren zur Ermittlung von aufgabenbezogenen Risiken und einer Qualifizierung von Werkzeugen entlang der Risiken

AQI Automotive Quality Institute

1. Ermittlung von Risiken in den Entwicklungsaufgaben (nach FMEA)

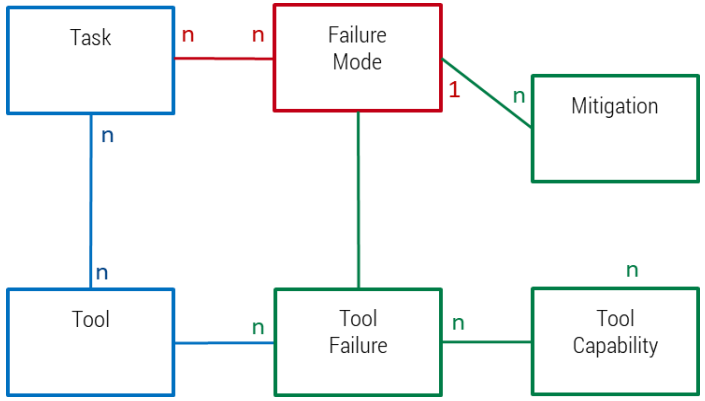
RPN = Severity * Occurrence * Detection

1.1 Auswahl der relevanten Aufgaben und Phasen	1.2 Auswahl der relevanten Konformitätsbereiche und Fehlermodi	1.3. Berechnung des RPN für ausgesuchte Fehlermodi in einer Aufgabe



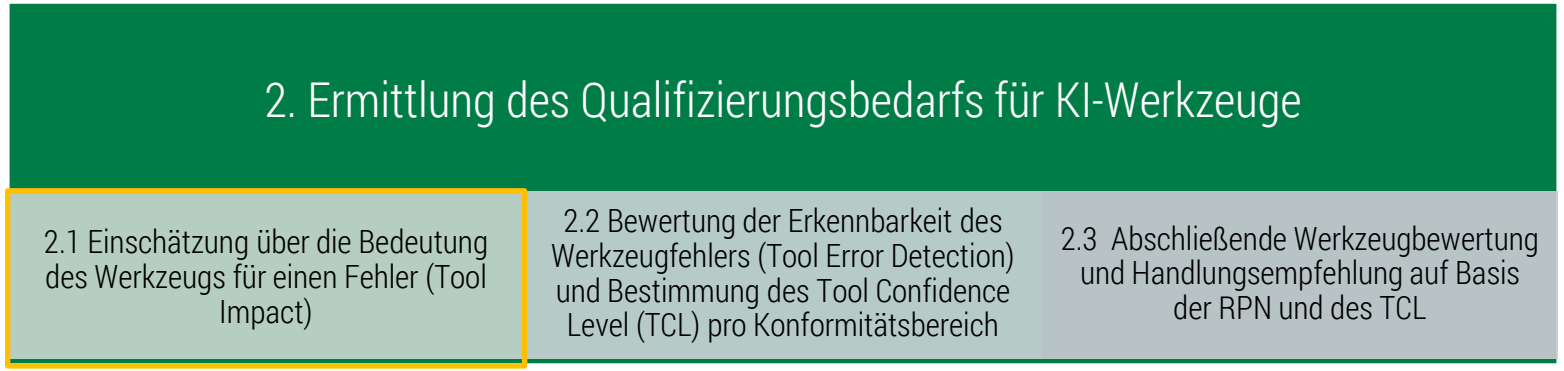
2. Ermittlung des Qualifizierungsbedarfs für KI-Werkzeuge

2.1 Einschätzung über die Bedeutung des Werkzeugs für eine Abweichung (Tool Impact)	2.2 Bewertung der Erkennbarkeit des Werkzeugfehlers (Tool Error Detection) und Bestimmung des Tool Confidence Level (TCL) pro Konformitätsbereich	2.3 Abschließende Werkzeugbewertung und Handlungsempfehlung auf Basis der RPN und des TCL



2. Ermittlung des Qualifizierungsbedarfs für KI- Werkzeuge

2.1 Einschätzung über die Bedeutung des Werkzeugs für einen Fehler (Tool Impact)



FMEA Excel: Auswirkung eines Werkzeugfehlers

AQI Automotive Quality Institute

Auswirkungsgrad des Werkzeuges

Ausgangslage: Hat ein potenzieller Fehler des Tools Einfluss auf die Funktionalität sicherheitsrelevanter Systeme?

