

AI Engineering

Human-AI UX & Vertrauen

Vertrauen soll nicht maximiert, sondern kalibriert werden.

Band 2 · Kapitel 35

Öffentliche Leseprobe · Auszug (§1 „Warum Human-AI-UX anders ist“ bis §6 „Vorschlag ≠ Entscheidung ≠ Aktion ≠ bestätigter Effekt“)

Sebahattin Özata · Diplom-Informatiker

OMEX Press

Aus Kapitel 35 in Band 2, dem Produktteil: wie technische Wahrheit in verständliche Produktinteraktion übersetzt wird. Kapitelverweise beziehen sich auf das vollständige dreibändige Werk.

Kapitel 35 — Human-AI UX & Vertrauen

Teil IX — Produkt, Vertical & wissenschaftlicher Abschluss.

Voraussetzung: K24 (AI Control Plane), K28

(Governance/Audit/Replay). Dieses Kapitel setzt die technische Wahrheit der Kapitel K24–K34 voraus und fragt: *Wie erlebt ein Mensch diese Mechanismen — und wie wird technische Wahrheit in verständliche, vertrauenswürdige Produktinteraktion übersetzt?*

AI (Artificial Intelligence), **LLM** (Large Language Model), **UX** (User Experience) und **UI** (User Interface): Der bequemste Fehlschluss im AI-Produktbau lautet, gute Human-AI-UX entstehe aus einem schönen Chatfenster. Das ist falsch. Ein Chatfenster ist eine Oberfläche; Vertrauen ist ein *Verhältnis* zwischen dem, was das System wirklich kann, und dem, was der Mensch glaubt, dass es kann. Dieses Kapitel handelt nicht davon, einen Chatbot hübsch zu machen. Es handelt von **kalibriertem Vertrauen** (*calibrated trust*): einem Produkt, das den Menschen befähigt, Fähigkeiten, Unsicherheiten, Grenzen, Entscheidungen und Auswirkungen des Systems *richtig einzuschätzen*.

Die zentrale These: Vertrauen ist kein UI-Feature. Vertrauen entsteht, wenn **Systemverhalten**, **Nutzererwartung** und **sichtbare Evidenz** ausreichend gut übereinstimmen. Und: **Gute Human-AI-UX maximiert nicht Vertrauen — sie kalibriert es.**

1. Warum Human-AI-UX anders ist

Klassische Software ist überwiegend deterministisch: gleiche Eingabe, gleiches Ergebnis, klar definierte Fehler. Ein AI-System bricht drei stille Annahmen der klassischen UX:

1. **Nichtdeterminismus.** Dieselbe Frage kann zweimal verschiedene Antworten liefern (K33). Die UI kann nicht so tun, als sei jede Ausgabe eine feststehende Wahrheit.

2. **Unschärfe Grenzen.** Ein Button „Speichern“ hat eine klare Fähigkeit. Ein Assistent, der „bei Steuerfragen hilft“, hat eine *unscharfe* Fähigkeit — er kann in einem Fall exzellent und im Nachbarfall zuversichtlich falsch sein.
3. **Wirkung jenseits der Oberfläche.** Ein Agent kann Werkzeuge aufrufen, Dokumente schreiben, Nachrichten senden — reale Effekte mit realer Haftung (K26/K30).

Damit wandert die eigentliche UX-Aufgabe vom *Aussehen* zum *Verständnis*: Der Mensch muss aus der Oberfläche ablesen können, **wie sicher** eine Aussage ist, **worauf** sie sich stützt, **was** das System tun darf und **was schon geschehen ist**. Fehlt das, entsteht kein Vertrauen, sondern eine Illusion davon.

Merksatz: In klassischer Software gestaltet man Bedienbarkeit. In AI-Software gestaltet man zusätzlich das *Mentalmodell* — die innere Vorstellung des Nutzers davon, was das System kann und darf.

2. Vertrauen kalibrieren — Undertrust, Calibrated, Overtrust

Das Ziel ist nicht, dass Nutzer der AI *maximal* vertrauen. Es ist, dass ihr Vertrauen der *realen Fähigkeit* entspricht — was die Human-Factors-Forschung **appropriate reliance** nennt (Lee & See, 2004 [REAL]).

