

## LLM-basierte Fehlerbäume mit RAG und Data Governance

# Adaptive Fehlersuche in der Instandhaltung

S. Dixit, J. Schwab

**ZUSAMMENFASSUNG** Der Beitrag stellt ein LLM-gestütztes, adaptives Assistenzsystem für die Diagnose und Instandsetzung von Industrieanlagen vor. Freitexte werden bereinigt, kategorisiert und in kuratierte Fehlerbäume überführt. Das System kombiniert RAG mit Dialog, ordnet Anfragen zu, führt durch Diagnosepfade und bindet Kontextdaten ein. Es bewertet die Gespräche und aktualisiert die Bäume. Freigabe sichert Qualität. Die Validierung basiert auf einer zwei-monatigen Testphase an der Anlage.

### STICHWÖRTER

Digitalisierung, Instandhaltung, Künstliche Intelligenz (KI)

## Adaptive troubleshooting in maintenance – LLM-based fault trees with RAG and data governance

**ABSTRACT** This article presents an LLM-based, adaptive assistance system for the diagnosis and repair of industrial machines. Free texts are cleaned up, categorized and transferred to curated fault trees. The system combines RAG with dialogue, assigns queries, guides users through diagnostic paths, and integrates contextual data. It evaluates the dialogues and updates the trees. Approval ensures quality. Validation is based on a two-month test phase on the machine.

## 1 Einleitung

Deutsche Unternehmen investieren rund 140 Milliarden Euro jährlich in die Instandhaltung von Maschinen und Anlagen [1]. Die Reduktion ungeplanter Stillstände erschließt erhebliche Einsparpotenziale und verbessert die Wirtschaftlichkeit [2].

Kommt es zu einem Produktionsausfall, kann eine Anlage abhängig von Architektur, Redundanz und Robustheit teilweise oder vollständig funktionsunfähig werden. Mit zunehmender Systemgröße und Komplexität steigt der Aufwand für die Fehlersuche deutlich. Eine zügige Diagnose setzt tiefes domänenspezifisches Instandhaltungswissen voraus. In der Praxis sind die erforderlichen Expertinnen und Experten häufig stark ausgelastet oder nicht verfügbar [3]. In solchen Situationen übernehmen weniger erfahrene Mitarbeitende die Störungsbeseitigung. Da das Wissen von erfahrenen Mitarbeitenden überwiegend implizit ist, fehlen neuen Mitarbeitenden häufig diese strukturierten Wissensinhalte sowie praxisbewährte, schrittweise Entscheidungslogiken, um eine schnelle Störungsbehebung durchführen zu können [4].

Im Störfall stehen als Datenbasis meist einfache Fehlermeldungen und Schichtbücher mit Freitexteinträgen heterogener Qualität zur Verfügung. Wissen über wiederkehrende Störungen und wirksame Abhilfemaßnahmen ist verteilt über verschiedene, überwiegend unstrukturierte Quellen, etwa Schichtbücher, Erfahrungswissen einzelner Mitarbeitender, Störungsmeldungen und Instandhaltungsaufträge. Bekannte Lösungen sind im akuten Fall meist nicht schnell auffindbar.[5]

Um Stillstandszeiten zu reduzieren, ist es entscheidend, bekannte Lösungen und Instandhaltungswissen in umsetzbaren

Anleitungen bereitzustellen [6]. Die bestehende Wissensbasis lässt sich jedoch nur mit erheblichem Aufwand systematisch erweitern und aktualisieren. Somit treten Probleme wiederholt auf und die Stillstandszeiten verlängern sich [7].

Um diese Lücken zu schließen, ist eine zentral verfügbare, kontinuierlich gepflegte, wachsende und semantisch durchsuchbare Wissensbasis nötig. Sie bildet die diagnostische Denkweise erfahrener Mitarbeiter ab, führt systematisch von Symptomen zu wahrscheinlichen Ursachen und liefert kontextbezogene Handlungsempfehlungen für Wartung und Instandsetzung. Ergänzend sollten standardisierte Diagnosepfade, Standardarbeitsanweisungen, klar definierte Eskalationsschleifen und Feedbackmechanismen integriert werden. So lassen sich Lernschleifen schließen, die Wirksamkeit von Maßnahmen messbar erhöhen, die „Mean-Time-To-Repair“ senken, Wiederholstörungen vermeiden und die Gesamtanlageneffektivität nachhaltig steigern.

Das Aufkommen generativer künstlicher Intelligenz (KI), insbesondere großer Sprachmodelle (Large Language Models, LLM) eröffnet neue Möglichkeiten für die Nutzung von Instandhaltungswissen [8]. Natürlichsprachliche Texte – wie Störungsmeldungen in Freitextform oder Auszüge aus Handbüchern – können nicht nur verstanden, sondern auch semantisch eingeordnet, zusammengefasst und in strukturierte Formate (etwa Fehlerbäume oder Maßnahmenlisten) überführt werden. Dasselbe Modell kann dieses Wissen anschließend wieder in Form eines dialogbasierten Assistenzsystems bereitstellen: Instandhalter beschreiben ein Problem in Alltagssprache, das System stellt gezielte Rückfragen, schlägt Diagnoseschritte und Maßnahmen vor und greift dabei auf die zuvor strukturierte Wissensbasis

zurück. Auf diese Weise wird unstrukturiertes Erfahrungswissen, das bisher schwer auffindbar war, in einen systematischen, reproduzierbaren Diagnoseprozess überführt.

## 2 Grundlagen

### 2.1 Instandhaltung und Fehlersuche-Prozess

Aufgabe der Instandhaltung ist es, die Funktion und die Leistungsfähigkeit von Maschinen und Anlagen dauerhaft zu sichern. Mit der steigenden Komplexität moderner Produktionssysteme gewinnt sie weiter an Bedeutung. Sie gliedert sich in vorbeugende Instandhaltung und störungsbedingte Instandsetzung. [1]

Die Instandsetzung umfasst nach DIN 31051 alle Maßnahmen zur Rückführung einer Betrachtungseinheit in einen funktionsfähigen Zustand [9]. Verbesserungen sind davon ausgenommen. Die Instandsetzung lässt sich in drei Arten unterscheiden: geplante Instandsetzung, vorbereitende Instandsetzung und unvorhergesehene Instandsetzung. Bei unvorhergesehener Instandsetzung ist eine systematische Fehlersuche erforderlich. Die Diagnose zählt zu den größten nicht wertschöpfenden Zeitaufwänden der Instandhaltung. [1]

Fehlerdiagnose bezeichnet den Prozess, die zugrunde liegenden Ursachen eines Fehlers präzise zu bestimmen. Nach der Fehlererkennung identifizieren Instandhalter die Symptome, werten Hinweise aus, interpretieren Meldungen und leiten ab, welche Komponenten versagt haben und aus welchen Gründen. Die Vielzahl an Einflussfaktoren macht diesen Prozess komplex und anspruchsvoll. Er erfordert fundiertes technisches Können und Erfahrung. Steht die Diagnose fest, folgt die zielgerichtete Fehlerbehebung. [3]