Eingabe	Tool Impact Value - Einschätzung über das Vorhandensein von sicherheitsrelevanten Auswirkungen
Skala	1 (nein) , 2 (ja)

Potential Tool Failure	Recommended Tool Capabilities	Tool Impact Value	Tool Error Detection	Tool Confidence Value
May fail to guarantee deterministic training execution and runtime validation, resulting in trained models that do not meet real-time safety-critical performance requirements.	Framework support for model optimization (e.g., TensorRT, ONNX Runtime optimizations). Support for hardware-specific runtime optimizations and deployment targeting (e.g., EdgeTPU, CUDA kernels). Framework support for pipeline fusion, minimal preprocessing APIs, and asynchronous inference.	1		
May fail to provide comprehensive error logging and runtime monitoring, resulting in trained models with limited traceability for failure analysis in safety-critical systems.	Framework support for structured error reporting, custom callbacks, and logging integration. Built-in hooks for anomaly detection during training and validation (gradient checkers, divergence detectors). ML lifecycle management tools (e.g., MLflow, Weights & Biases) integration for metadata and error tracking.	2		
May fail to enforce deterministic training procedures and runtime validation, resulting in trained models with unpredictable behavior and poor generalization in safety-critical applications.	Explicit deterministic training support with reproducibility settings and controlled parallelism. Built-in validation dashboards and metric tracking APIs. Numerical stability modules with configurable regularization options.	2		

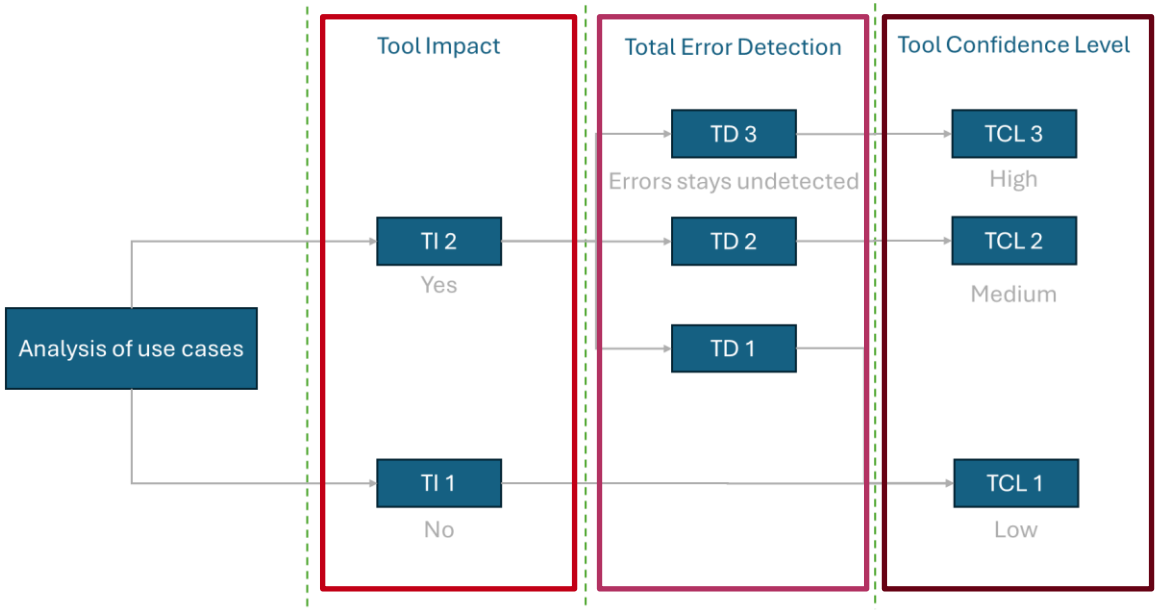
Hintergrund: Auswirkung eines Werkzeugfehlers