Drei Zustände (T1, D1):

- **Undertrust (Untertrauen).** Der Nutzer ignoriert hilfreiche, korrekte AI — Doppelarbeit, verworfener Nutzen. Verwandt mit **Algorithm Aversion** (Dietvorst et al., 2015 [REAL]).
- **Calibrated Trust (kalibriert).** Vertrauen $\approx$ Fähigkeit. Bei niedriger Fähigkeit ist der Nutzer skeptisch, prüft, eskaliert; bei hoher Fähigkeit verlässt er sich und prüft stichprobenartig.
- **Overtrust (Übertrauen).** Der Nutzer akzeptiert Ergebnisse und Aktionen, die er nicht ausreichend prüfen kann — **Automation Bias**, Rubber-Stamp-Approval, ungeprüfte Zitate (Parasuraman & Riley, 1997 [REAL]).

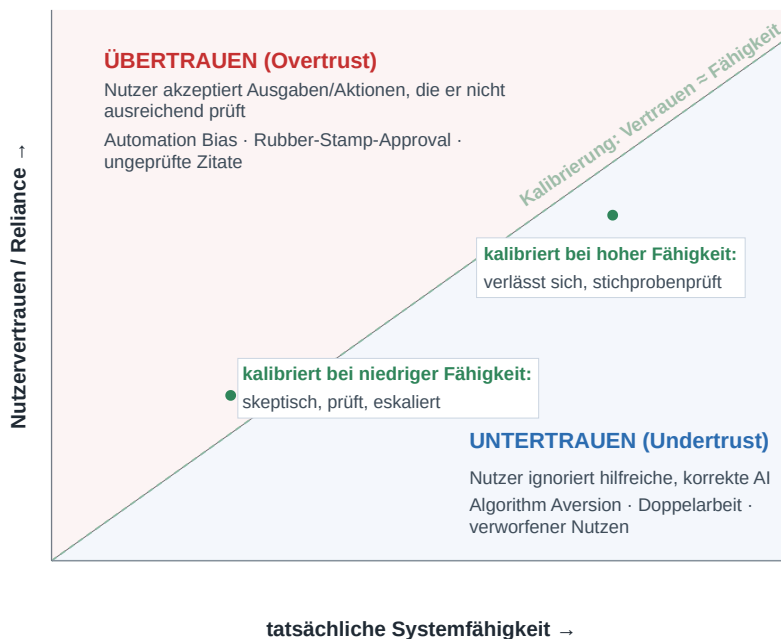
T1 · Vertrauens-Begriffe (nicht synonym)

Begriff	Bedeutung	Fallstrick
Model Confidence	modellinterner Score / Log-Prob	oft nicht kalibriert, nicht als Wahrscheinlichkeit lesbar
System Confidence	Produkt-Einschätzung aus mehreren Signalen	Konstruktion des Produkts, kein Modellwert
Evidence Strength	wie belastbar die Belege sind	prüfbar — schlägt Prosa
Uncertainty	was das System <i>nicht</i> weiß	fehlende Evidenz ≠ niedrige Wahrscheinlichkeit
Trust	Nutzer-Einstellung zum System	Ziel: kalibriert, nicht maximal
Reliance	tatsächliches Verlassen/Handeln	messbares Verhalten, nicht Gefühl

Merksatz: Eine UX, die Vertrauen *maximiert* statt *kalibriert*, verschiebt den Nutzer systematisch ins Übertrauen. Das ist kein besseres Produkt — es ist eine Haftungsquelle.

Kalibriertes Vertrauen — nicht maximales Vertrauen

gutes Produkt richtet das Nutzervertrauen an der tatsächlichen Systemfähigkeit aus, nicht darüber



Ziel: Nutzer-Mentalmodell $\approx$ reale Fähigkeit. Weder blindes Vertrauen noch pauschales Misstrauen — passendes Verlassen (appropriate reliance).