Ein Fehlersuchbaum hilft, komplexe Industrieanlagen schnell und systematisch zu diagnostizieren und zu warten, um Ausfallzeiten zu minimieren. Er führt die Mitarbeitenden schrittweise durch Fragen, Prüfungen und Entscheidungen bis zur Fehlerursache und zur geeigneten Maßnahme. Ausgangspunkt ist eine Komponente. Daran schließen sich mögliche Fehler, deren Ursachen und die empfohlene Reparaturaktion an. Breite und Tiefe der Struktur variieren je nach Anwendungsfall. Für die Erstellung sind die Betriebsweise und die Fehlermodi zu analysieren sowie Handbücher, Wartungshistorien, Konstruktionsunterlagen und Expertenwissen auszuwerten. Der manuelle Aufbau ist aufwendig und kostenintensiv. Große Sprachmodelle können die Erstellung deutlich beschleunigen. [10]

### 2.2 Large Language Models (LLM)

LLM sind tiefneuronale Sprachmodelle auf Basis der Transformer-Architektur [11]. Sie werden auf sehr großen Textkorpora vortrainiert und lernen dabei statistische Zusammenhänge zwischen Wörtern, Sätzen und Textstrukturen. Dadurch sind sie in der Lage, natürlichsprachliche Eingaben zu verarbeiten, zu interpretieren und kontextabhängige Ausgaben zu erzeugen [12]. Die Leistungsfähigkeit von LLMs beruht nicht auf explizitem Fachwissen, sondern auf der Verallgemeinerung sprachlicher Muster aus Trainingsdaten. Domänenspezifisches Wissen muss daher in der Regel durch zusätzliche Mechanismen bereitgestellt werden, beispielsweise über externe Wissensquellen oder strukturierte Datenbanken.

Für industrielle Anwendungsszenarien lassen sich drei grundlegende, technologiebedingte Fähigkeiten von LLMs identifizieren, die unabhängig vom konkreten Einsatzfall relevant sind:

1. Sprachverstehen und Dialogfähigkeit: Verarbeitung von Freitext-Eingaben und Generierung kontextabhängiger Antworten.
  2. Strukturierung unstrukturierter Daten: Transformation von Text in formalisierte Repräsentationen (zum Beispiel Tabellen, Attribute, Relationen).
  3. Generative Wissensrepräsentation: Überführung vorhandener Textinformationen in alternative Darstellungsformen (etwa Entscheidungsbäume, Schrittfolgen, Zusammenfassungen).
- Beim Einsatz generativer Sprachmodelle in technischen Domänen treten zwei Spannungsfelder auf:

- Allgemeines Sprachwissen versus Domänenwissen: LLMs verfügen primär über allgemeines Sprach- und Weltwissen. Fachlich korrekte Aussagen in industriellen Anwendungen erfordern daher die gezielte Einbindung geprüfter, domänenspezifischer Informationsquellen.
- Wissensgenerierung versus Wissensakkumulation: Ohne strukturiertes Wissensmanagement kann die fortlaufende Ablage neuer Interaktionen zu Redundanzen und widersprüchlichen Informationen führen, was die Qualität von Assistenzsystemen langfristig beeinträchtigt.

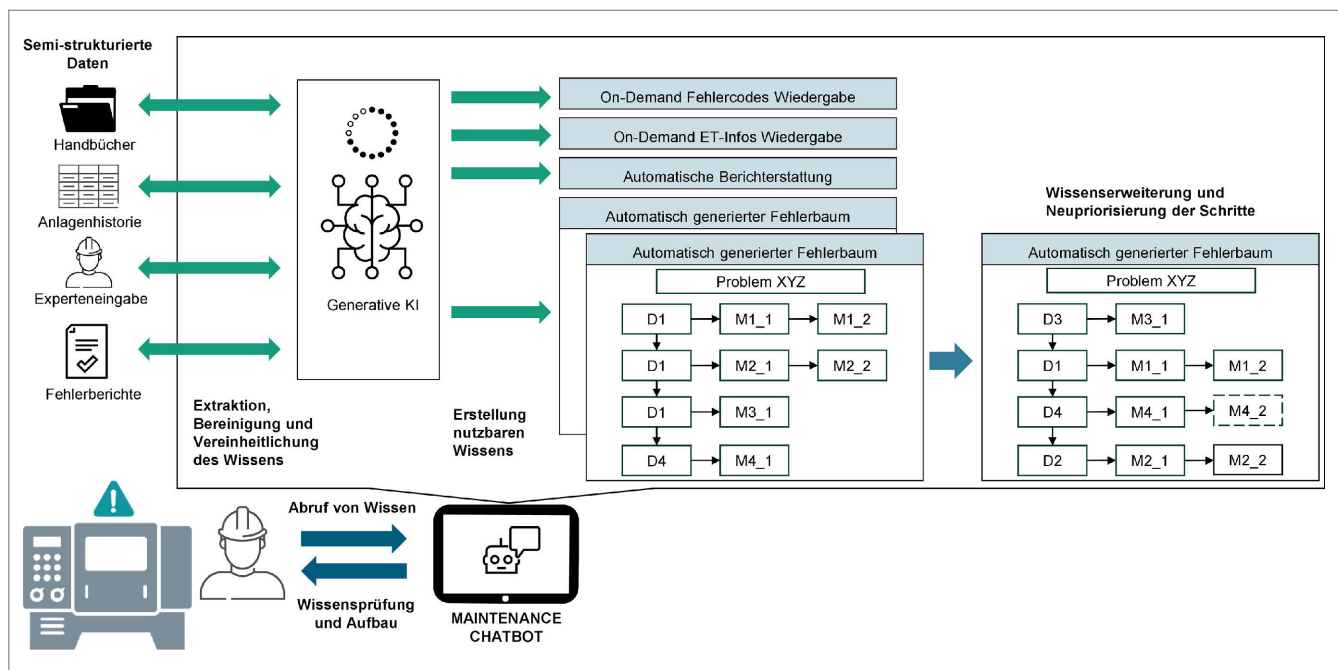
Reine LLMs stoßen an Grenzen, wenn aktuelle oder unternehmensspezifische Anlageninformationen benötigt werden [13]. Ein technischer Ansatz zur Erweiterung des verfügbaren Wissens besteht in der Einbindung externer Dokumente in den Generierungsprozess (Retrieval Augmented Generation, RAG) [14].

Das Grundprinzip von RAG besteht darin, Nutzeranfragen in natürlicher Sprache zu verarbeiten, relevante Wissensseinheiten aus einer angebotenen Wissensbasis abzurufen und diese als zusätzlichen Kontext für die Antwortgenerierung durch ein LLM zu verwenden. Durch die Kopplung von Generierung und Dokumentenabruf können Antworten erzeugt werden, welche auf spezifischen und nachvollziehbaren Informationsquellen basieren. Wissensbasierte Assistenzsysteme erfordern darüber hinaus Konzepte zur strukturierten Organisation und Pflege der zugrunde liegenden Wissensbasis. Im Unterschied zu klassischen Machine-Learning-Ansätzen steht dabei nicht primär das erneute Training des Modells im Vordergrund, sondern die Anpassung und Weiterentwicklung der Wissensrepräsentation.