Automotive Quality Institute

Auswirkungsgrad des Werkzeuges

Die Abbildung wurde in Anlehnung an das Kapitel 11.4.5.2 der **ISO 26262:2018**, erarbeitet und stellt den Prozess zur Bestimmung des **Tool Confidence Levels (TCL)** dar, basierend auf:

- **Tool Impact (TI),** beurteilt, ob ein Fehler des Tools sicherheitsrelevant sein kann,
- **Tool Error Detection (TD) ,** beurteilt die Wahrscheinlichkeit, dass ein Fehler entdeckt wird, und resultiert im
- **Tool Confidence Level (TCL),** dem erforderlichen Maß an Vertrauen in das Tool (TCL1 bis TCL3).



Quelle: Eigene Darstellung nach ISO 26262:2018 Kap. 11.4.5.2

2. Ermittlung des Qualifizierungsbedarfs für KI- Werkzeuge

2.2 Bewertung der Erkennbarkeit des Werkzeugfehlers (Tool Error Detection) und TCL

FMEA Excel: Erkennbarkeit des Werkzeugfehlers

AQI Automotive Quality Institute

Erkennbarkeit von Werkzeugfehlern

Ausgangslage: Wie gut können Fehler, die durch das Tool entstehen, im weiteren Entwicklungsprozess erkannt werden?

Eingabe	Tool Error Detection - Einschätzung über das Vorhandensein von sicherheitsrelevanten Auswirkungen
Skala	Fehlerfindung (1 = wird erkannt, 2 = könnten unentdeckt bleiben, 3 = bleiben unentdeckt)

Potential Tool Failure	Testable Tool Capabilities	Tool Impact Value	Tool Error Detection	Tool Confidence Value
May fail to guarantee deterministic training execution and runtime validation, resulting in trained models that do not meet real-time safety-critical performance requirements.	Framework support for model optimization (e.g., TensorRT, ONNX Runtime optimizations). Support for hardware-specific runtime optimizations and deployment targeting (e.g., EdgeTPU, CUDA kernels). Framework support for pipeline fusion, minimal preprocessing APIs, and asynchronous inference.	1	1	
May fail to provide comprehensive error logging and runtime monitoring, resulting in trained models with limited traceability for failure analysis in safety-critical systems.	Framework support for structured error reporting, custom callbacks, and logging integration. Built-in hooks for anomaly detection during training and validation (gradient checkers, divergence detectors). ML lifecycle management tools (e.g., MLflow, Weights & Biases) integration for metadata and error tracking.	2	2	
May fail to enforce deterministic training procedures and runtime validation, resulting in trained models with unpredictable behavior and poor generalization in safety-critical applications.	Explicit deterministic training support with reproducibility settings and controlled parallelism. Built-in validation dashboards and metric tracking APIs. Numerical stability modules with configurable regularization options.	2	3	

Hintergrund: Berechnung TCL

AQI Automotive Quality Institute

Vertrauensbedarf für die Werkzeugnutzung

Ausgangslage: Wie hoch muss das Vertrauen in das Tool sein (bzw. ist eine Toolqualifikation notwendig)?