UX, die Vertrauen maximiert statt kalibriert, verschiebt die Kurve nach oben in die ÜBERTRAUEN-Zone — das erzeugt Übertrauen, kein besseres Produkt.

D1 · Kalibriertes Vertrauen

3. Mentalmodelle und die Vertrauens-Übertragungskette

Vertrauen entsteht nicht direkt, sondern über eine **Kette** (D2): reale Fähigkeit → System-Evidenz → UI-Darstellung → Mentalmodell → Reliance/Aktion → Ergebnis → Feedback. Jede Stufe kann die Wahrheit verzerren:

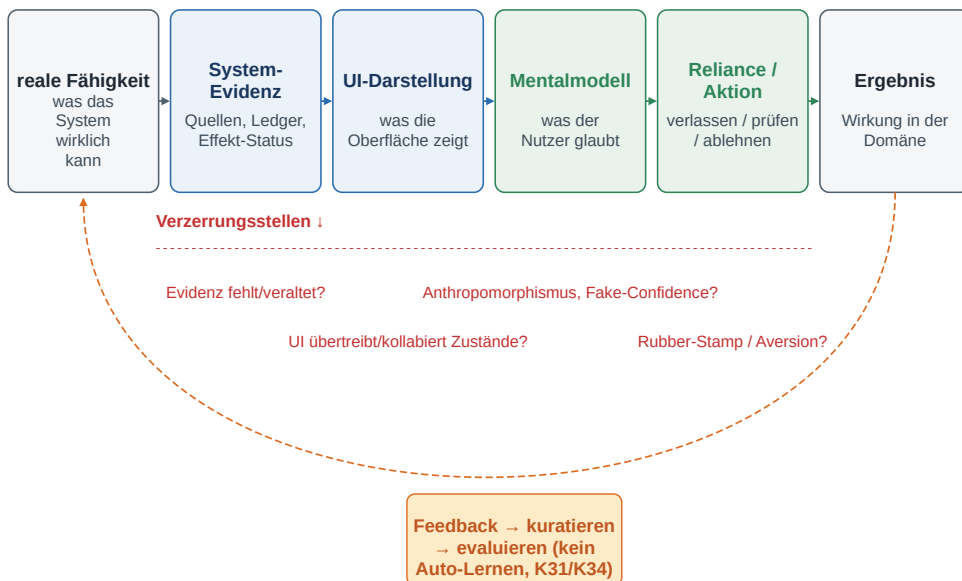
- Die **Evidenz** fehlt oder ist veraltet (K32).
- Die **UI** übertreibt oder kollabiert Zustände zu einem grünen „Erledigt“.

- Das **Mentalmodell** wird durch Anthropomorphismus oder Fake-Confidence verschoben.
- Die **Reliance** kippt in Rubber-Stamp (Übertrauen) oder Aversion (Untertrauen).

Ein starkes Backend nützt nichts, wenn die UI-Stufe die Evidenz verschluckt; eine hübsche UI *schadet*, wenn sie mehr Fähigkeit suggeriert als vorhanden. Das Produkt hat die Aufgabe, die Kette *ehrlich* zu halten.

Die Kette, über die Vertrauen entsteht — und wo sie bricht

jede Stufe kann die Wahrheit verzerren; Ziel ist Deckung von Mentalmodell und realer Fähigkeit



Vertrauen ist kein UI-Feature — es entsteht, wenn Systemverhalten, Nutzererwartung und sichtbare Evidenz übereinstimmen.

Ein starkes Backend nützt nichts, wenn die UI-Stufe die Evidenz verschluckt; eine hübsche UI schadet, wenn sie mehr Fähigkeit suggeriert als vorhanden.

D2 · Die Vertrauens-Übertragungskette

Der reale Lehrfall für eine gebrochene Kette ist **Moffatt v. Air Canada** (BC Civil Resolution Tribunal, 2024) [REAL]: Der Website-Chatbot der Airline gab eine falsche Auskunft zu einem Trauertarif; der Kunde verließ sich darauf. Das Tribunal wies das Argument zurück, der Chatbot sei „a separate legal entity“, und hielt die Airline für die Aussage ihres eigenen Bots haftbar. Die UX-Lehre ist nicht „Chatbots sind gefährlich“, sondern: **wer die Oberfläche betreibt, haftet für das Mentalmodell, das sie erzeugt** — die Verantwortung wandert nicht zum Modell (vgl. K34, „Produktverantwortung nicht auslagern“).