Die Speicherung von Interaktionen kann grundsätzlich zur Erweiterung der Wissensbasis beitragen. Ohne geeignete Strukturierungs- und Validierungsmechanismen besteht aber die Gefahr von Redundanzen und Inkonsistenzen. Für den langfristigen Einsatz solcher Systeme ist daher eine kontrollierte Integration neuer Informationen in bestehende Wissensstrukturen erforderlich. Dies umfasst Prozesse zur Erfassung von Feedback, zur Überprüfung von Änderungen sowie zur Sicherstellung der Konsistenz der gespeicherten Inhalte. Die fachliche Verantwortung für die Validität der Inhalte verbleibt hier bei domänenspezifischen Expertinnen und Experten. [15]

## 3 Stand der Technik

Große Sprachmodelle haben sich in den vergangenen Jahren als leistungsfähige Werkzeuge für wissensintensive Aufgaben etabliert. Anwendungen in der Bankkommunikation und der klinischen Entscheidungsunterstützung zeigen, dass solche



**Bild 1** Gesamtkonzept von Bereinigung, Nutzung und Aufbau von Wissen in der Instandhaltung. Grafik: Fraunhofer IPA

Modelle unstrukturierte Freitextquellen effizient durchsuchen und nutzerorientierte sowie kontextrelevante Antworten erzeugen können [16–18]. Kommerzielle Systeme kombinieren hierzu in der Regel RAG mit kontrollierter Nutzung von Werkzeugen. So werden Halluzinationen verringert und die Nachvollziehbarkeit der Antworten wird erhöht.

Im technischen Service und in der Instandhaltung adressieren LLM-gestützte Chatbots vergleichbare Anforderungen. Sie unterstützen das schnelle Auffinden relevanter Informationen aus heterogenen Wissensbeständen, darunter etwa Handbücher, Tickets sowie Wartungspläne, und ermöglichen eine geführte Fehlersuche. Rittikusittichai *et al.* [19] demonstrieren dies anhand eines wissensbasierten Leitfadens für die Leiterplattenfertigung, der strukturierte Lösungen aus einem zentralen Wissensspeicher ableitet.

Historisch beruhen digitale Unterstützungssysteme für die Fehlersuche auf regelbasierten und baumartigen Entscheidungsstrukturen. Bereits Patel *et al.* [3] zeigten im Jahr 1996 ein datenwissenschaftlich motiviertes System mit Fehlerbäumen für die schrittweise Diagnoseführung. Seither hat sich der Schwerpunkt von statischen und manuell kuratierten Strukturen zu datengetriebenen und dynamisch generierten Troubleshooting-Prozessen verschoben. Vidyaratne *et al.* leiten aus Betriebsdaten kausale Ketten ab, die von Equipment über Komponente und Fehlerbild zu Ursache und Maßnahme führen [10], und zeigen zudem, dass sich Fehlerbäume direkt aus FMEA-Tabellen extrahieren lassen [20]. Wen *et al.* entwickeln ein mehrstufiges Schlussfolgerungsverfahren, das komplexe Kausalzusammenhänge in der Fehlersuche handhabbar macht und evidenzbasierte Schlussfolgerungen stützt [21].

Der Stand der Technik bietet kein System, das fortlaufend neue Fehlerbäume sowohl aus vorhandenen Daten als auch aus Konversationen erstellt und diese anhand weiterer im System geführter Konversationen kontinuierlich aktualisiert. Für komplexe Anlagen ist dies entscheidend, da viele Probleme nicht von Anfang an vorliegen und immer wieder neue entstehen. Diese Lücke bildet den Ausgangspunkt für den nachfolgenden Ansatz.

## 4 Konzept und Umsetzung

Zur Unterstützung von Instandhaltern wird ein kontrolliert adaptiver Fehlersuchassistent entwickelt, der Nutzerinteraktionen systematisch auswertet und zur gezielten Weiterentwicklung bestehender Wissensstrukturen nutzt. Das System lernt nicht autonom, sondern basiert auf versionierten und fachlich geprüften Anpassungen von Fehlerbäumen. Technisch erfolgt dies über Konversationsanalyse, Evidenzgewichtung und die kontrollierte Aktualisierung strukturierter Graphen.

Das Assistenzsystem folgt einem dreistufigen Ansatz, der den Lebenszyklus des Instandhaltungswissens abbildet:

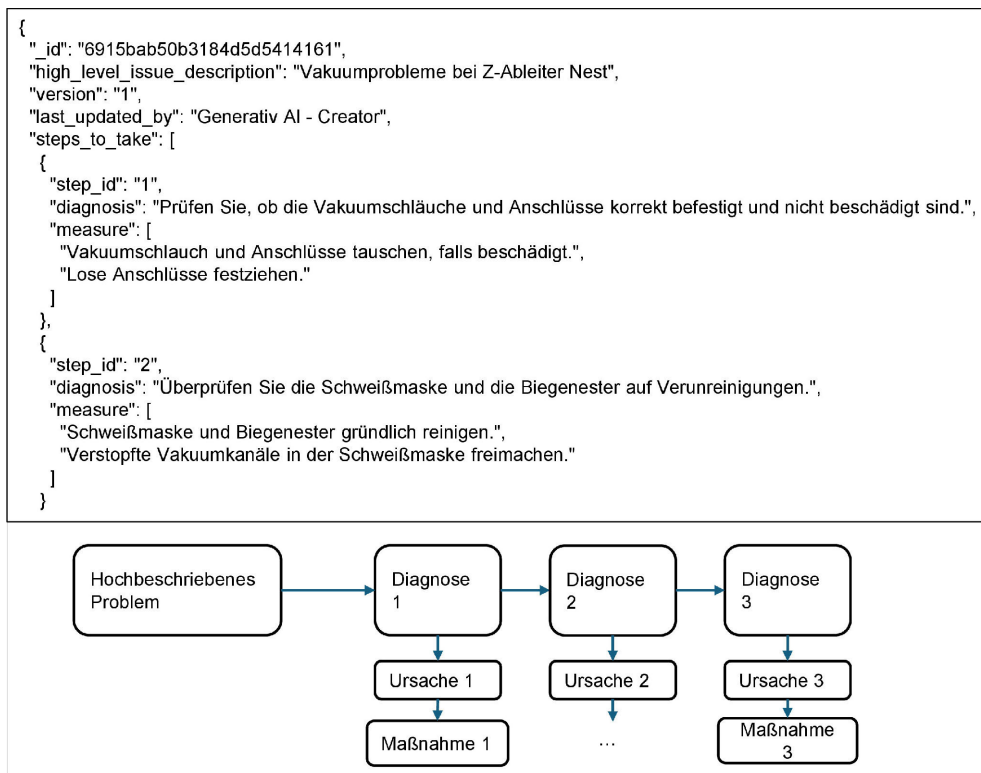
1. Bereinigung und Strukturierung heterogener Datenquellen,
2. geführte Nutzung im Störfall,
3. kontrollierte Erweiterung der Wissensbasis auf Basis realer Nutzung (**Bild 1**).

Im Zentrum steht der Einsatz von LLMs, die ausschließlich über Prompt Engineering und Kontextsteuerung eingesetzt werden. Das Modell selbst wird nicht nachtrainiert; stattdessen wird das Kontextfenster gezielt mit strukturierten Wissens-elementen, Dokumenten und Entscheidungsstrukturen angereichert, um reproduzierbare und domänenspezifische Ausgaben zu gewährleisten. Die folgenden Abschnitte beschreiben die drei Schritte „Wissen bereinigen“, „Wissen nutzen“ und „Wissen erweitern“ im Detail.

### 4.1 Schritt 1: Wissen bereinigen

Die Datenbereinigung und Strukturierung erfolgt vollständig LLM-basiert über einen definierten Prompt-Flow. Regelbasierte Verfahren werden lediglich zur Entfernung personenbezogener Daten eingesetzt. Ziel ist die semantische Verdichtung heterogener Freitexteinträge.