Table 3 — Determination of the Tool Confidence Level (TCL)

		Tool error detection		
		TD1	TD2	TD3
Tool impact	TI1	TCL1	TCL1	TCL1
	TI2	TCL1	TCL2	TCL3

Quelle: ISO 26262:2018, Kap. 11.4.5.4

↓

Aspekt	Bild	Textbeschreibung
Ziel	***Zuordnung des passenden Tool Confidence Levels (TCL)*** *auf Basis von Tool Impact (TI) und Tool Error Detection (TD)*	*Festlegen,* ***wie hoch das Vertrauen*** *in das Tool sein muss (bzw. ob eine Toolqualifikation notwendig ist)*
TCL1 (niedrig)	▪ Tool hat **keinen sicherheitsrelevanten Einfluss (TI1)** oder ▪ Fehler werden **sicher erkannt (TI2 + TD1)** → **Keine Toolqualifikation notwendig**	▪ **Geringe Bedeutung** für Produktqualität ▪ Keine Toolqualifikation notwendig → Vertrauen in Toolverhalten aus Sicht ISO 26262 **nicht kritisch**
TCL2 (mittel)	▪ Tool hat sicherheitsrelevanten Einfluss (TI2) ▪ Fehler **nicht sicher erkennbar (TD2)**	▪ Tool **wichtig für Produktqualität** ▪ Toolqualifikation erforderlich ▪ Anforderungen abhängig vom ASIL-Level
TCL3 (hoch)	▪ Tool hat sicherheitsrelevanten Einfluss (TI2) ▪ Fehler **bleiben unentdeckt (TD3)**	▪ **Hohe Bedeutung** für Produktqualität ▪ **Toolqualifikation zwingend nötig**, hohe Anforderungen ▪ Unterschiede zwischen TCL2 und TCL3 sind methodisch relevant, aber nicht dramatisch (je nach ASIL)

Hintergrund: Berechnung TCL

AQI Automotive Quality Institute

Vertrauensbedarf für die Werkzeugnutzung

Ausgangslage: Wie hoch muss das Vertrauen in das Tool sein (bzw. ist eine Toolqualifikation notwendig)?

Table 3 — Determination of the Tool Confidence Level (TCL)

		Tool error detection		
		TD1	TD2	TD3
Tool impact	TI1	TCL1	TCL1	TCL1
	TI2	TCL1	TCL2	TCL3

Quelle: ISO 26262:2018, Kap. 11.4.5.4

Aspekt	Bild	Textbeschreibung
Ziel	***Zuordnung des passenden Tool Confidence Levels (TCL)*** *auf Basis von Tool Impact (TI) und Tool Error Detection (TD)*	*Festlegen,* ***wie hoch das Vertrauen*** *in das Tool sein muss (bzw. ob eine Toolqualifikation notwendig ist)*
TCL1 (niedrig)	▪ Tool hat **keinen sicherheitsrelevanten Einfluss (TI1)** oder ▪ Fehler werden **sicher erkannt (TI2 + TD1)** → **Keine Toolqualifikation notwendig**	▪ **Geringe Bedeutung** für Produktqualität ▪ Keine Toolqualifikation notwendig → Vertrauen in Toolverhalten aus Sicht ISO 26262 **nicht kritisch**
TCL2 (mittel)	▪ Tool hat sicherheitsrelevanten Einfluss (TI2) ▪ Fehler **nicht sicher erkennbar (TD2)**	▪ Tool **wichtig für Produktqualität** ▪ Toolqualifikation erforderlich ▪ Anforderungen abhängig vom ASIL-Level
TCL3 (hoch)	▪ Tool hat sicherheitsrelevanten Einfluss (TI2) ▪ Fehler **bleiben unentdeckt (TD3)**	▪ **Hohe Bedeutung** für Produktqualität ▪ **Toolqualifikation zwingend nötig**, hohe Anforderungen ▪ Unterschiede zwischen TCL2 und TCL3 sind methodisch relevant, aber nicht dramatisch (je nach ASIL)

Hintergrund: Berechnung TCL

AQI | Automotive Quality Institute

Vertrauensbedarf für die Werkzeugnutzung

Ausgangslage: Wie hoch muss das Vertrauen in das Tool sein (bzw. ist eine Toolqualifikation notwendig)?