4. Automation Bias und Algorithm Aversion — beide Richtungen

Zwei gegenläufige menschliche Tendenzen bedrohen die Kalibrierung — und keine davon ist „der Nutzer ist dumm“:

- **Automation Bias / Misuse.** Menschen akzeptieren Systemvorschläge zu leicht, besonders unter Zeitdruck oder wenn die Oberfläche Autorität ausstrahlt. Sie übersehen Fehler, die sie bei einem menschlichen Kollegen bemerkt hätten (Parasuraman & Riley, 1997 [REAL]).
- **Algorithm Aversion / Disuse.** Menschen verwerfen ein algorithmisches System nach einem *sichtbaren* Fehler stärker als einen vergleichbaren menschlichen Fehler (Dietvorst et al., 2015 [REAL]). Die Gegenrichtung — **Algorithm Appreciation** — existiert ebenfalls (Logg et al., 2019 [REAL]); der Effekt ist kontextabhängig, keine Einbahnstraße.

Die Produktfolgen sind konkret, nicht psychologische Prosa: Ein **Confirmation-Schritt** bremst Automation Bias bei riskanten Aktionen; **sichtbare Evidenz** ersetzt blindes Akzeptieren durch Prüfen; ein **Override-/Reject-Weg** und **Eskalation** fangen Aversion produktiv auf, statt den Nutzer die AI abschalten zu lassen. Die Richtung, in die eine UX kippt, ist eine Design-Entscheidung — kein Naturgesetz.

5. Capability Boundaries und Affordances

Eine gute AI-UX macht die *Grenzen* der Fähigkeit sichtbar, nicht nur die Fähigkeit. Der Nutzer soll ablesen können: **Was kann die AI? Was nicht? Was darf sie? Was braucht Freigabe? Was ist rückgängig, was irreversibel? Was ist nur Vorschlag, was wird ausgeführt?**

Dafür braucht die Oberfläche präzise **Verben** — jedes signalisiert eine andere Wirkung und Reversibilität:

Verb	Bedeutung	Wirkung
<i>Suggest / Draft</i>	Vorschlag/Entwurf	keine — verwerfbar
<i>Preview</i>	Vorschau des Effekts	keine — nur Anzeige
<i>Approve</i>	Freigabe erteilen	schaltet Ausführung frei
<i>Execute</i>	Aktion ausführen	erzeugt Wirkung/Effekt
<i>Undo</i>	rückgängig	nur bei reversiblen Zustand
<i>Retry</i>	erneut versuchen	gefährlich bei externem Effekt (§6/§11)
<i>Escalate</i>	an Mensch/Instanz weiterreichen	verlagert Entscheidung

Merksatz: Eine Fähigkeit, die die UI *nicht benennt*, existiert für den Nutzer nicht — und eine Grenze, die die UI *verschweigt*, wird zur Falle. Affordances sind Ehrlichkeit über Wirkung.

6. Vorschlag ≠ Entscheidung ≠ Aktion ≠ bestätigter Effekt

Die wichtigste einzelne Unterscheidung dieses Kapitels (D3, T2): Vier Zustände, die die UI **nicht** zu einem grünen „Erledigt“ kollabieren darf.