Zunächst generiert das LLM übergeordnete Problemkategorien entlang vorgegebener Dimensionen (zum Beispiel Anlagen-teil, Funktionsgruppe, Symptomtyp). Danach wird jede Störungs-



**Bild 2** Einfacher, durch LLM (Large Language Models) generierter Fehlerbaum. Grafik: Fraunhofer IPA

meldung anhand eines im Prompt definierten Ähnlichkeitskriteriums (Symptom, betroffene Komponente, Fehlerwirkung) einer Kategorie zugeordnet. Kann keine eindeutige Zuordnung erfolgen, wird eine neue Kategorie vorgeschlagen. Für jede Kategorie werden die zugehörigen Einträge aggregiert und dem LLM übergeben, dass daraus einen Fehlerbaum erzeugt. Dabei identifiziert das Modell typische Symptome, leitet Diagnosepfade ab und extrahiert wirksame Maßnahmen, die innerhalb des Baums priorisiert werden. Das Ergebnis ist ein strukturierter, konsolidierter Diagnose- und Maßnahmenpfad je Kategorie (**Bild 2**).

#### 4.2 Schritt 2: Wissen nutzen

Die erzeugten Fehlerbäume bilden die Basis des Assistenzsystems und sind in ein Chat-Interface eingebettet, das Anwender im Störfall schrittweise durch Diagnose und Maßnahmen führt. Die Ergebnisse werden strukturiert erfasst, wodurch auch weniger erfahrene Personen unterstützt und Maßnahmen nachvollziehbar dokumentiert werden.

Beschreibt ein Anwender ein Problem, analysiert das LLM die Eingabe und ordnet sie einer Problemkategorie zu. Existiert ein Fehlerbaum, wird dieser als Diagnosegrundlage verwendet. Der Chatbot stellt gezielte Fragen oder schlägt Prüfschritte vor, die jeweils einem Knoten im Baum entsprechen. Auf Basis der Antworten wird der nächste Schritt ausgewählt.

Zusätzliche Informationen werden über ein RAG-System bereitgestellt, das gezielt auf freigegebene Datenquellen zugreift (zum Beispiel Handbuchauszüge, historische Störungen, Ersatzteilinformationen, freigegebene Fehlerbäume). Der Abruf erfolgt

nur, wenn Metadaten wie Anlage, Modul oder Fehlercode übereinstimmen.

Für nicht strukturierte oder neue Probleme signalisiert das System transparent, dass kein Diagnosepfad vorliegt, und leitet einen definierten Eskalationsprozess ein. Relevante Kontextinformationen werden dabei gebündelt bereitgestellt.

#### 4.3 Schritt 3: Wissen erweitern

Die kontrollierte Weiterentwicklung der Wissensbasis basiert auf der Auswertung realer Nutzung. Nach Abschluss einer Konversation wird der Diagnosepfad durch ein Sprachmodell bewertet (zum Beispiel: korrekt, wirkungslos, erfolgreich). Anwender können ergänzende Maßnahmen über ein geführtes Dialogformular einbringen, das Eingaben direkt in strukturierte Felder überführt.

Liegt ausreichend Feedback vor, wird eine neue Version des Fehlerbaums erzeugt. Das LLM erhält den bestehenden Baum, bewertete Konversationen und gegebenenfalls neue Maßnahmen und erzeugt daraus eine modifizierte Struktur, in der wirksame Maßnahmen priorisiert und redundante entfernt werden.

Die neue Version wird nicht automatisch produktiv gesetzt, sondern in einem Administrationsfrontend visualisiert und fachlich geprüft. Erst nach Freigabe wird sie in das Chatssystem übernommen (**Bild 3**).

Damit wird neues Wissen integriert, ohne dass die Wissensbasis unkontrolliert wächst. Auch neuartige Probleme können nach Freigabe in strukturierter Form in das System übernommen werden, wodurch sich die Wissensbasis schrittweise und kontrolliert weiterentwickelt (**Bild 4**).

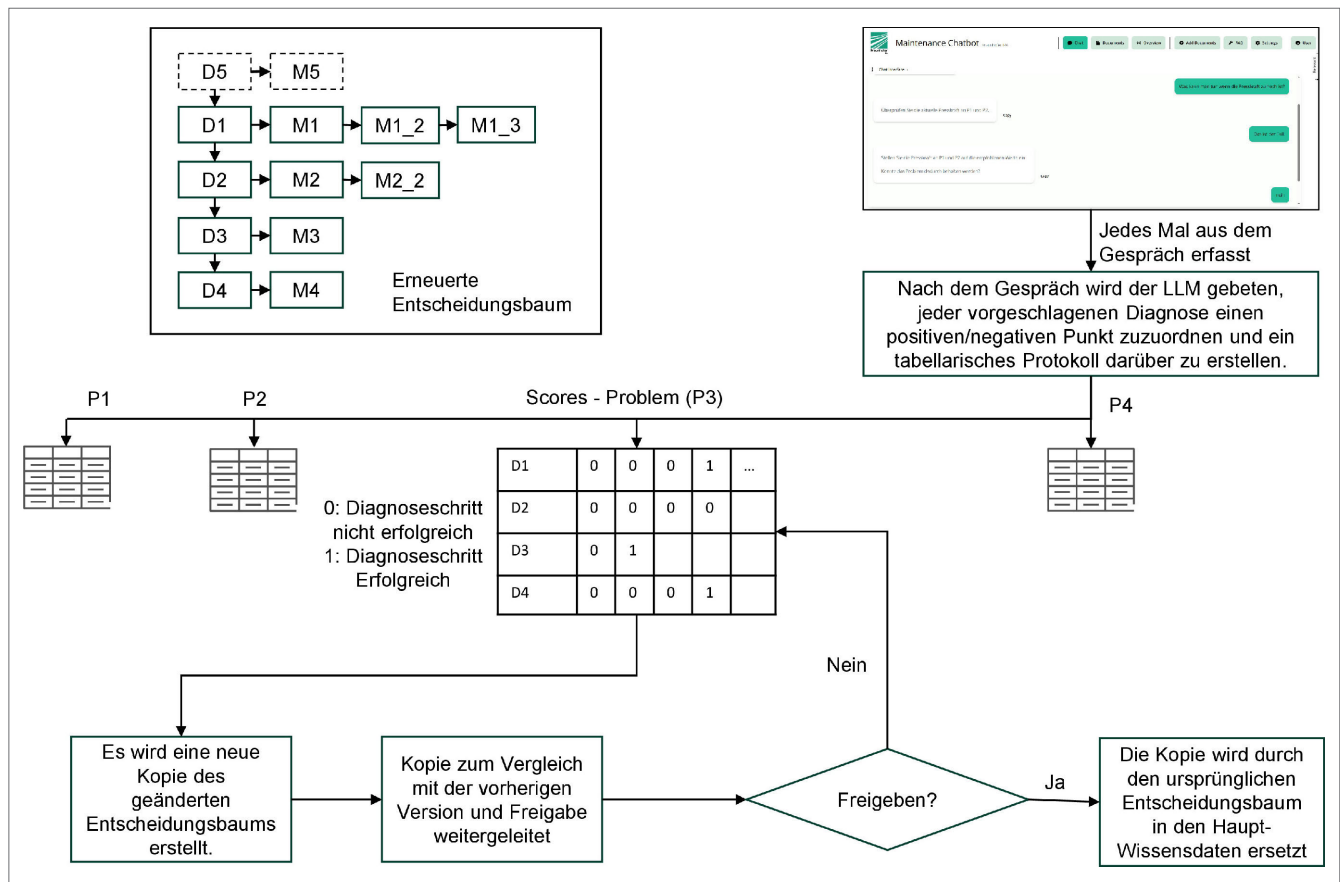


Bild 3 Ein Adaptionprozess basiert auf der Auswertung von Konversationen und der Freigabe. Grafik: Fraunhofer IPA