Table 3 — Determination of the Tool Confidence Level (TCL)

		Tool error detection		
		TD1	TD2	TD3
Tool impact	TI1	TCL1	TCL1	TCL1
	TI2	TCL1	TCL2	TCL3

Quelle: ISO 26262:2018, Kap. 11.4.5.4

↓

Aspekt	Bild	Textbeschreibung
Ziel	***Zuordnung des passenden Tool Confidence Levels (TCL)** auf Basis von Tool Impact (TI) und Tool Error Detection (TD)*	*Festlegen, **wie hoch das Vertrauen** in das Tool sein muss (bzw. ob eine Toolqualifikation notwendig ist)*
TCL1 (niedrig)	▪ Tool hat **keinen sicherheitsrelevanten Einfluss** (TI1) oder ▪ Fehler werden **sicher erkannt** (TI2 + TD1) → **Keine Toolqualifikation notwendig**	▪ **Geringe Bedeutung** für Produktqualität ▪ Keine Toolqualifikation notwendig → Vertrauen in Toolverhalten aus Sicht ISO 26262 **nicht kritisch**
TCL2 (mittel)	▪ Tool hat sicherheitsrelevanten Einfluss (TI2) ▪ Fehler **nicht sicher erkennbar (TD2)**	▪ Tool **wichtig für Produktqualität** ▪ Toolqualifikation erforderlich ▪ Anforderungen abhängig vom ASIL-Level
TCL3 (hoch)	▪ Tool hat sicherheitsrelevanten Einfluss (TI2) ▪ Fehler **bleiben unentdeckt** (TD3)	▪ **Hohe Bedeutung** für Produktqualität ▪ **Toolqualifikation zwingend nötig**, hohe Anforderungen ▪ Unterschiede zwischen TCL2 und TCL3 sind methodisch relevant, aber nicht dramatisch (je nach ASIL)

Hintergrund: Berechnung TCL

AQI | Automotive Quality Institute

Vertrauensbedarf für die Werkzeugnutzung

Ausgangslage: Wie hoch muss das Vertrauen in das Tool sein (bzw. ist eine Toolqualifikation notwendig)?

Table 3 — Determination of the Tool Confidence Level (TCL)

		Tool error detection		
		TD1	TD2	TD3
Tool impact	TI1	TCL1	TCL1	TCL1
	TI2	TCL1	TCL2	TCL3

Quelle: ISO 26262:2018, Kap. 11.4.5.4

Aspekt	Bild	Textbeschreibung
Ziel	***Zuordnung des passenden Tool Confidence Levels (TCL)** auf Basis von Tool Impact (TI) und Tool Error Detection (TD)*	*Festlegen, **wie hoch das Vertrauen** in das Tool sein muss (bzw. ob eine Toolqualifikation notwendig ist)*
TCL1 (niedrig)	▪ Tool hat **keinen sicherheitsrelevanten Einfluss** (TI1) oder ▪ Fehler werden **sicher erkannt** (TI2 + TD1) → **Keine Toolqualifikation notwendig**	▪ **Geringe Bedeutung** für Produktqualität ▪ Keine Toolqualifikation notwendig → Vertrauen in Toolverhalten aus Sicht ISO 26262 **nicht kritisch**
TCL2 (mittel)	▪ Tool hat sicherheitsrelevanten Einfluss (TI2) ▪ Fehler nicht sicher erkennbar (TD2)	▪ Tool wichtig für Produktqualität ▪ Toolqualifikation erforderlich ▪ Anforderungen abhängig vom ASIL-Level
TCL3 (hoch)	▪ Tool hat sicherheitsrelevanten Einfluss (TI2) ▪ Fehler **bleiben unentdeckt** (TD3)	▪ **Hohe Bedeutung** für Produktqualität ▪ **Toolqualifikation zwingend nötig**, hohe Anforderungen ▪ Unterschiede zwischen TCL2 und TCL3 sind methodisch relevant, aber nicht dramatisch (je nach ASIL)