T2 · Vier-Zustands-Semantik

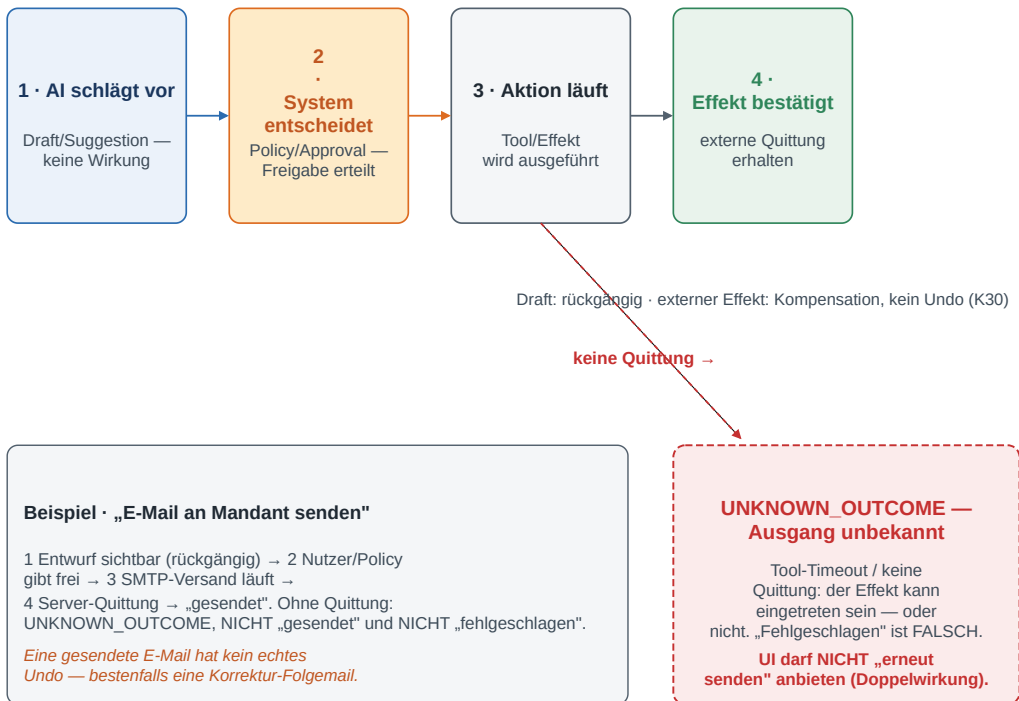
Zustand	Was gilt	Was die UI zeigen muss	Bezug
Vorschlag	AI hat etwas <i>vorgeschlagen</i>	keine Wirkung; verwerfbar	K34 §9
Entscheidung	System/Policy/Mensch hat <i>freigegeben</i>	wer, warum, gegen welche Argumente	K24/K26
Aktion	Ausführung <i>läuft</i>	Status, laufender Effekt	K30
Bestätigter Effekt	externe <i>Quittung</i> liegt vor	„bestätigt“ erst mit Quittung	K25/K28

Der gefährlichste Zweig ist der **unbestätigte Ausgang**. Konsistent mit K25/K26/K30 gilt: es gibt **kein external exactly-once**, und UNKNOWN_OUTCOME bedeutet **nicht** „kein Effekt“. Wenn ein Werkzeug ohne Quittung zurückkommt (Timeout), *kann* der Effekt eingetreten sein — oder nicht.

Merksatz: Bei UNKNOWN_OUTCOME darf die UI **nicht** „Fehlgeschlagen — erneut versuchen“ anzeigen. Ein blinder Retry ist potenziell eine **zweite reale Wirkung** (doppelte E-Mail, doppelte Buchung). „Vielleicht ausgeführt“ ist ein *eigener Zustand*, kein Fehler.

Vorschlag ≠ Entscheidung ≠ Aktion ≠ bestätigter Effekt

die vier Zustände dürfen in der Oberfläche nicht zu „erledigt“ kollabieren



Kein external exactly-once (K25/K26/K30): die UI muss „vielleicht ausgeführt“ als eigenen Zustand führen, nicht als Fehler.

Kollabiert die UI diese vier Zustände in ein grünes „Erledigt“, verliert der Nutzer die Fähigkeit, Wirkung von Absicht zu unterscheiden.

D3 · Vorschlag ≠ Entscheidung ≠ Aktion ≠ bestätigter Effekt

Markierungen in dieser Leseprobe

Einzelne Sätze tragen eine Markierung, die sagt, woher die Aussage stammt und wie belastbar sie ist. Hier stehen nur die, die in dieser Leseprobe vorkommen.

Markierung	Bedeutung
[REAL]	ein dokumentierter realer Vorfall, kein konstruiertes Beispiel.