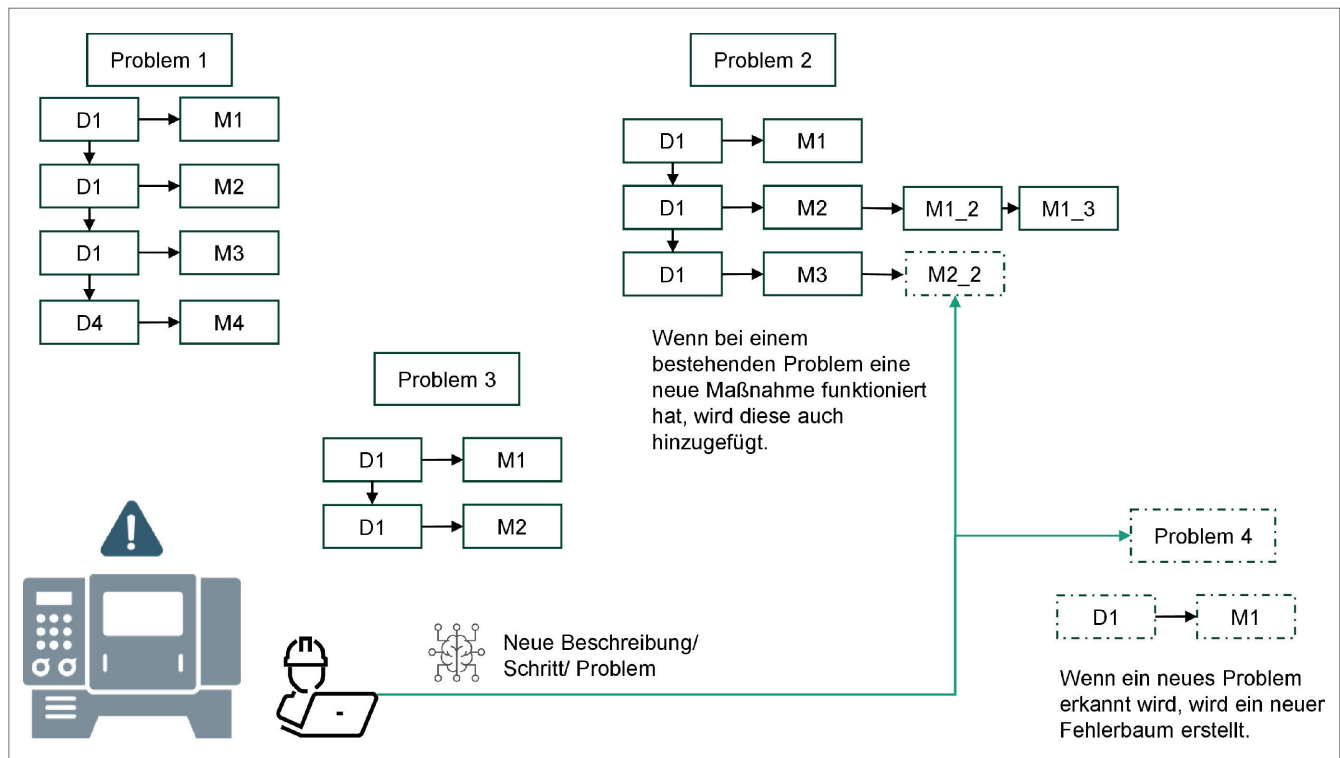


Bild 4 Darstellung: Varianten des adaptiven Lernens bei neuen Problemen und Maßnahmen. Grafik: Fraunhofer IPA

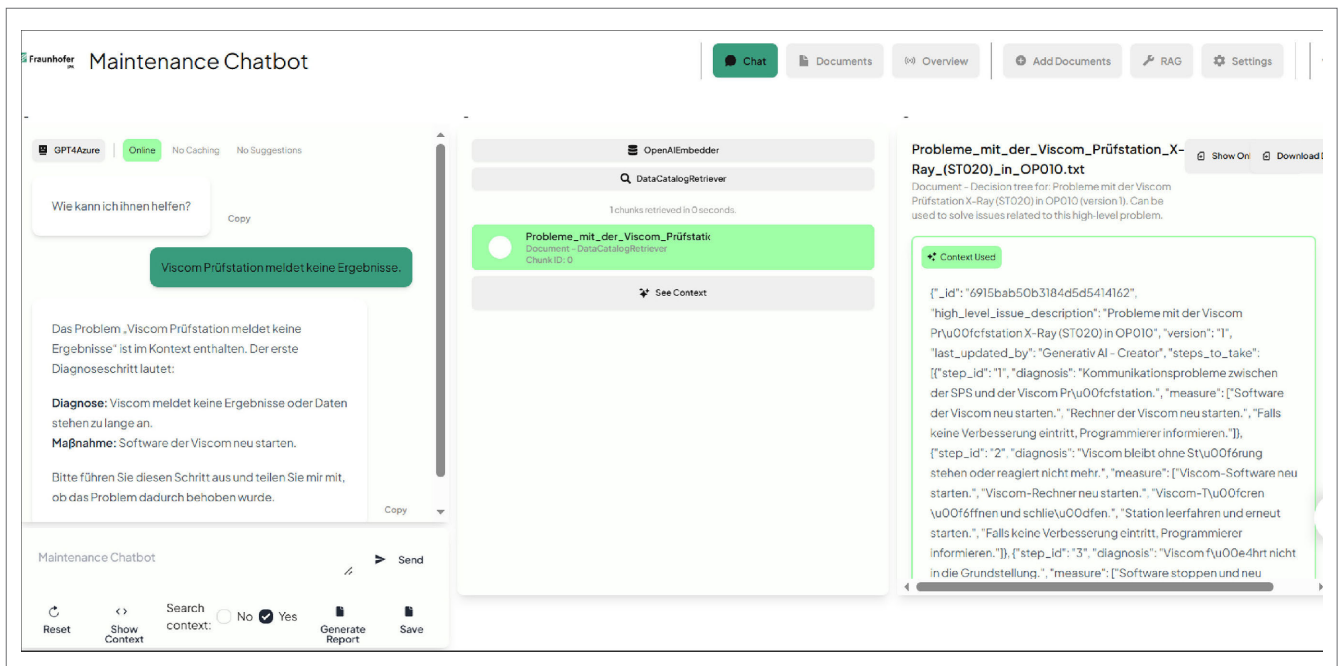


Bild 5 GUI (Graphical User Interface): Chatsystem. Foto: Fraunhofer IPA

## 5 Ergebnisse und Validierung

Im Rahmen des Projekts wurde zunächst ein LLM-basiertes Chatbot-System für eine Produktionslinie an mehreren Anlagen entwickelt. In der ersten Ausbaustufe stellt das System zentrale Informationen zu Fehlercodes und Ersatzteilen bereit, ermöglicht den Abruf der Anlagenhistorie und führt durch Diagnoseschritte, die von Expertinnen und Experten definiert wurden.