FMEA Excel: Berechnung TCL

AQI Automotive Quality Institute

Vertrauensbedarf für die Werkzeugnutzung

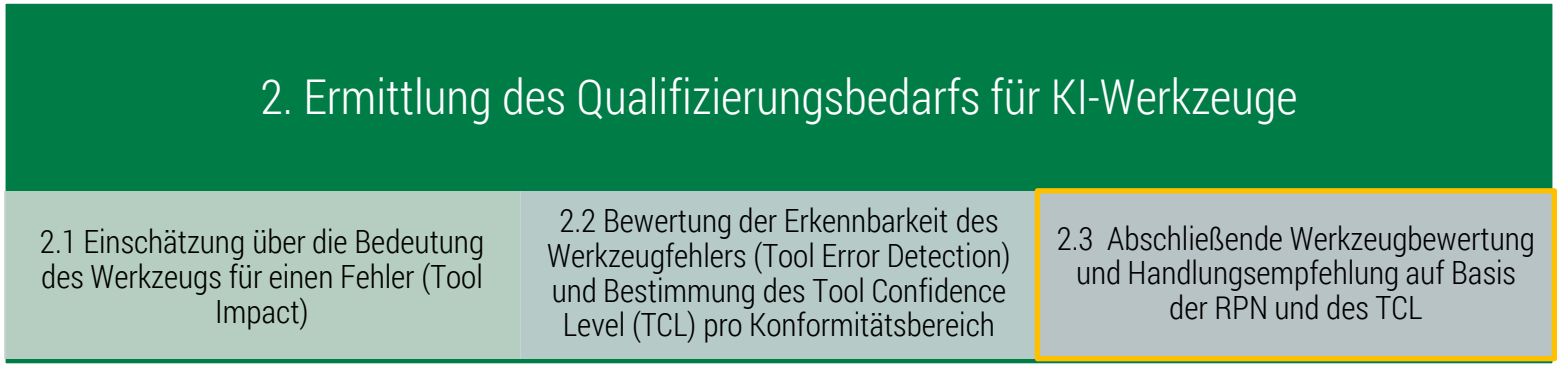
Ausgangslage: Wie hoch muss das Vertrauen in das Tool sein (bzw. ist eine Toolqualifikation notwendig)?

Eingabe	Tool Confidence Level - Einschätzung über das Vorhandensein des Vertrauens
Skala	Vertrauenslevel (1 = niedrig, 2 = mittel, 3 = hoch)