Anschließend wurde eine automatisierte Generierung von Fehlerbäumen für Diagnose und Störungsbeseitigung auf Grundlage der Störungsmeldungstabelle entwickelt. Sie ersetzt die manuelle Erarbeitung durch Expertinnen und Experten und erlaubt eine schnelle, konsistente und skalierbare Pflege des Wissensbestands. Darauf aufbauend wurde ein adaptives System entwickelt, das sich selbst aktualisierende Fehlerbäume nutzt und den Wissensbestand kontinuierlich erweitert. Die grafische Benutzeroberfläche (GUI) des Systems ist in **Bild 5** unten zu sehen.

Da große Sprachmodelle im Jahr 2025 weiterhin zu Halluzinationen neigen, wurde ein Freigabesystem mit verbindlicher Versionskontrolle eingeführt. Zusätzlich wurde eine Funktion zur korrekten Nachverfolgung aller vorgenommenen Änderungen implementiert, die Transparenz, Nachvollziehbarkeit und Auditierbarkeit sicherstellt.

**Bild 6** zeigt die GUI des Freigabesystems. Links wird der aktuelle Fehlerbaum dargestellt. Sobald genügend Nutzerinteraktionen stattgefunden haben und bewertet wurden, erscheint rechts eine neue Version, die aus der Bewertung der zugehörigen Konversationen abgeleitet wird. Der Unterschied kann dann von einem Experten geprüft und freigegeben werden, damit der Datensatz ersetzt wird.

### 5.1 Validierungsergebnisse

Die Validierung wurde auch für die drei Schritte „Wissen bereinigen“, „Wissen nutzen“ und „Wissen erweitern“ durchgeführt.

Bei „Wissensbereinigung“ wurden aus circa 1000 Einträgen einer Instandhaltungstabelle zunächst mittels einer LLM-Pipeline 25 Entscheidungsbäume mit maximal 10 Diagnosen und Maßnahmen pro Entscheidungsbaum generiert.

Die generierten Problembeschreibungen für eine Batteriemontageanlage sehen wie folgt aus:

1. Höhenkontrolle
2. Kameraprüfung
3. Schrumpfschläuche oder Luftschläuche
4. Vorfalzen, Anode und Kathodenseite wegen starker Teleskopie
5. DMC-Code, einschließlich Lesefehlern
6. Schneckenantrieb
7. Ablage und Funktionalität des Stäubli Roboters

Hier ist zu sehen, dass die Probleme tatsächlich für verschiedene Fehlergruppen aus der Störungsmeldungstabelle und dem Schichtbuch definiert werden. Wie in Abschnitt 4.1 gezeigt, werden dazu entsprechende Fehlerbäume mit Diagnose und Maßnahmen erstellt. Dadurch können effizient Duplikate, Schreibfehler und irrelevante Einträge entfernt werden.

Anschließend umfasste die Validierung der beiden Phasen „Wissensermittlung“ und „Wissensaktualisierung“ dreißig Anfragen bezüglich der Anlage sowie die Fehlerbehebung über einen Zeitraum von zwei Monaten. Grundlage der Anfragen waren alle Störungen, die in diesem Zeitraum aufgetreten sind.

Der Prozess der Wissensnutzung wurde für detaillierte und komplexere Fragen sowie bei der Wissensaktualisierung validiert. Da das gesamte Funktionsspektrum für die korrekte Unterstützung wichtig ist, gehören die beiden Prozesse zusammen und müssen nicht separat voneinander validiert werden. Dabei wurden 75 % der Anfragen durch das System korrekt bearbeitet. Die erfolgreiche Unterstützung umfasste:

- die korrekte Führung durch bereits in der Datenbank vorhandene Fehlerbäume, bei der mehrere Diagnoseschritte durchlaufen wurden, bis eine korrekte Diagnose vorlag,

**Version 1**

```

{
  "_id": "6915bab50b3184d5d5414150",
  "high_level_issue_description": "Probleme mit der Amada Widerstandsschweißstation, insbesondere Datenfehler, Ausschuss und mechanische Störungen.",
  "version": "1",
  "last_updated_by": "Generativ AI - Creator",
  "steps_to_take": [
    {
      "step_id": "1",
      "diagnosis": "Prüfen, ob die Schieberegister korrekt arbeiten und keine Datenfehler vorliegen.",
      "measure": [
        "Schieberegister löschen und neu initialisieren.",
        "Station leerräumen und alle Positionen zurücksetzen."
      ]
    },
    {
      "step_id": "2",
      "diagnosis": "Untersuchung von Problemen mit der Ablegeposition und den Pucks.",
      "measure": [
        "Leere Pucks aus der Ablegeposition entfernen.",
        "Schlitten A und B leeren und neu starten."
      ]
    }
  ]
}

```

**Version 2**

```

{
  "_id": "691edba40b3184d5d541419f",
  "high_level_issue_description": "Probleme mit der Amada Widerstandsschweißstation, insbesondere Datenfehler, Ausschuss und mechanische Störungen.",
  "version": "2",
  "last_updated_by": "Generativ AI - Updater",
  "steps_to_take": [
    {
      "step_id": "1",
      "diagnosis": "Kontaktplatte berührende Teile prüfen",
      "measure": [
        "Reinigen bzw. gegebenenfalls wechseln"
      ]
    },
    {
      "step_id": "2",
      "diagnosis": "Schweißnest prüfen",
      "measure": [
        "Falls kaputt ist, ausbauen und neue polieren."
      ]
    },
    {
      "step_id": "3"
    }
  ]
}

```

**Ratings für Version A**

conversation_id	step 1	step 2	step 3	step 4	step 5	step 6	step 7	step 8	step 9	step 10
691ec9960b3184d5d5414190	1	1	1	1	1	1				
691ed60a0b3184d5d5414197	1	1	1	1	1	1	1	1	1	1
691ed78a0b3184d5d5414199	0	0	0	0	0	0	0	0	0	0
691ed9ae0b3184d5d541419b	0	0	0	0	0	0	0	0	0	0

**Bild 6** GUI: Änderungen im Fehlerbaum basieren auf dem jeweiligen Scoring der Konversation. Foto: Fraunhofer IPA

- die erfolgreiche Erstellung neuer Diagnoseschritte und Maßnahmen zur Störungsbeseitigung für neu beobachtete Fälle eines bestehenden Problems sowie
- die Generierung neuer Fehlerbäume für bislang nicht erfasste Problemtypen

Bei 25 % der Anfragen erfolgte keine angemessene Unterstützung. Hauptursachen waren:

- fehlerhafte Wissensvermittlung (durch fehlerhafte Auswahl des Entscheidungsbaums, Missverständnisse oder einen zu geringen Informationsgehalt)
- fehlerhafte Wissensaktualisierungsszenarien bei der Neupriorisierung und Formulierung

Die Ursachen der oben genannten 25-prozentigen Fehlschlagquote werden in Abschnitt 5.2 detailliert beleuchtet.

## 5.2 Auswertung der Ergebnisse

### 5.2.1 Fehlerhafte Wissensermittlung

Das System basiert vollständig auf natürlicher Sprache in großen Sprachmodellen. Dieses schlanke Design reduziert die Systemkomplexität, führt jedoch zu Unsicherheiten bei der präzisen Zuordnung von Anlagenkontexten und Problemstellungen zu passenden Wissensselementen. Dies betrifft vor allem ähnliche Komponenten oder Prozesse, deren Unterschiede rein sprachbasiert nicht immer zuverlässig differenziert werden können. Dadurch kann es zu einer fehlerhaften Auswahl von Entscheidungsstrukturen sowie zu Missverständnissen in der Problemformulierung kommen.

Eine stärker strukturierte Abbildung der Produktionsumgebung sowie deren Integration in Backend und Benutzeroberfläche kann die eindeutige Zuordnung unterstützen und die Skalierbarkeit des Systems verbessern. Ein weiteres Problem besteht darin, dass der Abruf relevanter Fehlerstrukturen unzuverlässig sein kann, wenn Nutzeranfragen zu wenig Kontext enthalten. In

solchen Fällen ist eine eindeutige Zuordnung nicht immer möglich. Dies gilt auch für intern gebräuchliche oder domänenspezifische Begriffe, die nicht Teil der zugrunde liegenden Wissensbasis sind und daher nicht erkannt werden. Dies kann zu Missverständnissen bei Suche, Zuordnung und Diagnose führen.

Die Herausforderung liegt darin, dass das System ausschließlich sprachliche Eingaben verarbeitet und keine zusätzlichen strukturierten Interaktionselemente (zum Beispiel grafische Auswahloptionen der konkreten Anlage) nutzt, die eine präzisere Lokalisierung des Problems ermöglichen könnten. Um die Analysequalität zu erhöhen, müssen domänenspezifische Begriffe systematisch in die Wissensbasis integriert werden. Darüber hinaus kann ein strukturierter Rückfrageprozess dazu beitragen, unklare oder unvollständige Eingaben schrittweise zu präzisieren.

### 5.2.2 Fehlerhafte Wissenserweiterung und -aktualisierung

Die Bewertung von Konversationen sowie ihre Überführung in ein Priorisierungs- oder Rankingsystem erfolgt ausschließlich auf Basis der Interpretation des Gesprächsverlaufs durch ein Sprachmodell. Da Antworten abhängig vom Wortlaut unterschiedlich interpretiert werden können, besteht die Gefahr inkonsistenter Bewertungseinträge und daraus resultierender Fehlpriorisierungen. Eine stärker strukturierte Interaktion könnte die Bewertungsgenauigkeit erhöhen, indem Antwortoptionen präziser vorgegeben oder klarer differenziert werden. Somit ließe sich die Ambiguität freier sprachlicher Rückmeldungen reduzieren.

Eine weitere Herausforderung besteht in der konsistenten und eindeutigen Formulierung neu erfasster Störungen zur Erweiterung der Wissensbasis. Erfolgt die Generierung neuer Wissensselemente primär auf Basis freier Nutzereingaben, kann dies zu variierenden Bezeichnungen und Strukturinkonsistenzen führen. Für eine nachhaltige Wissenspflege ist daher eine systematische Einordnung neuer Informationen in bestehende Strukturen erforder-

lich. Dazu sind Prozesse notwendig, die unvollständige Angaben durch gezielte Rückfragen präzisieren und sicherstellen, dass neue Inhalte formal konsistent und inhaltlich valide gespeichert werden.

## 6 Fazit

Der vorgestellte Use Case zeigt, dass Large Language Models in Kombination mit einem Retrieval-Augmented-Generation-Ansatz geeignet sind, unstrukturierte Instandhaltungsdaten systematisch in eine für die Praxis nutzbare Wissensrepräsentation zu überführen. Durch den dreistufigen Prozess aus Wissensbereinigung, geführter Nutzung und kontrollierter Erweiterung konnte aus einer Störungstabelle mit rund 1000 heterogenen Einträgen eine konsolidierte Wissensbasis in Form strukturierter Fehlerbäume erzeugt werden. Diese bildet die Grundlage für einen standardisierten und reproduzierbaren Diagnoseprozess.

Insbesondere der Übergang von unsystematischen Freitexten zu kuratierten Entscheidungsstrukturen erwies sich als zentraler Mehrwert: Wiederkehrende Störungen werden explizit, auffindbar und konsistent bearbeitbar. Die kontrollierte Erweiterung der Wissensbasis zeigt, wie Nutzungsdaten und Feedback genutzt werden können, um bestehende Diagnosepfade gezielt zu verbessern, anstatt lediglich weitere Daten anzuhäufen. Gleichzeitig wird deutlich, dass dieser Ansatz eine klare Governance und fachliche Freigabe erfordert, um eine Verwässerung der Wissensbasis zu vermeiden.

Die Interaktion mit dem System erfolgt aktuell primär über eine natürlichsprachliche Schnittstelle, die vor allem für die offene Problemaufnahme einen niedrigen Einstieg bietet. Gleichzeitig zeigen Arbeiten zu Conversational User Interfaces, dass rein dialogische Schnittstellen bei komplexeren Aufgaben Einschränkungen aufweisen, etwa hinsichtlich Discoverability, Effizienz und kognitiver Belastung [22, 23]. Für strukturierte Diagnosepfade sind visuelle und interaktive Darstellungen daher effizienter und robuster als freie Texteingaben [24]. Langfristig erscheint ein hybrider Ansatz vielversprechend, bei dem Sprachinteraktion und spezialisierte User Interface (UI)-Elemente kombiniert werden. [25, 26]

## 7 Ausblick

Aus methodischer Sicht liegt ein nächster Entwicklungsschritt darin, das System stärker mit statistischen und dynamischen Kontextinformationen zu verknüpfen. Beispiele sind:

- Einbezug von Häufigkeiten und Ausfallwahrscheinlichkeiten (zum Beispiel priorisierte Maßnahmen auf Basis historischer Erfolgsquoten pro Anlage, Schicht, Produktvariante).
- Nutzung zeitlicher Muster (zum Beispiel saisonale Effekte, Wartungszyklen), um Diagnosepfade zu gewichten.
- Verknüpfung mit Prozess- und Maschinendaten (zum Beispiel aktuelle Prozessparameter, Trendverläufe), sodass das System zwischen typischen und atypischen Störungsverläufen unterscheiden kann.

Damit könnte das Assistenzsystem über die rein textuelle Ebene hinauswachsen und vermehrt dynamische Zustände der Anlage berücksichtigen.

Für die Praxis bedeutet das: Der eigentliche Paradigmenwechsel liegt weniger im „Reden mit der Maschine“, sondern in der Fähigkeit, verteiltes Wissen – ob in Texten, Datenbanken oder

Köpfen – semantisch zu integrieren, aufzubereiten und zielgerichtet zu verändern. Reine Spracheingabe ist hier ein wichtiges, aber nicht das einzige Interface. Entscheidend wird sein, wie gut es gelingt, generative KI als augmentierende Schicht in bestehende Arbeitsabläufe einzubetten, ohne die Bedienbarkeit zu verschlechtern oder neue Komplexität zu erzeugen.

Der hier beschriebene Use Case liefert dafür einen ersten Baustein: Er zeigt, wie man aus unstrukturierten Störungsmeldungen strukturierte Fehlerbäume gewinnen, diese im Alltag nutzen und über Feedback gezielt weiterentwickeln kann. Die nächsten Schritte liegen darin, diese Ansätze auf weitere Anlagen und Domänen zu übertragen, quantitative Effekte systematisch zu messen und das Interaktionsdesign von der reinen Konversation hin zu einer integrierten, augmentierten Instandhaltungsumgebung zu entwickeln.

### FÖRDERHINWEIS


Diese Arbeit wurde durch eine Förderung des Bundesministeriums für Bildung und Forschung (BMBF) für das Projekt „Digi-BattPro 4.0“ unter der Fördernummer 03XP0374C unterstützt.