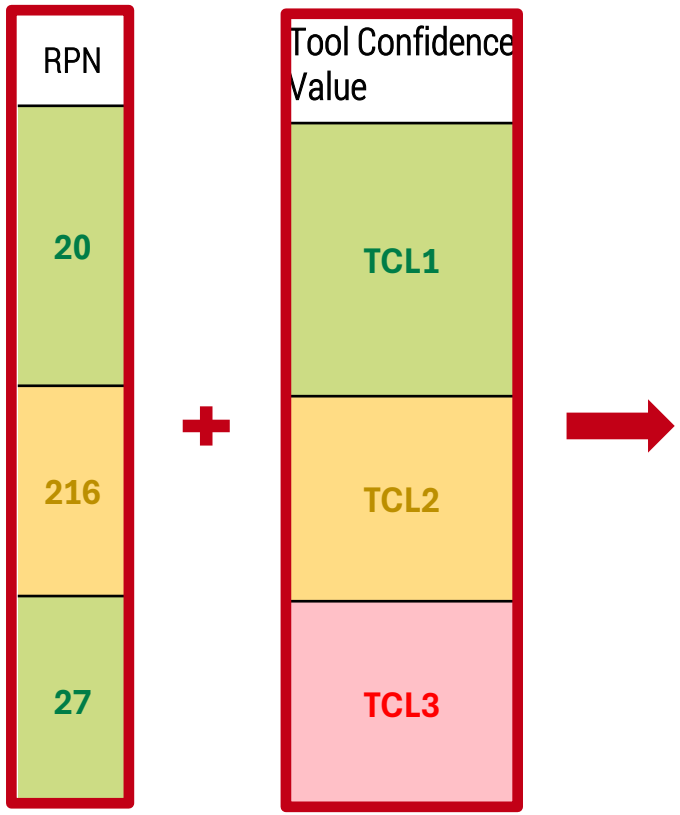
Potential Tool Failure	Testable Tool Capabilities	Tool Impact Value	Tool Error Detection	Tool Confidence Value
May fail to guarantee deterministic training execution and runtime validation, resulting in trained models that do not meet real-time safety-critical performance requirements.	Framework support for model optimization (e.g., TensorRT, ONNX Runtime optimizations). Support for hardware-specific runtime optimizations and deployment targeting (e.g., EdgeTPU, CUDA kernels). Framework support for pipeline fusion, minimal preprocessing APIs, and asynchronous inference.	1	1	TCL1
May fail to provide comprehensive error logging and runtime monitoring, resulting in trained models with limited traceability for failure analysis in safety-critical systems.	Framework support for structured error reporting, custom callbacks, and logging integration. Built-in hooks for anomaly detection during training and validation (gradient checkers, divergence detectors). ML lifecycle management tools (e.g., MLflow, Weights & Biases) integration for metadata and error tracking.	2	2	TCL2
May fail to enforce deterministic training procedures and runtime validation, resulting in trained models with unpredictable behavior and poor generalization in safety-critical applications.	Explicit deterministic training support with reproducibility settings and controlled parallelism. Built-in validation dashboards and metric tracking APIs. Numerical stability modules with configurable regularization options.	2	3	TCL3

2. Ermittlung des Qualifizierungsbedarfs für KI- Werkzeuge

2.3 Werkzeugbewertung und Handlungsempfehlung auf Basis der RPN und des TCL



FEMA Excel: Zusammenfassung und Handlungsempfehlung



Qualifizierungsempfehlung gemäß ISO 26262:2018, Kap. 11.4.6

1. Schritt: Bewertung des Vertrauensbedarf (TCL)

TCL = 1

keine Tool-Qualifizierung notwendig, da bereits ausreichend Maßnahmen vorhanden sind, um Fehler zu erkennen!

TCL = 2 oder 3

Tool-Qualifizierung notwendig, da diese eine **höhere Auswirkung** auf die **funktionale Sicherheit** haben!

Weiter auf Folgefolie

FEMA Excel: Zusammenfassung und Empfehlung

Automotive Quality Institute

Nutze „Mitigations“ and „Testable Tool Capabilities“ für weitere Absicherungs- und Qualifizeriungsmaßnahmen.

Mitigations	Testable Tool Capabilities	Tool Confidence Value
✓ Integrate structured exception handling and mandatory logging for all critical training and inference operations. ✓ Implement runtime monitors for numerical instabilities (e.g., NaNs, divergence) and alert on thresholds. – Embed pipeline-wide metadata capture and systematic experiment logging into the training framework.	✓ Framework support for structured error reporting, custom callbacks, and logging integration. – Built-in hooks for anomaly detection during training and validation (gradient checkers, divergence detectors). ✓ ML lifecycle management tools (e.g., MLflow, Weights & Biases) integration for metadata and error tracking.	TCL2
✓ Implement deterministic training pipelines with seed control and precision configuration to ensure repeatability. ✓ Integrate continuous performance monitoring during training with automatic validation checkpoints. – Use frameworks that incorporate gradient clipping, normalization, and regularization for improved numerical robustness.	✓ Explicit deterministic training support with reproducibility settings and controlled parallelism. ✓ Built-in validation dashboards and metric tracking APIs. – Numerical stability modules with configurable regularization options.	TCL3

Bitte beachten Sie: Die Excel-Anwendung unterstützt derzeit keine direkte Kennzeichnung und Einfärbung der oben dargestellten “Mitigations”und “Testable Tool Capabiliites”.