### LITERATUR

- [1] Matyas, K.: Instandhaltungslogistik. Qualität und Produktivität steigern. München: Hanser Verlag 2022
- [2] Al Ameeri, T. A.; Ab Rahman, M. N.; Muhamad, N.: Analysing Effective and ineffective impacts of maintenance strategies on electric power plants: A comprehensive approach. *Energies* 16 (2023) 17, #6243
- [3] Patel, S. A.; Kamrani, A. K.: Intelligent decision support system for diagnosis and maintenance of automated systems. *Computers & Industrial Engineering* 30 (1996) 2, pp. 297–319
- [4] Cárcel-Carrasco, J.; Cárcel-Carrasco, J.-A.: Analysis for the knowledge management application in maintenance engineering: perception from maintenance technicians. *Applied Sciences* 11 (2021) 2, #703
- [5] Sala, R.; Pirola, F.; Pezzotta, G. et al.: Improvement of maintenance-based Product-Service System offering through field data: a case study. *Production & Manufacturing Research* 11 (2023) 1, <https://doi.org/10.1080/21693277.2023.2278313>
- [6] Yew, K.-H.; Foong, O.-M.; Sivarajan, T. P.: Knowledge-based improvement of machine downtime management for IR4.0. 2019 9th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia, 2019, pp. 94–98
- [7] Woods, C.; French, T.; Hodkiewicz, M. et al.: An ontology for maintenance procedure documentation. *Applied Ontology* 18 (2023) 2, pp. 169–206
- [8] Getz, N.; Tong, X.: Large Language Model Accelerated Maintenance Insights. *Annual Conference of the PHM Society* 17 (2025) 1, <https://doi.org/10.36001/phmconf.2025.v17i1.4454>
- [9] DIN 31051:2019-06: Grundlagen der Instandhaltung. Deutsche Fassung, Ausgabe Juni 2019
- [10] Vidyaratne, L.; Lee, X. Y.; Kumar, A. et al.: Generating troubleshooting trees for industrial equipment using Large Language Models (LLM). 2024 IEEE International Conference on Prognostics and Health Management (ICPHM), Spokane, WA, USA, 2024, pp. 116–125
- [11] Vaswani, A.; Shazeer, N.; Parmar, N. et al.: Attention Is All You Need. *ArXiv* 2017, <https://doi.org/10.48550/arXiv.1706.03762>
- [12] Xinying Wang, X. L.: Leveraging Generative AI for predictive maintenance. Building a knowledge base for fault diagnosis. Stand: 2025. Internet: [odr.chalmers.se/server/api/core/bitstreams/5f979de4-1e26-4ba0-8d5d-ffa7fad5b260/content](https://odr.chalmers.se/server/api/core/bitstreams/5f979de4-1e26-4ba0-8d5d-ffa7fad5b260/content). Zugriff am 26.02.2026
- [13] Anjum, K.; Arshad, M. A.; Hayawi, K. et al.: Domain specific benchmarks for evaluating multimodal large language models. *Arxiv* 2025, <https://doi.org/10.48550/arXiv.2506.12958>

- [14] Jiang, C.; Pan, X.; Bao, C. et al.: Feedback-Guided Extraction of Knowledge Base from Retrieval-Augmented LLM Applications. Arxiv 2025, <https://doi.org/10.48550/arXiv.2411.14110>
- [15] Backlund, A.; Petersson, L.: Vending-Bench: A Benchmark for Long-Term Coherence of Autonomous Agents. Arxiv 2025, <https://doi.org/10.48550/arXiv.2502.15840>
- [16] Masalkhi, M.; Ong, J.; Waisberg, E. et al.: A side-by-side evaluation of Llama 2 by meta with ChatGPT and its application in ophthalmology. Eye (London, England) 38 (2024) 10, pp. 1789–1792
- [17] Karia, S.; Mehta, M.; Konar, K. et al.: BankBot: Contactless Machine Learning Chatbot for Communication during COVID-19 in Bank. SSRN Electronic Journal (2020), <https://doi.org/10.2139/SSRN.3747951>
- [18] Wang, C.; Ong, J.; Wang, C. et al.: Potential for GPT Technology to optimize future clinical decision-making using retrieval-augmented generation. Annals of biomedical engineering 52 (2024) 5, pp. 1115–1118
- [19] Rittikulsittichai, S.; Siriborvornratanakul, T.: An intelligent chatbot assistant for comprehensive troubleshooting guidelines and knowledge repository in printed circuit board production. IEEE Transactions on Artificial Intelligence 6 (2025) 5, pp. 1259–1268
- [20] Vidyaratne, L.; Shao, H.; Watanabe, T. et al.: Generating troubleshooting trees with FMEA using Large Language Models (LLM). 2025 IEEE International Conference on Prognostics and Health Management (ICPHM), Denver, CO, USA, 2025, pp. 1–10
- [21] Wen, S.; Chen, Y.; Pan, X. et al.: Enhancing Fault Troubleshooting through Human-Machine Collaboration: A Multi-Stage Reasoning Approach. 2024 IEEE 20th International Conference on Automation Science and Engineering (CASE), Bari, Italy, 2024, pp. 460–467
- [22] Chandel, P.; Balkrishan, D. K.; Doke, P.: A comparative study of voice and graphical user interfaces with respect to literacy levels. ACM DEV '13: Annual Symposium on Computing for Development, Bangalore India, 2013, pp. 1–2
- [23] Furqan, A.; Myers, C.; Zhu, J.: Learnability through adaptive discovery tools in voice user interfaces. CHI '17: CHI Conference on Human Factors in Computing Systems, Denver Colorado USA, 2017, pp. 1617–1623
- [24] Allen, J.; Galescu, L.; Teng, C. M. et al.: Conversational agents for complex collaborative tasks. AI Magazine 41 (2020) 4, pp. 54–78
- [25] Le Bigot, L.; Caroux, L.; Ros, C. et al.: Investigating memory constraints on recall of options in interactive voice response system messages. Behaviour & Information Technology 32 (2013) 2, pp. 106–116
- [26] Pasztor, D.: Combining Graphical And Voice Interfaces For A Better User Experience. Stand: 2017. Internet: [www.smashingmagazine.com/2017/10/combining-graphical-voice-interfaces/#:~:text=The%20freedom%20that%20a%20voice,Spotify%20albums%20with%20dirty%20hands](http://www.smashingmagazine.com/2017/10/combining-graphical-voice-interfaces/#:~:text=The%20freedom%20that%20a%20voice,Spotify%20albums%20with%20dirty%20hands). Zugriff am 26.02.2025

**Siddhesh Vivek Dixit, M.Sc.**   
[siddhesh.vivek.dixit@ipa.fraunhofer.de](mailto:siddhesh.vivek.dixit@ipa.fraunhofer.de)

**Jannik Schwab, M.Sc.**   
[jannik.schwab@ipa.fraunhofer.de](mailto:jannik.schwab@ipa.fraunhofer.de)

Fraunhofer-Institut für Produktionstechnik  
 und Automatisierung IPA   
 Nobelstr. 12, 70569 Stuttgart  
[www.ipa.fraunhofer.de](http://www.ipa.fraunhofer.de)

## LIZENZ



Dieser Fachaufsatz steht unter der Lizenz Creative Commons  
 Namensnennung 4.0 International (CC BY 4.0)